

上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

课程作业

COURSE HOMEWORK



题目： 数字体温计的设计与实现

姓名： 张露雨

学号： 519130910018

# 数字体温计的设计与实现

## 1. 系统需求

在 Proteus 环境下，基于 MCS-51，采用并行接口的 ADC0809 模数转换器和模拟温度传感器，设计一个数字体温计，具体包括

- (1) 模拟温度信号的采集和模数转换；
- (2) AD 采集，以及 ADC 与单片机的接口，
- (3) 单片机对输入数字信号的处理，显示和传输（基于 UART 的无线蓝牙传输和接收）等功能。

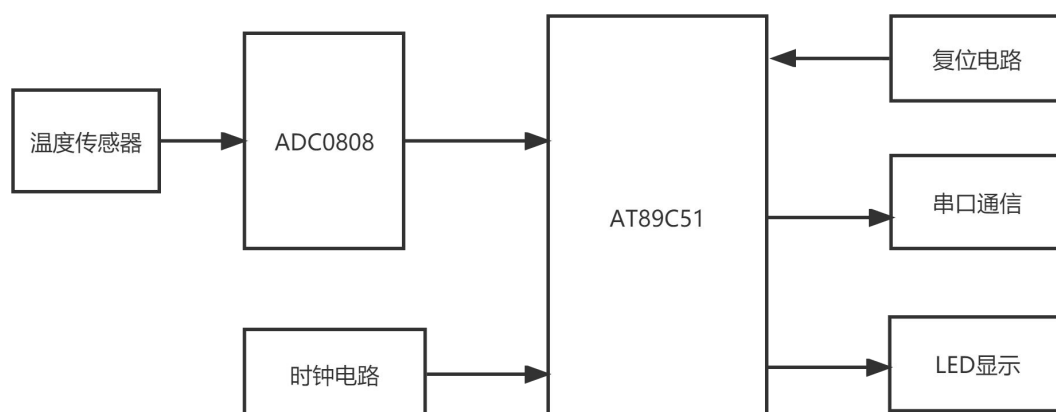
## 2. 人体体温的特征：

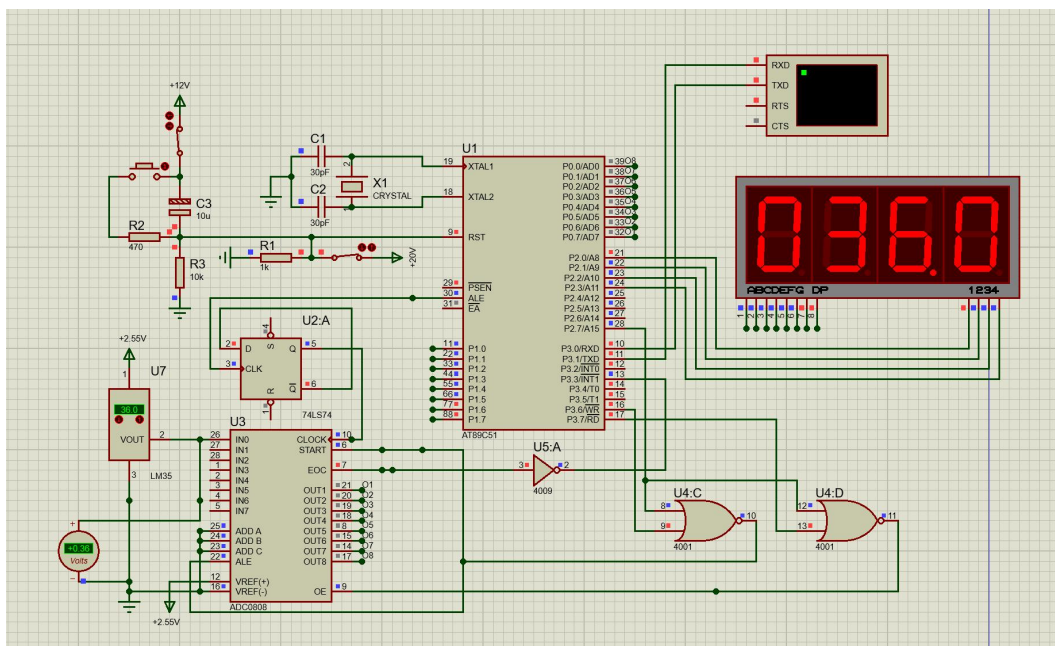
人体体温是指人体内部温度的度量，通常以摄氏度（℃）或华氏度（°F）为单位。正常情况下，人体的体温在 36.1℃ 到 37.2℃ 之间，具体数值还会受到多种因素的影响，如时间、环境温度、个体差异等。

