

# 数字温度计的设计

曾令安<sup>1</sup>, 林宝诚<sup>2</sup>

(1. 无 93 2019080013; 2. 无 98, 2019080030)

## 0 课程任务和实验要求

### 0.1 课程任务

《电子电路课程设计》是一门综合应用模拟电路和数字电路理论进行电子系统设计的课程, 要求设计并制作具有较完整功能的小型电子系统, 它侧重于电子技术理论知识的灵活运用和设计的创新, 因此具有系统性、综合性和探索性。课程任务有:

- (1) 掌握一般电路系统的设计思路和方法
- (2) 培养系统观念和工程观念、解决电路实际问题的能力和探索创新精神
- (3) 培养实验研究的总结和表达能力

### 0.2 实验要求

设计并制作一个数字温度计, 温度传感器选用负温度系数的热敏电阻。

#### (1) 基本要求

模拟电压测量: 在 16~55℃ 范围内测量输出直流电压误差小于 1℃;

数字显示温度值, 在 16~55℃ 范围内测量误差小于 1℃;

自动刷新温度的显示值。

#### (2) 提高要求:

在 -18~80℃ 范围内测量误差小于 1℃, 分辨率 0.1℃。

采用与参考设计不同的电路结构或有较大的改进。

## 1 实验设计

### 1.1 总体方案设计

温度计的实现可划分成两大重要部分, 首先需要能够检测温度变化的传感器, 实验中使用热敏电阻, 其阻值会随温度变化。另外, 需要一个用来解读前述物理量并显示结果的装置, 即将阻值与温度(摄氏度)互相映射。实验中运用一连串电路模块, 逐步完成阻值与温度之间的映射, 最终将温度通过数码管(温度值)和 LED 灯(正负号)进行输出。

### 1.2 电路框图设计

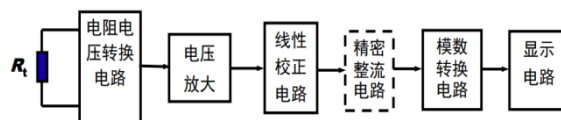


图 1 电路结构框图

根据图 1, 可将温度计的实现细分为 4 个功能模块:

#### (1) 传感器——电阻电压转换电路

将代表温度变化的电阻转换为电压, 因其较方便处理和容易显示

#### (2) 线性校正电路

由于电阻电压部分的设计是根据常温阶段的数据进行拟合(电路设计部分会详细说明), 加上电阻与温度的非线性特性, 高低温的电压会有较大误差, 因此需要通过校正电路获取一个能够在给定温度范围内(-18 至 80 度)随温度线性变化的电压。

#### (3) 精密整流电路

由于设计中, 负温度对应负电压, 因此为了简化模数转换电路的设计, 需要将电压进行取绝对值的操作。

注: 在取电压绝对值前, 已使用 LED 电路将负号显示出来。

#### (4) 模数转换电路

此模块将模拟电压转换成数字信号, 并通过七段数码管将其对应的温度进行输出。

### 1.3 电路设计

#### 1.3.1 电阻电压转换电路

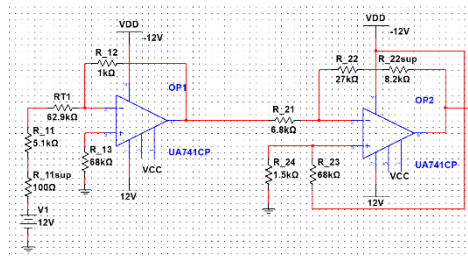


图 2 电阻电压转换电路

图 2 电路模块由两个部分组成，对 U1 与 U2 分别使用电路知识进行推导，能够得出如下结论：

- 运放 U1

$$V_{out1} = -V_1 \frac{R_{12}}{R_T + R_{11}}$$

其中， $V_{out1}$  为 U1 输出

- 运放 U2

$$V_{out2} = V_{2s} \frac{R_{22}R_{24}}{R_{24} + R_{23}} \left( \frac{1}{R_{21}} + \frac{1}{R_{22}} \right) - V_{out1} \frac{R_{22}}{R_{21}}$$

其中， $V_{out2}$  为 U2 输出

将两个模块合在一起，可得：

$$V_{out2} = V_1 \frac{R_{12}}{R_T + R_{11}} \frac{R_{22}}{R_{21}} + V_{2s} \frac{R_{22}R_{24}}{R_{24} + R_{23}} \left( \frac{1}{R_{21}} + \frac{1}{R_{22}} \right)$$

为了方便后续的电路设计，我们希望将温度与模块输出电压有此关系：温度=1，电压=0.1V；温度=55，电压=5.5V，以此类推。利用 Python 库 SciPy 对中间温度（16℃至 55℃）进行曲线拟合，可得：

$$V_{out2} = \frac{61.956}{R_T + 5.195} - 1.599$$

利用上述推导的方程可获得图 2 中的电压与电阻参数。

表 1 全范围温度仿真数据

温度/℃	电阻值/kΩ	输出电压(仿真)/V
-18	62.9	-0.657
-8	39.2	-0.170
5	22.1	0.707
16	14.1	1.650
25	10.0	2.519
33	7.46	3.339
46	4.76	4.671
55	3.55	5.533
66	2.52	6.483
80	1.68	7.465