在特殊情况下，由于环境的剧烈变化以及疾病等因素的干扰，人体体温还会再 35℃ 到 42℃ 之间变换，这一系列特征为元器件选型奠定了基础。

## 3 硬件设计

### 3.1 硬件逻辑图





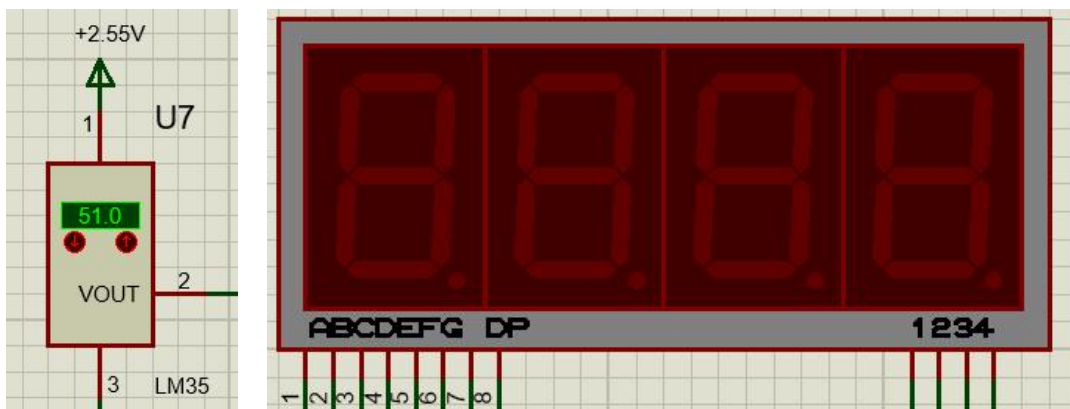
## 3.2 硬件选型:

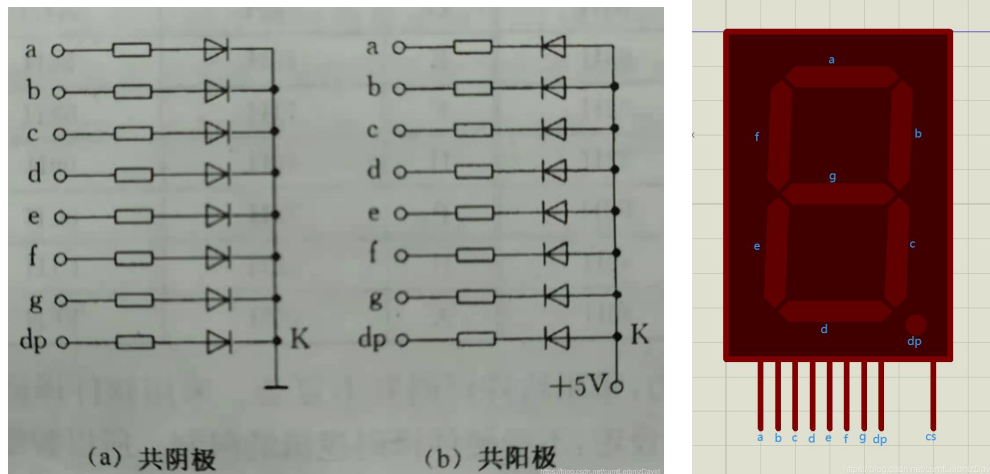
### 3.2.1 LM35 温度传感器:

LM35 是一种精度高、线性好、输出电压直接表示摄氏温度的模拟温度传感器。它广泛应用于各种电子设备中，特别是需要精确温度控制的电路中。LM35 传感器采用了 National Semiconductor 公司的微功耗工艺，耗电量极小，一般不超过 60 微安。LM35 传感器的工作电压范围为  $+4V \sim +30V$ ，测量温度范围为  $-55^{\circ}C \sim 150^{\circ}C$ ，测量精度高达  $\pm 0.5^{\circ}C$ 。

LM35 传感器的输出电压与测量的温度成正比，即每摄氏度温度变化会引起 10 毫伏的输出电压变化。因此，只需将传感器输出的电压通过一个简单的电路进行放大和滤波，就可以得到非常精确的温度读数。

LM35 传感器非常易于使用，只需要将其连接到电源和测量电路即可。它不需要校准或调整，而且具有很高的可靠性和长寿命。因此，LM35 传感器在各种温度测量应用中都是一种非常优秀的选择。