从表 1 易见高低温阶段误差不在要求范围内，因此接下来我们需要设计校正电路解决问题。

### 1.3.2 线性校正电路

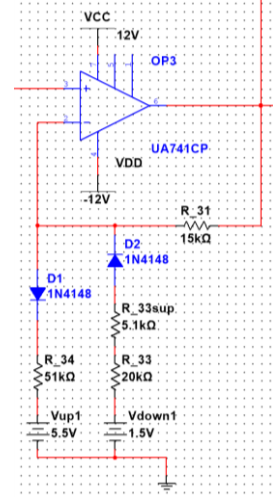


图 3 线性校正电路

通过推导和仿真，可得知此模块的输入输出关系：

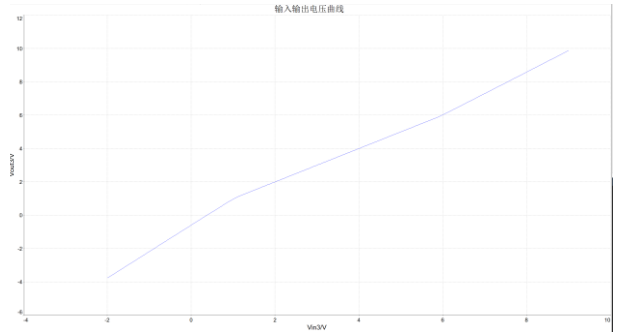


图 4 线性校正曲线

低温部分，当  $V_{in3} < V_{down} - V_{diode}$ ：

$$V_{out3} = V_{in3} \left( \frac{R_{31}}{R_{33}} + 1 \right) + (V_{diode} - V_{down}) \left( \frac{R_{31}}{R_{33}} \right)$$

中温部分，当  $V_{down} - V_{diode} < V_{in3} < V_{up} + V_{diode}$ ：

$$V_{out3} = V_{in3}$$

高温部分，当  $V_{in3} > V_{up} + V_{diode}$ ：

$$V_{out3} = V_{in3} \left( \frac{R_{31}}{R_{34}} + 1 \right) + (-V_{diode} - V_{up}) \left( \frac{R_{31}}{R_{34}} \right)$$

其中， $V_{in3}$  为运放  $V_+$ ， $V_{out3}$  为模块输出

可知在高低温部分可进行不同的线性变换，因此只需对上一个模块的输出电压与理想输出电压进行线性拟合，获取直线参数，即可推出所需的电阻值。

表 2 校正后全范围温度仿真数据

温度/°C	电阻值/kΩ	校正后输出电压 (仿真)/V
-18	62.9	-1.632
-8	39.2	-0.859
5	22.1	0.522
16	14.1	1.651
25	10.0	2.520
33	7.46	3.340
46	4.76	4.672
55	3.55	5.535
66	2.52	6.633
80	1.68	7.896

从表 2 的仿真结果可以发现校正后大部分输出电压都有在误差范围内。-18°C与 80°C 仍不在误差范围内,因为图 3 中所使用得电阻值存在些误差(为了配合所提供的电阻),这部分待进行实验时再根据实际情况进行微调,或者使用可调电阻。此外,选取电阻时,我们假定二极管的电压为约 0.5V,若实验中有差距,可以根据方程调整电阻,并需保证 $V_{up}$ ,  $V_{down}$ 满足下列不等式:

$$\begin{aligned} 0.5 \leq V_{down} - V_{diode} \leq 1.6 \\ 5.5 \leq V_{up} + V_{diode} \leq 6.6 \end{aligned}$$

### 1.3.3 精密整流电路

本实验的精密整流电路设计是为了将负温度值予以显示,所以我们所需做的是实现一个标志着负号的电路,以及表示数值部分大小的电路。对于后者,我们采取精密全波整流器,或者将其称为绝对值电路。我们采取的电路如下所示:

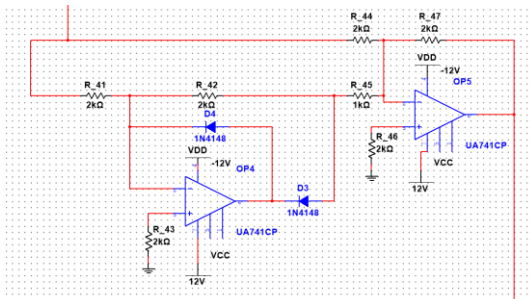


图 5 精密整流电路

如图 5,左上的引线为输入端,最右侧的引线为输出端。该电路的设计由两部分组成:反相半波整流器与加权求和运算放大器。图中两个二极管连

至输出端的运放就是反相半波整流器,它的电压增益为 1,当输入信号为正的时候,D4 处于关断,而 D3 构成反馈环路,将正输入信号进行反相。同理,当输入信号为负时,导致输出端的 D3 关断,D4 导通,且由于 Non-inverting input 是连接至 GND 的,运放处于非饱和状态。对于幅输入信号,电路输出为 0,因此实现了反相半波整流。可以注意到在上图的电路设计中,除了一个电阻为 1K 欧姆外,其余均为 2K 欧姆,这边的 1K 欧姆是为了将第二个运放的电压增益设为-2 (inverting configuration),并使前序进行反相半波整流的输出进行权重为 2 的加权后与原输入信号相加完成总体的精密整流。因此第二个运放的功能为加权求和放大器。假如我们以正弦信号作为输入,则上述解释转化为图像形式为:

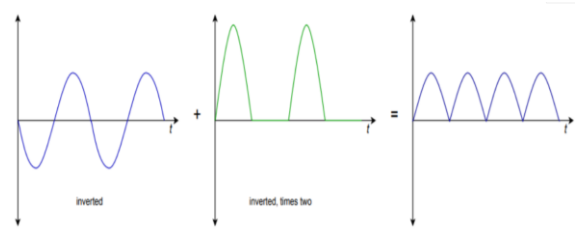


图 6 精密整流图像解释

用于显示负号的电路设计较为简单,只需采取一个发光二极管 LED,如下图所示:

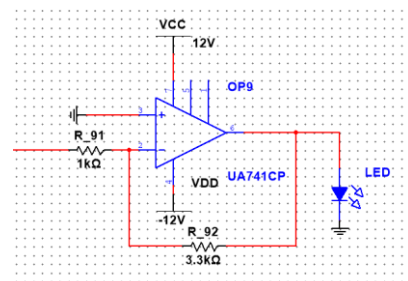


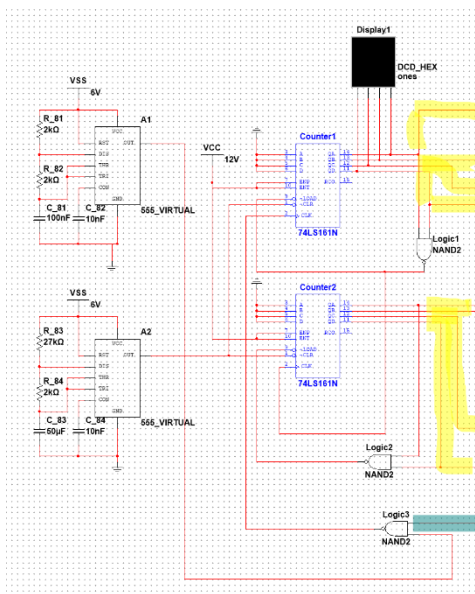
图 7 LED 显示电路

当输入信号为负时,只要调节反馈回路上的电阻使需显示的最小负电压输入满足 LED 的点亮条件以及最大负电压输入不会击穿 LED 即可。

### 1.3.4 模数转换电路

由于需要显示的度数范围为-18°C~ 80°C,因此我们只需实现一个模大于 80 的计数器即可,符号由 LED 进行显示,我们采用的是模为 100 的计数器,计数范围从 0 至 99。计数器的设计采用两片同步加法计数器 74HC161 进行级联,配合与非门 CD4011

由于 74HC161 为 16 进制的计数器, 为了实现模为 100 的计数器, 需将其改为 10 进制的计数模式, 将两片 74HC161 进行级联, 并使低位向高位进位 6,



也正由于我们采用的是模 100 的计数器, 因此两片 74HC161 都应该为 10 进制, 因此两片 IC 在连线上具有高相似性。

具体连线如上图所示，由于实验室中采用的为 5V 的 74HC161 芯片，在 Multisim 中并未有 5V 的 74HC161，便采用了 5V 的 74LS161N 在仿真时替代，其在功能及连线上与 74HC161 一致。上图中带有黄色荧光的为连接至电阻梯形网络的连线，带有绿色荧光的为电平匹配及整形电路的输出连线。

两片计数器均采用了置数法改为 10 进制计数，具体实现为使  $Q_D Q_C Q_B Q_A$  在计数至 1000 时直接跳转到 1111 以省去中间冗余的 6 个无用状态实现 10 进制计数。通过以 0000 作为起始状态，当第十个时钟上升沿到达时产生 LOAD 置数信号将计数器置为 0000 即可。

两片 74HC161 最大的差异就在与时钟信号 CLK 与复位信号 CLR 的连线上, 我们采用了两个 555 定时器产生两个不同频率的方波, 其中 A1 的实测频率为 2.5kHz, A2 的实测频率为 937.8mHz。

低位计数芯片 U3 的时钟信号是 A1 产生的时钟信号与整形电路输出的与非输出结果，这是因为比较器的输出结果需要与时钟上升沿结合判断计数器在何时该停止计数，所以这也采用了较快的时钟频

从图中可以看出对于两片计数器芯片，他们的复位信号 **CLR** 均接至 **A2**，这是为了实时刷新任何温度的变化，但为了计数及显示效果，这个时钟信号也不能太快，防止在计数结束前就进行了刷新以及防止刷新过快显示效果不佳，因此将此时钟信号控制在 **1Hz** 左右。