### 3. 2. 2. 7SEG-MPX4-CA 四位数码管

7SEG-MPX4-CA 是一种常见的数码管，常用于数字显示。它由 4 个 7 段数码管和一个集中式驱动器组成，因此可以同时显示 4 位数字。这种数码管的 7 段式显示可以显示数字 0 到 9 以及一些字母，如 A、B、C、D、E 和 F。每个数码管还有一个小数点，可以用于显示小数。

7SEG-MPX4-CA 数码管的工作电压通常在 2V 到 4V 之间，它们可以被串联起来以显示更多的数字。在使用时，需要将每个数码管的引脚连接到电路板上的相应引脚上，以便能够控制每个数码管的显示。控制器可以是基于微控制器的系统或专用驱动器芯片。

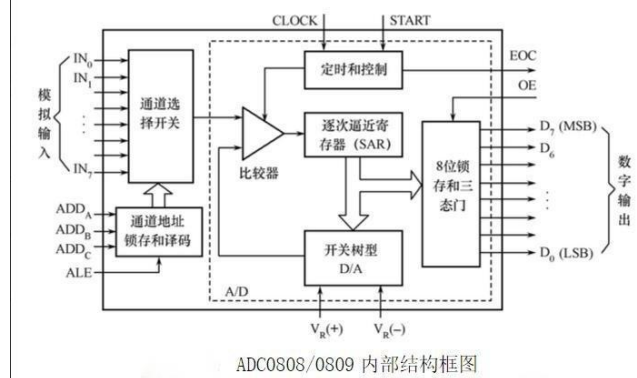
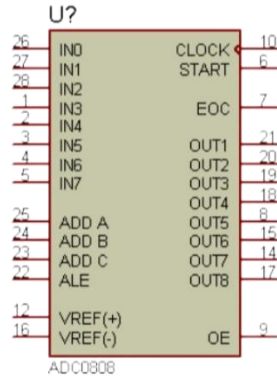
总之，7SEG-MPX4-CA 数码管是一种常见的数字显示设备，广泛用于计数器、计时器、温度计等应用中。

### 3. 2. 3. ADC0808 模数转换器

ADC0808 是一种 8 位的模数转换器 (ADC)，由 National Semiconductor 公司（现在是德州仪器公司的一部分）制造。它可以将模拟信号转换为数字信号，可以被数字系统（如微处理器）处理。ADC0808 是一种单通道、逐次逼近型 ADC，它采用的是逐次逼近转换技术。

ADC0808 采样率较低，最大转换速率为 100kHz。它有 8 个输入通道，可以选定其中的一个进行转换。输入电压范围为 0V 到 VCC (5V 或者 5.5V)，输出数据以并行方式提供。它还有一些特殊的控制引脚，如转换开始引脚和转换结束引脚，以及一个片选引脚。

由于 ADC0808 只有 8 位分辨率，因此它的精度相对较低。然而，它的价格相对较低，适合于许多应用，如温度测量、电压测量等。现在，更高精度的 ADC 已经问世，但 ADC0808 仍然是一种可靠的、经济实惠的解决方案。



### 3.2.4. AT89C51

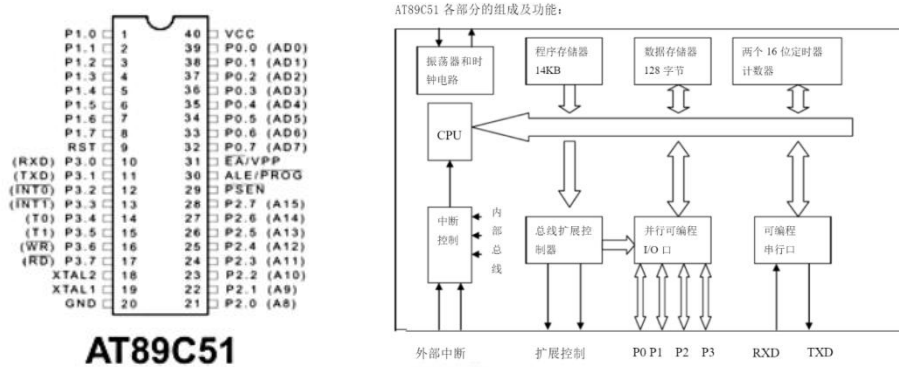
AT89C51 是基于 8051 内核的一款单片机，它是一款高性能、低功耗的单片机，具有 8 位 CPU、片内 ROM、RAM、I/O 端口、定时器/计数器、串行口等基本特征。

AT89C51 与传统的 8051 单片机相比，具有更高的工作频率，最高可达 33MHz。它采用的是 Flash 技术，内部集成了 32KB 的 Flash 程序存储器和 1KB 的 RAM 数据存储器，可以存储大量的应用程序。