完成计数电路后，下一步就是设计数模转换电路，以将计数器的状态转换为模拟电压，并与待测电压进行比较，完成数模转换的功能。实验中我们选择的是梯形电路。

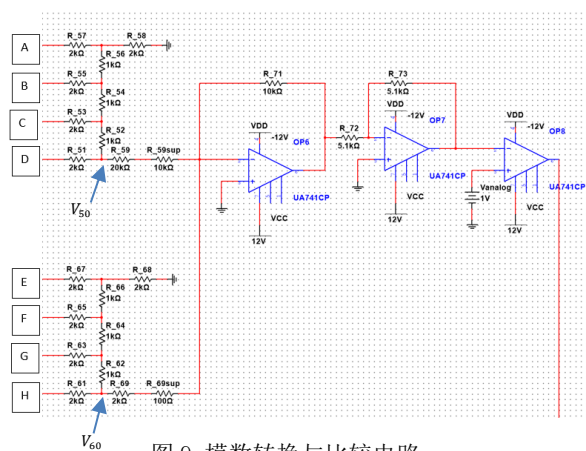


图 9 模数转换与比较电路

为了计算 OP6 的输入电压, 可利用叠加原理, 将电阻网络的输出电压分为左上角 ( $V_{50}$ ) 与左下角 ( $V_{60}$ ) 两部分。其中, 左上角部分对应计算个位数的 Counter1, 左下角对应计算十位数的 Counter2。  $V_{50}$ ,  $V_{60}$  表达式分别为:

$$V_{5o} = \frac{R_{59}}{R_{59} + R} \left( \frac{V_A}{16} + \frac{V_B}{8} + \frac{V_C}{4} + \frac{V_D}{2} \right)$$

$$V_{6o} = \frac{R_{69}}{R_{69} + R} \left( \frac{V_E}{16} + \frac{V_F}{8} + \frac{V_G}{4} + \frac{V_H}{2} \right)$$

其中,

$$\begin{aligned} R &= R_{52} = R_{54} = R_{56} = R_{62} = R_{64} = R_{66} \\ 2R &= R_{51} = R_{53} = R_{55} = R_{57} = R_{58} = R_{61} = R_{63} \\ &= R_{65} = R_{67} = R_{68} \end{aligned}$$

进一步推导, 可得数模转换结果 (OP7 输出):

$$V_{out7} = \frac{R_{71} R_{73}}{R_{72}} \left( \frac{V_{50}}{R_{59}} + \frac{V_{60}}{R_{69}} \right)$$



从上述推导可得知 VA 与 VE 分别为个位与十位数计数器的最低比特（LSB）输出电压。已知设计中，1V 对应 10°C，令：

$$\frac{R_{71}}{R_{59} + R} \frac{V_A}{16} = 0.1$$

$$\frac{R_{71}}{R_{69} + R} \frac{V_E}{16} = 1$$

这时，根据计数器输出的高低电平就能够推导出所需的电阻值。完成数模转换部分后，与要将计数器状态对应的电压与待测电压进行比较，实验中使用运放 OP8 完成此功能。令 V<sub>analog</sub> 为模拟电路部分所获取的电压，比较器的运作如下：

- 若 V<sub>analog</sub> 大于 V<sub>dac</sub>，运放输出为正计数器会收到有效的时钟信号，并进行计数。
- 若 V<sub>analog</sub> 小于等于 V<sub>dac</sub>，运放输出为零或负，计数器不会收到有效时钟信号，计数停止，这时数码管显示的值即为当前测量温度。

### 1.3.5 电平匹配与整形电路

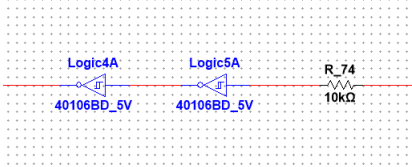


图 10 电平匹配与整形电路

由于 OP8 运放比较器的输出信号幅度介于 -10 至 10V，与数字电路的输入电平要求不符，因此需要进行电平匹配。另外，由于匹配后的输出波形前后沿不陡，当电压处在数字电路的开关电压之间或附近时，其噪声容限低，可能会在输出端出现毛刺脉冲，因此需增加整形电路。这部分电路可以使用电阻和两个反相器实现<sup>[1]</sup>。

### 1.3.6 分压电路与单位增益缓冲器（线上实验）

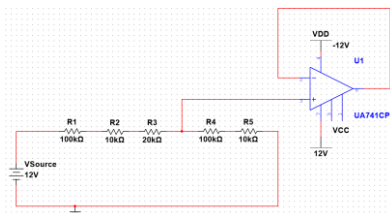


图 11 分压电路与单位增益缓冲器

因线上实验缺乏电压源，需要使用元器件获取实验需要的 5.5V，使用的电路如图 11 所示。由于分压电路的输出电压会根据负载网络的阻抗变化，因此为了使输出电压稳定，需要使用高输入阻抗的单位增益缓冲器（电路中的运放部分）。不难推导出，运放输出为：

$$V_{out} = \frac{R_4 + R_5}{R_1 + R_2 + R_3} V_{source}$$

### 1.3.7 Arduino 模数转换与显示电路（线上实验）

```
#include <SevSeg.h>
SevSeg sevsegDisplay;

void setup() {
  // put your setup code here, to run once:
  byte numDigits=4;
  byte digitPins[]={2,3,4,5};
  byte segmentPins[]={6,7,8,9,10,11,12,13};
  bool resistorsOnSegments=false;
  byte hardwareConfig=CONFIG_ASKODE;
  bool updateWithDelays=false;
  bool leadingZeros=false;
  bool disableDecPoint=false;
  sevsegDisplay.begin(hardwareConfig, numDigits, digitPins, segmentPins, resistorsOnSegments,
    updateWithDelays, leadingZeros, disableDecPoint);
}

void loop() {
  // put your main code here, to run repeatedly:
  int sensorValue=analogRead(A0);
  float voltage=sensorValue*(5.0/1024.0);
  float temperature=voltage*20.0;

  sevsegDisplay.setNumberF(temperature,1);
  sevsegDisplay.refreshDisplay();
}
```

图 12 Arduino 代码

Arduino 单片机的模数转换非常直接，只需使用 analogRead 函数。对于提高要求部分由于 Arduino 可接受的最大输出为 5V，故需要对精密整流电路的输出使用图 13 电路进行缩小（除以 2）。

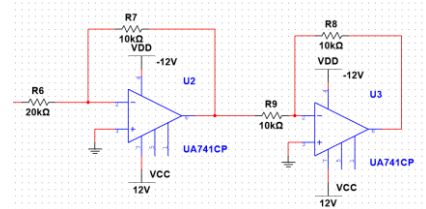


图 13 电压缩小电路

完成模数转换后，将电压乘以 20 即为当下温度，接着利用图 14 电路显示结果即可。

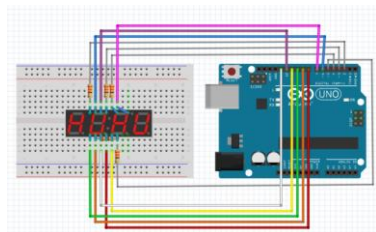


图 14 显示电路

来源：

<https://www.instructables.com/Using-a-4-digit-7-segment-display-with-arduino/>

需要提及的是，在七段数码管的显示中利用了开源的库 <https://github.com/DeanIsMe/SevSeg>。另一方面，已知 Arduino 模数转换的精度为 10bit，通过计算：

$$\frac{5V}{1024} = 0.004883V$$

即每一增量对应0.004883V，经过映射（乘以 20），可得每一增量对应约0.098℃，因此使用 Arduino 理论上可以达成提高精度的要求。

2 实验数据整理与分析

（附：由于线上线下组队关系，各自都完成了实际的电路搭建，以下部分内容均分为线下和线上）

2.1 线下-曾令安

表 3 中的实际输出温度为数码管的显示值，输出电压为示波器对精密整流后输出的直接测量值并结合 LED 的工作状态加上了负号以便于仿真值比较。从表格中对比可以看出，温度计除了在-18℃时出现了 4℃的偏差,其余的点均得到了准确的或偏差为 1℃的值。

可以看出其实我们的电路在经过精密整流后的电压输出已与仿真时的非常接近，但在实际数码管上显示是却在-18℃出现了较大的偏差。这是因为在数码管能显示目前温度前，电压还需经过模数转换模块，而在那里由于电阻的误差，以及计数器芯片实际的高电平输出电压（4.6V）等因素造成了数码管的显示温度出现了偏差。

表 3 线下实验数据

电阻值 /kΩ	温度 /℃	实际输出 /℃	校正后输出 电压 (仿真) /V	实际输出 电压/V
62.9	-18	-14	-1.63	-1.59
39.2	-8	-7	-0.86	-0.89
22.1	5	5	0.52	0.81
14.1	16	15	1.65	1.76
10.0	25	24	2.52	2.59
7.46	33	32	3.34	3.47
4.76	46	47	4.67	4.76
3.55	55	55	5.54	5.53
2.52	66	67	6.63	6.53
1.68	80	80	7.90	7.85

2.2 线上-林宝诚

由于本身拥有的电阻与实验室提供的有些不同（附录 A 中的阻值是根据实验的电阻设计），因此需对阻值做了些许改动，如表 4 所示。另外，线上实验中模数转换电路与温度值显示电路是以 Arduino 单片机完成。进行实验时，首先使用电阻电压转换电路和 Arduino 对温度及基本要求进行实现，结果记录在表 5。其中，实际输出为 Arduino 所控制的数码管数值，实际输出电压为电阻电路转换网络的输出。易见此部分的结果满足了设计要求。

再者，为了完成提高要求部分，需加入线性校正、LED 显示与精密整流电路，实验结果列在表 6 中。表 6 中，实际输出温度的定义不变，实际输出电压为校正电路的输出电压。从结果中不难发现实际输出电压误差较大，但是较奇怪的是，测量出的温度大部分是在误差范围内的，这就表明了校正电路和精密整流电路都有些问题。经过仔细的检查后，发现在中温部分，线性校正电路会对输出添加一个约 0.2V~0.3V 的偏置，而精密整流电路造成的偏置则还未发现明显规律。因此，这部分的电路仍需要一些工作来完善，遗憾的是无法如期完成提高要求的部分。

表 4 线上实验电阻改动

电阻	原电阻值/kΩ	新电阻值/kΩ
R <sub>21</sub>	6.8	5.1
R <sub>22</sub>	35.2	26.8
R <sub>23</sub>	68	100
R <sub>24</sub>	1.5	2.2
R <sub>31</sub>	15	15.1
R <sub>33</sub>	25.1	23.3

表 5 线上实验数据（基本要求）

电阻值 /kΩ	温度 /℃	实际输出 /℃	输出电压 (仿真) /V	实际输出 电压/V
14.1	16	15	1.68	1.58
10.0	25	24	2.56	2.44
7.46	33	32	3.39	3.25
4.76	46	46	4.75	4.61
3.55	55	55	5.62	5.45