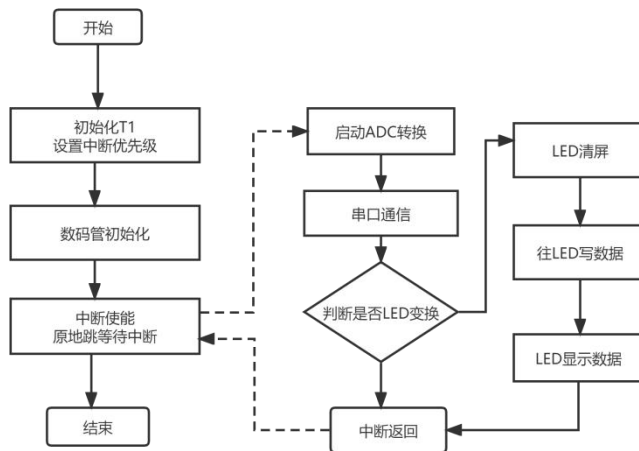
AT89C51 还具有丰富的外设，如增强型 PWM 输出、高速串行通信接口、多重定时器/计数器等。它还支持多种通信协议，如 SPI、I2C 等，方便与其他器件进行通信。

AT89C51 还支持中断控制，能够快速响应外部事件，提高系统的实时性和可靠性。此外，它还支持 ISP (In-System Programming) 技术，可以通过串行口进行在线编程和调试，方便开发人员进行调试和测试。

总的来说，AT89C51 是一款功能强大、易于开发、可靠性高的单片机，适用于各种应用场景，如智能家居、工业自动化、安防监控等。



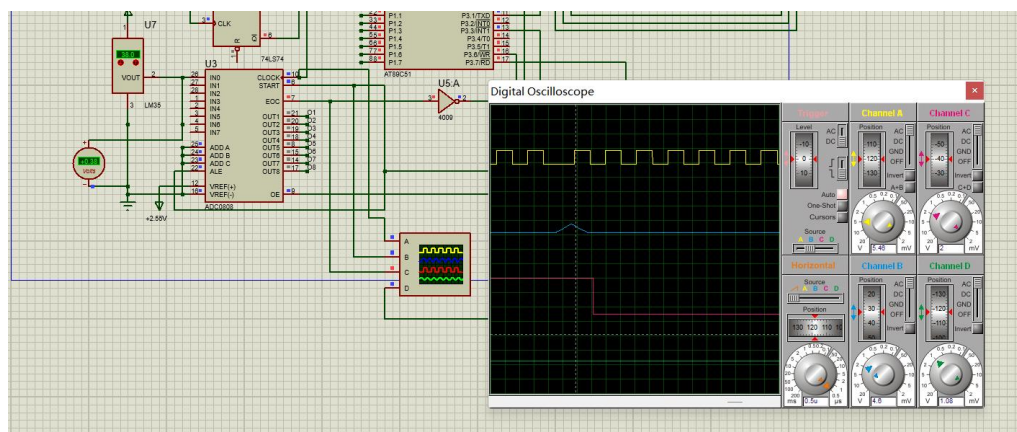
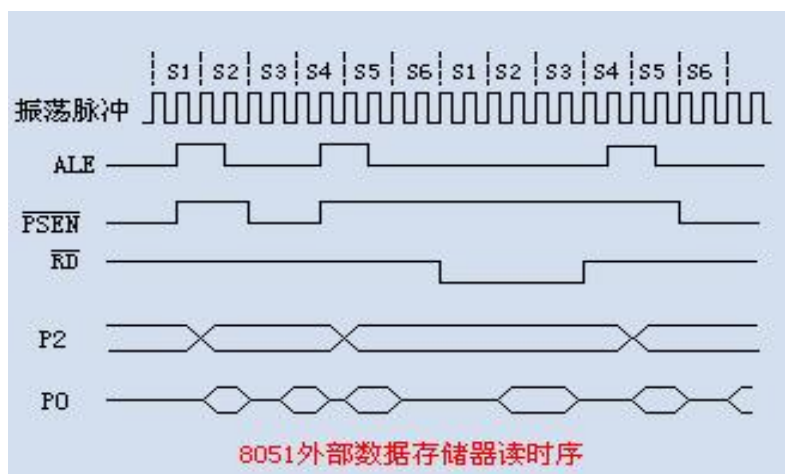
## 4 软件设计



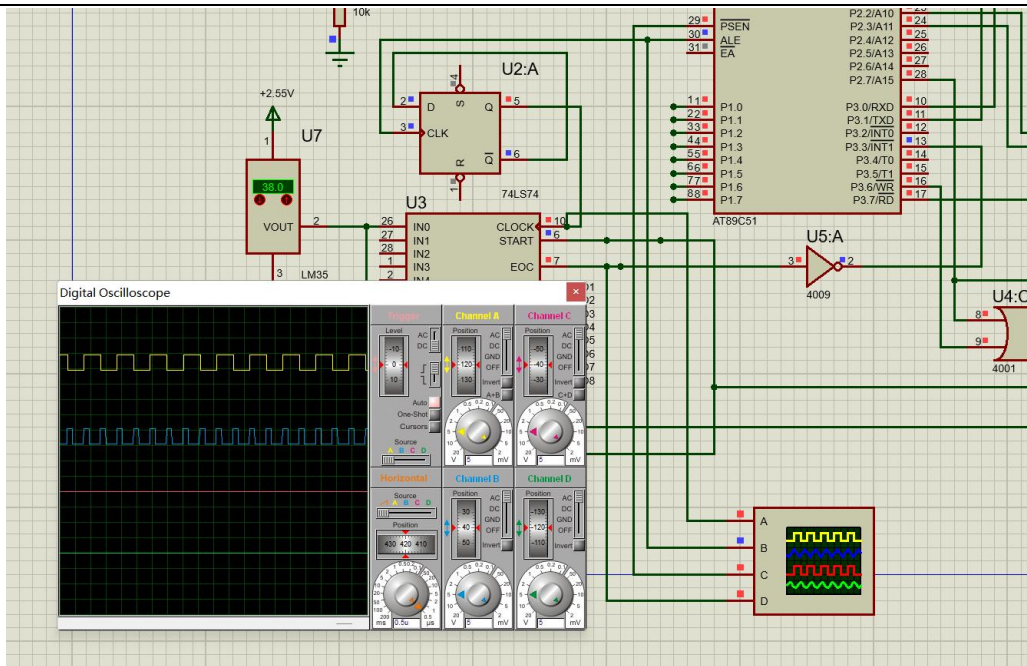
## 5. 最终结果

### 5.1 时序图

8051 外部数据存储器的理论时序图如下所示，由于示波器接口数量显示，我们将其分开抓取。

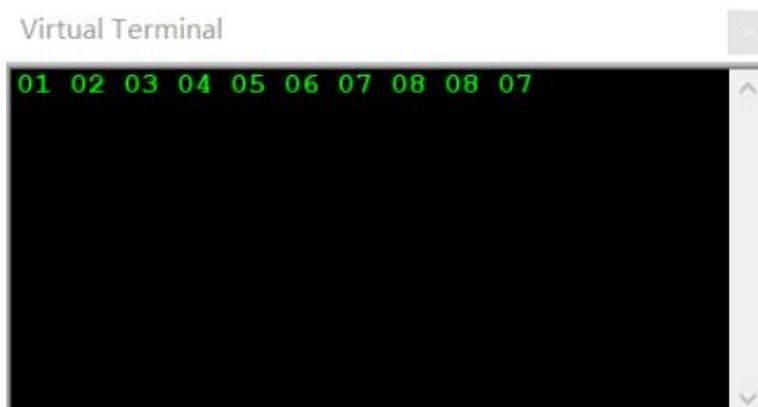
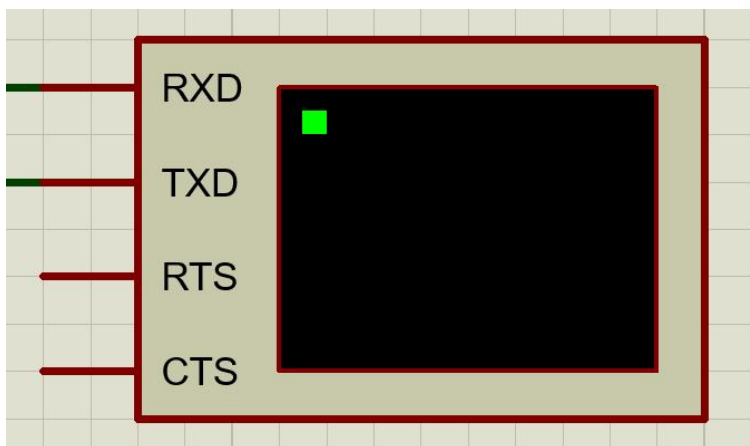