表 6 线上实验数据（提高要求）

电阻值 /k $\Omega$	温度 /°C	实际输出 /°C	校正后输出 电压 (仿真) /V	校正后实际输出 电压/V
62.9	-18	-17	-1.72	-1.71
39.2	-8	-9	-0.91	-0.96
22.1	5	4	0.54	0.45
14.1	16	15	1.68	1.79
10.0	25	24	2.56	2.84
7.46	33	32	3.39	3.65
4.76	46	46	4.75	4.91
3.55	55	55	5.62	5.80
2.52	66	66	6.77	6.84
1.68	80	69	8.05	7.67

### 3 实验总结

（附：由于线上线下组队关系，各自都完成了实际的电路搭建，以下部分内容均分为线下和线上）

#### 3.1 线下-曾令安

本次实验虽然是组队完成，但由于是线上线下组队关系，实际上我们两位都是各自对我们所设计的电路进行搭建。因为组队时较为匆忙，再加上线上线下合作，我们在设计电路时先对电路的模块进行了划分，便分头对电路进行设计以及调试等等，并且实时通过腾讯会议对所遇到的困难等等进行讨论。

因为我们都需要独自对电路进行接线，所以在所有模块设计完时，我们都对电路图上的走线进行了较为工整的排布，以加快实际在面包板上进行插接时的进度。

尽管我们在仿真时可以看见温度计对每个范围内都输出正确的值，但当我对电路进行实际的搭建后才发现理论与实操存在重大的差异。在仿真时虽然都可通过采取实际型号的元器件，如 1N4148 和 uA741 等等对电路进行设计，但是电阻的实际阻值却是一个非常随机化的变量。实际电阻阻值于理想化的电阻阻值所产生的偏差也是我在电路搭建完毕后的调试过程中遇到的较为棘手的问题。

在电路搭建时所遇到较为烦人的问题在于无法确定所拿到的芯片是好与坏。我都经历了 uA741 和 74HC161 损坏的情况。因为对运放的电路比较熟悉，

我通过搭建一个十分简单且确定其输出特性的电路对 uA741 是否损坏进行一个判断，并且在使用每个 uA741 前都通过此电路进行检查。但对于 74HC161 我并不清楚该如何对他的好坏进行判断，因此在搭计数电路时因为芯片的问题导致我把整体电路推翻重搭浪费了时间。

而电阻精度带来的影响主要体现于模数转换电路模块当中。在前续的电路搭建完毕后，我均使用了示波器对每个模块的输出进行检查，实际的输出结果也与仿真时的结果几乎一致。因此问题很快就锁定在了模数转换电路上。

由于电阻的阻值以及计数器芯片的输出偏差，按照所设计的电路原样搭建后，为了方便调试，我使用了可调电压源模拟前序电路的电压输入观察数码管上的输出变化。当我的输入逐渐增大时，数码管上的值并没有按照正常情况下那样逐一增大，而时出现了跳过某些值的情况。例如，如下诡异的计数变化 11,12,13,14,15,19,21,22,...等情况发生。而在完成搭建对电路进行检查时，我清楚计数器的连线并没有问题，可以完成 0-99 的正常计数，产生问题的部分便锁定在电阻梯形网络的部分。并且由于在仿真时我们采用的设计是以计数器芯片的高电平为 5V 进行设计的，在我实际完成电路连接后测量发现，计数器芯片的输出实际上为 4.6V，因此我在发现这个问题后也请线上同学帮我对电路中的电阻参数进行调整，并同时精密整流电路进行搭建以节省时间。当线上同学将新参数告知我时，我对电阻进行更换后，发现数码管上的计数显示已经不会随着电压增大而出现跳过的情况，但仍面临着输入电压与输出显示不匹配的问题。而由于需同步配合的可调节变量实在太多，电阻阻值影响带来的偏差实在难以解决，我便只能再通过可变电阻器对本该正常显示的电压输出进行了拟合，也正为了拟合更大的温度范围，我选择牺牲了-18 度的数据点去拟合其余的数据点，这也是在-18 度的点上出现了 4 度偏差的根本原因。

#### 3.2 线上-林宝诚

由于长期不在校园，我此前并未完成较基础的电子电路实验，因此本次实验对我非常有挑战性（第一次做实验就做较复杂的电子系统）。虽然我可以选择下一学年再选择本门课程，但考虑到实习和组队等因素，我仍旧决定在大二的暑假中完成实验。

毫无疑问，线上完成实验遇到的主要问题就是缺乏实验器材。举个例子，由于我没有电压源，只能使用电池作为替代，因此在完成实验过程中，我需要克服电压不太稳定的情况，尽快完成实验。而且，我也需要自行设计实验所需的电压，较不方便，如为满足实验所需的 5.5V，我使用了电阻网络进行分压，并利用单位增益缓冲器以产生稳定的 5.5V 电压。为了降低实验费用，我所添购的元器件数量有限，因此灵活性相对受限（比如没有足够的可变电阻进行调参）。至于数模转换电路，基于工作量和时间限制（等待元器件花了一些时间）的考虑，我决定使用现有的 Arduino 实现。