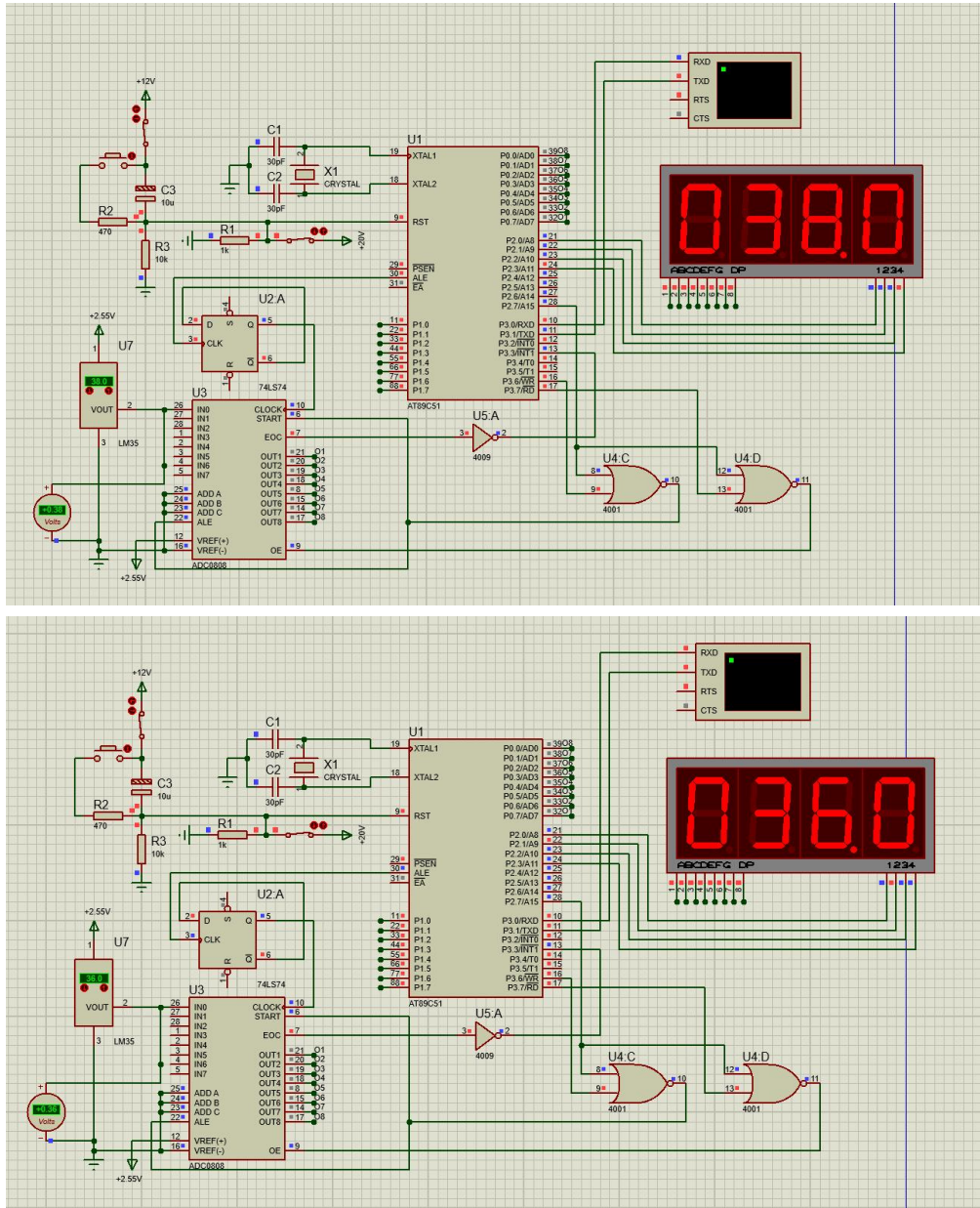
## 5.2 串口通信

我们使其在终端中输出按顺序排列的数字，具体如图所示

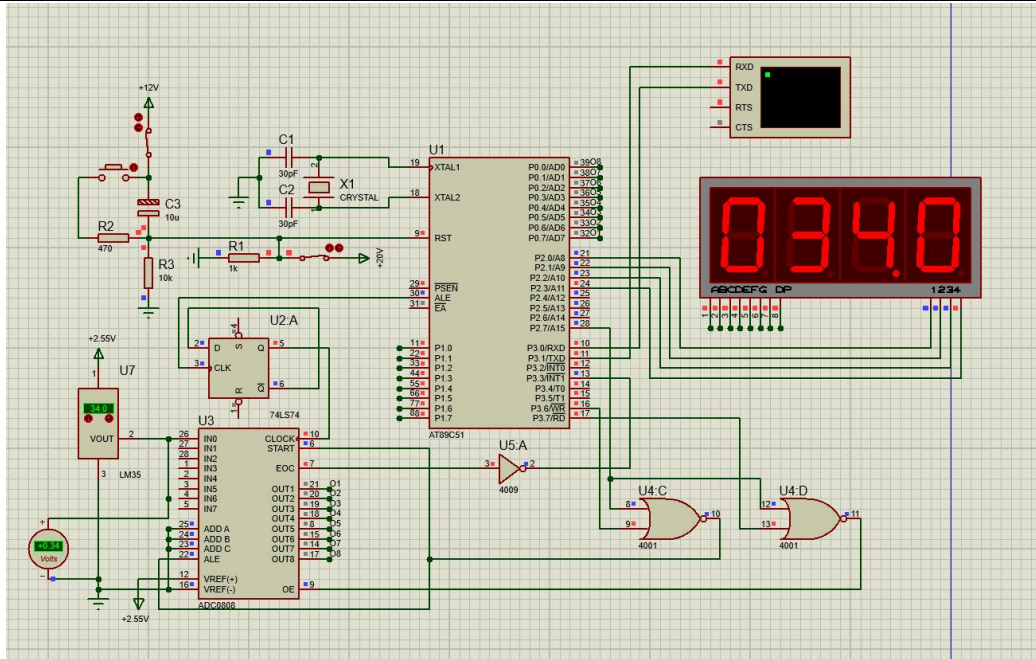


## 5.3 结果展示

下面进行不同温度下结果展示图如图所示







## 6 总结与展望

在这个项目中，成功实现了一个数字体温计的设计，其是用来测量人体温度的，它通过模拟温度信号的采集和模数转换，将实际温度转换为数字信号，再通过 AD 采集，将数字信号转换为电子信号，最后通过 ADC 与单片机的接口的设计，将电子信号传输到单片机中，实现对体温的准确测量。此外，还完成了串口接收等相关的调试和验证工作，确保数字体温计的可靠性和稳定性。

在项目开始的时候，并不知道如何设计数字体温计，但是在老师和同学的帮助下，一步步学习并完成了这个项目。在设计过程中遇到了很多问题，例如如何选择合适的模拟转换器和数字转换器，如何进行电路的布局等等。但是，最终克服了这些问题，并逐步熟悉了数字体温计的设计流程。在项目的调试和验证过程中，也经历了很多的挑战。为了解决这些问题，通过不断地修改和优化设计方案，最终成功地实现了一个稳定可靠的数字体温计。

这个项目的成功实现，不仅让我获得了实际的设计和调试经验，也让我体验到了交流合作和协作的重要性。通过相互之间的学习和帮助，不仅克服了项目中的各种问题，也相互激励，共同推动了项目的进展和成功实现。

## 附录一、源码

```

1.  ORG 0000H
2.      LJMP START
3.  ORG 0013H
4.      LJMP INTT
5.  ORG 040H
6.
7.  START:
8.      SETB EA      ;CPU 开中断
9.      SETB EX1     ;选择外中断为跳沿触发方式
10.     CLR  PX1
11.     SETB INT1     ;外部中断 1 初始化编程
12.     MOV  R1,#0FFH ;灭灯
13.     MOV  R2,#0FFH
14.     MOV  R3,#0FFH ;灭灯
15.     MOV  DPTR,#7FF8H ;DPTR 指向 0808 通道 0
16.     MOVX @DPTR,A ;启动 0808 对 IN0 通道转换
17.     MOV  SP,#60H ;设置堆栈指针指向 RAM 地址 60H
18.     MOV  SCON,#50H ;设置串行口工作方式 1,接收数据设为允许
19.     MOV  TMOD,#20H ;定时器 1 置为方式 2
20.     MOV  TH1,#0FDH ;装载定时器初值,波特率 9600
21.     MOV  TL1,#0FDH ;装载定时器初值,波特率 9600
22.     SETB TR1      ;启动定时器
23.     SJMP $
24.  INTT:
25.     MOV  P1,#0FFH ;灭灯
26.     MOVX A,@DPTR ;读取 A/D 结果
27.     MOV  30H,A ;将结果放到内部 RAM 单元 30H
28.     MOV  SBUF,A ;送出欲传送的数据
29.     JNB  TI,$ ;等待一帧数据发送结束
30.     CLR  TI ;清传送结束标志
31.
32.  NEXT:
33.     MOV  A,30H;
34.     MOV  B,#10;
35.     DIV  AB;
36.     PUSH B ;放入第 3 位
37.     MOV  B,#10;
38.     DIV  AB;
39.     PUSH B ;放入第 2 位
40.     MOV  B,#10;
41.     DIV  AB;
42.     MOV  A,B