事实上，在进行实验之前，我计划使用所设计好的模拟电路部分和 Arduino 完成全范围温度的测量和提高精度的要求。通过计算，可以发现 Arduino 的 10 bit 模数转换可以实现 0.1°C 的要求。但由于电压不稳定，使得 0.1°C 位置的电压值不稳定，因此无法完成提高精度得要求。所幸，我能够完成基本要求的部分，无法如期完成自己的目标固然遗憾，但也算差强人意吧。

总结地说，本次实验帮助我大大加深了对理论知识的理解（由于队友需要独自完成线下实验，当时我负责较多的理论推导部分），同时也训练了我的动手实践和解决问题得能力。即使无法完美地完成实验，但本次实验毋庸置疑令我收获不少！

### 3.3 实验改进

如果想提高精度，需使用积分电路设计模数转换器。考虑如下电路：

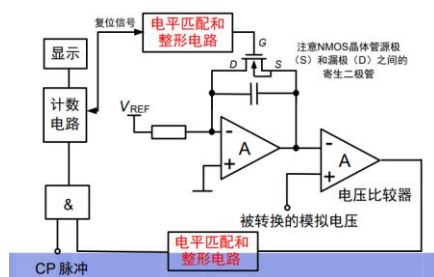


图 15 单斜率积分模数转换电路

首先，假设电容的初始电压为 0，可得：

$$V_{out1} = -\frac{V_{ref}}{RC}t$$

其中， $V_{out1}$  为图 15 左边运放输出电压。

根据比较器的特性，不难发现当  $V_{out1}$  增加至大

于等于给定的模拟电压时，计数器会停止，最终的数值会显示在数码管上。假设给定的模拟电压为  $V_{in}$ ，则：

$$T = -\frac{RC}{V_{ref}}V_{in}$$

即计数器的运行时间与输入模拟电压成正比。将温度与时间互相映射可得：

$$\text{温度} = -\frac{fRC}{V_{ref}}V_{in}$$

其中  $f$  为计数器的时钟频率。

注意到电路中含有一个晶体管，此 MOS 管的用途是为了将电容的两端电压恢复为 0（使用与计数电路相同的复位信号控制），为进行下一个模数转换的操作做准备，以满足自动刷新温度的要求。

进一步改进上述电路，考虑如下设计：

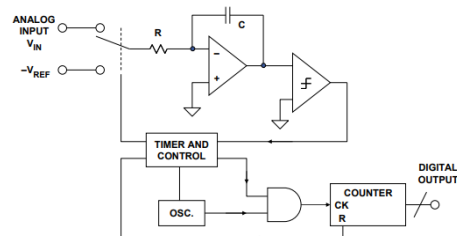


图 16 双斜率积分模数转换电路

首先，运放的输入会与待测模拟电压端  $V_{in}$  相连接，此过程持续  $T_1$  时间：

$$V_1 = -\frac{T_1}{RC}V_{in}$$

由于  $V_{in}$  为正，因此  $V_{out}$  会朝负方向进行积分。紧接着，运放输入端会接到  $V_{ref}$  端：

$$V_2 = -\frac{T_2}{RC}V_{ref} - \frac{T_1}{RC}V_{in}$$

其中  $T_2$  为电压从  $V_1$  变化至  $V_2$  所需的时间。令  $V_{ref}$  为负电压，这时输出电压会往正方向积分。根据比较器特性，当输出电压从  $V_1$  增加至 0 时，计数器会停止，模数转换完成。将  $V_2=0$  代入方程可得：

$$T_2 = -\frac{T_1}{V_{ref}}V_{in}$$

这里  $T_1$  通常为计数器从 0 到溢出所需的时间，即  $\frac{2^N}{f}$ ， $N$  为计数器的比特数。考虑  $T_2$  与温度的映射