```

```

43.  N10:    CJNE A,#0,N11;
44.          MOV R1,#0C0H    ;0
45.    JMP  N20
46.  N11:    CJNE A,#1,N12;
47.          MOV R1,#0CFH    ;1
48.    JMP  N20
49.  N12:    CJNE A,#2,N13;
50.          MOV R1,#0A4H    ;2
51.    JMP  N20
52.  N13:    CJNE A,#3,N14;
53.          MOV R1,#0B0H    ;3
54.    JMP  N20
55.  N14:    CJNE A,#4,N15;
56.          MOV R1,#099H    ;4
57.    JMP  N20
58.  N15:    CJNE A,#5,N16;
59.          MOV R1,#092H    ;5
60.    JMP  N20
61.  N16:    CJNE A,#6,N17;
62.          MOV R1,#082H    ;6
63.    JMP  N20
64.  N17:    CJNE A,#7,N18;
65.          MOV R1,#0F8H    ;7
66.    JMP  N20
67.  N18:    CJNE A,#8,N19;
68.          MOV R1,#080H    ;8
69.    JMP  N20
70.  N19:    MOV  R1,#090H    ;9
71.
72.  N20:    POP  B
73.          MOV  A,B
74.          CJNE A,#0,N21    ;
75.          MOV  R2,#0C0H    ;0
76.    JMP  N30
77.  N21:    CJNE A,#1,N22;
78.          MOV  R2,#0CFH    ;1
79.    JMP  N30
80.  N22:    CJNE A,#2,N23    ;
81.          MOV  R2,#0A4H    ;2
82.    JMP  N30
83.  N23:    CJNE A,#3,N24    ;
84.          MOV  R2,#0B0H    ;3
85.    JMP  N30
86.  N24:    CJNE A,#4,N25    ;

```

87.		MOV	R2,#099H	;4
88.		JMP	N30	
89.	N25:	CJNE	A,#5,N26	;
90.		MOV	R2,#092H	;5
91.		JMP	N30	
92.	N26:	CJNE	A,#6,N27	;
93.		MOV	R2,#082H	;6
94.		JMP	N30	
95.	N27:	CJNE	A,#7,N28	;
96.		MOV	R2,#0F8H	;7
97.		JMP	N30	
98.	N28:	CJNE	A,#8,N29	;
99.		MOV	R2,#080H	;8
100.		JMP	N30	
101.	N29:	MOV	R2,#090H	;9
102.				
103.	N30:	POP	B	
104.		MOV	A,B	
105.		CJNE	A,#0,N31	;
106.		MOV	R3,#0C0H	;0.
107.		JMP	N40	
108.	N31:	CJNE	A,#1,N32	;
109.		MOV	R3,#0CFH	;1.
110.		JMP	N40	
111.	N32:	CJNE	A,#2,N33	;
112.		MOV	R3,#0A4H	;2.
113.		JMP	N40	
114.	N33:	CJNE	A,#3,N34	;
115.		MOV	R3,#0B0H	;3.
116.		JMP	N40	
117.	N34:	CJNE	A,#4,N35	;
118.		MOV	R3,#099H	;4.
119.		JMP	N40	
120.	N35:	CJNE	A,#5,N36	;
121.		MOV	R3,#092H	;5.
122.		JMP	N40	
123.	N36:	CJNE	A,#6,N37	;
124.		MOV	R3,#082H	;6.
125.		JMP	N40	
126.	N37:	CJNE	A,#7,N38	;
127.		MOV	R3,#0F8H	;7.
128.		JMP	N40	
129.	N38:	CJNE	A,#8,N39	;
130.		MOV	R3,#080H	;8.

```
131.      JMP  N40
132. N39:    MOV  R3,#090H    ;9.
133.
134. N40:
135.      MOV  R0,#25
136. LOP:    MOV  P2,#01H    ;只显示第 1 位
137.      MOV  P1,R1    ;显示第 1 位数
138.      MOV  P1,#0FFH    ;灭灯
139.      MOV  P2,#02H    ;只显示第 2 位
140.      MOV  P1,R2    ;显示第 2 位数
141.      MOV  P1,#0FFH    ;灭灯
142.      MOV  P2,#04H    ;只显示第 3 位
143.      MOV  P1,R3    ;显示第 3 位数
144.      MOV  P1,#0FFH    ;灭灯
145.      MOV  P2,#08H    ;只显示第 4 位
146.      MOV  P1,#0C0H    ;0
147.      MOV  P1,#0FFH    ;灭灯
148.      DJNZ R0,LOP
149.      MOVX @DPTR,A    ;启动 0808 对 IN0 通道转换
150.
151.      RETI
152.
153.      END ;
```

## 附录二 展示视频

详见文件夹