可得：

$$\text{温度} = -\frac{2^N}{V_{ref}} V_{in}$$

对比两个积分电路可发现使用双斜率积分电路时，我们不需要关心电阻电容的值，因此也得以避免电阻电容的在环境变化下产生的误差。

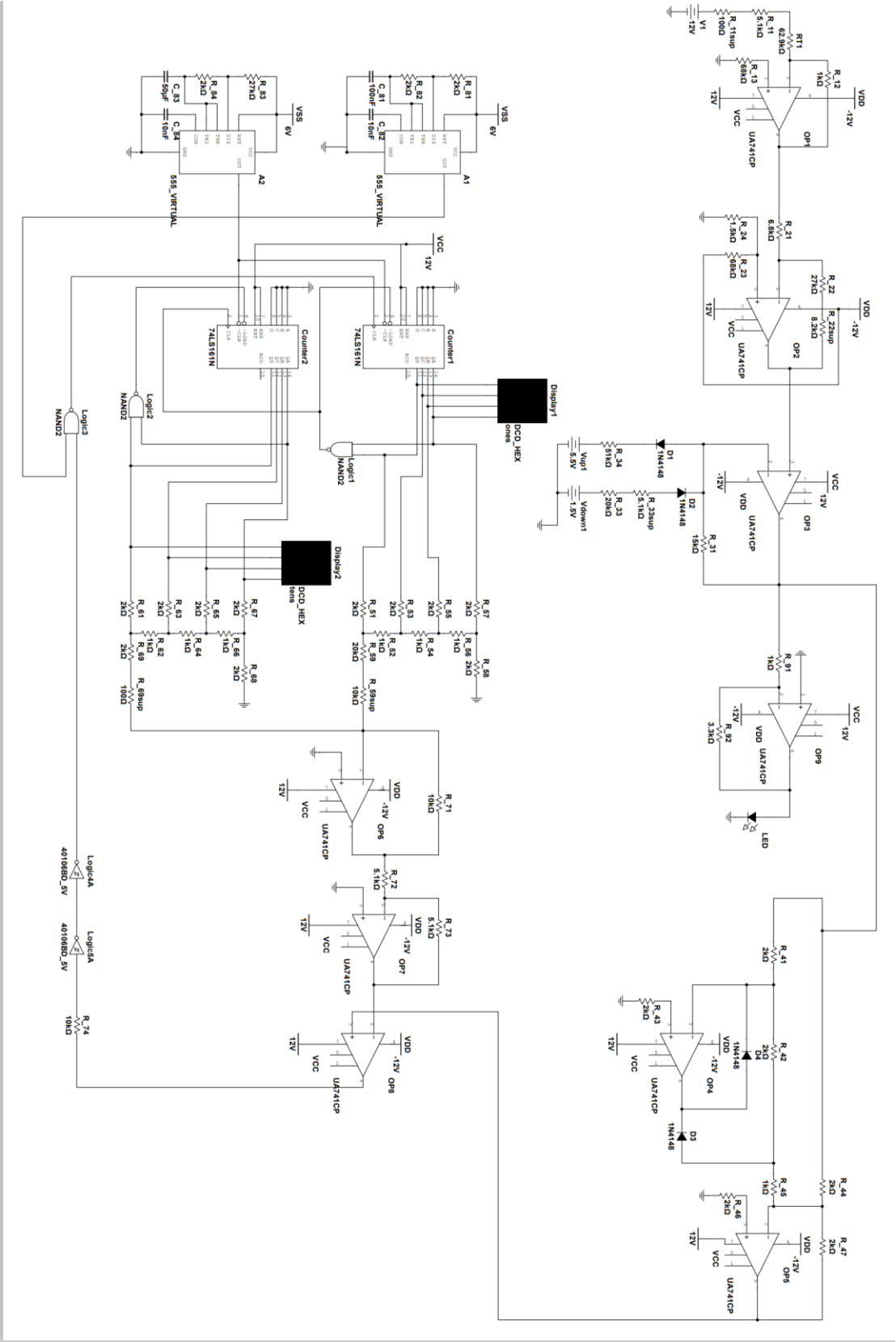
将积分电路与我们设计实现的电路相比。可以发现前者的元器件较少，因此更容易搭建和调试。

#### 参考文献：

[1] 徐淑正, 孙忆南. 电子电路课程设计[R]. 2021.

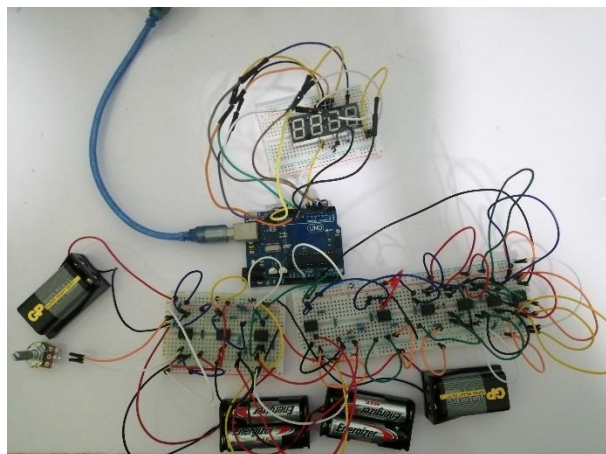
[2] 范有机. 热敏电阻温度传感器的非线性校正[J]. 武夷学院学报, 2008, 027(002):60-62.

附录 A：电路原理图



---

## 附录 A：实际电路（线上）



附录 B: 实验原始数据-曾令安 (线下)

$R/k\Omega$	Temp/ $^{\circ}C$	Actual output/V
62.9	-14	-1.59
39.2	-7	-0.89
22.1	5	0.81
14.1	15	1.76
10	<del>24</del> 24	2.59
7.46	<del>32</del> 32	3.47
4.76	47	4.76
3.35	55	5.53
2.52	67	6.53
1.68	80	7.85

CLK fast: 2.5013 KHz,  $V_{PP}=4.7V$

CLK slow: 937.8 mHz,  $V_{PP}=5.0V$

7.3

注: 线上实验数据已微信发送给孙忆南老师

附录 C：成本分析

表 7 线下实验成本

元器件	数量	单价/元	总价/元
电阻	51	0.10	5.10
电容	4	0.10	0.40
显示	1	20.00	20.00
电源	1	20.00	20.00
ua741	9	1.19	10.71
1N4148	4	0.07	0.28
NE555	2	0.69	1.38
74HC161	2	1.37	2.74
CD40106	1	0.63	0.63
74LS00	1	1.08	1.08

总价：62.32 元

表 8 线上实验成本（马来西亚购买）

元器件	数量	单价/元	总价/元
电阻	38	0.10	3.80
显示	1	20.00	20.00
电源	1	20.00	20.00
lm741	9	1.39	12.51
1N4148	4	0.31	1.24
Arduino Uno	1	163.15	163.15

总价：220.70 元