

本科实验报告

实验设计报告

课程名称： 电子电路设计实验二

姓 名： 周灿松

学 院： 信息与电子工程学院

系：

专 业： 信息工程

学 号： **3190105055**

指导老师： 李锡华、施红军、叶险峰

**2021** 年 **6** 月 **21** 日

专业： 信息工程姓名： 周灿松

学号： 3190105055

日期： 2021 年 6 月 21 日地点： 东 4-216

课程名称： 电子电路设计实验二 指导老师： 李锡华、施红军、叶险峰 成绩：

实验名称： 实验设计报告 实验类型： 设计实验 同组学生姓名： 宋凯

# 一、 实验目的

1. 学习掌握用 Arduino UNO 设计倒计时定时器；
2. 学习掌握 PCB 电路板的设计和制作；
3. 学习掌握 Arduino UNO 扩展板的设计与制作；
4. 学习掌握旋转编码器的使用。

# 二、 实验任务与要求

1. 用 Arduino UNO 设计倒计时定时器，要求如下：设定倒计时时间若干（设定标准时间数组），通过旋转编码器选择，时间到时报警。
2. 设计电路，完成相应器件的选择，制作 Arduino UNO 扩展板。
3. 编制与调试倒计时定时器程序。
4. 将制作的扩展板与 Arduino UNO 板组装后，进行系统联调。

# 三、 实验原理

## Arduino UNO 板

Arduino UNO R3 具有 14 个数字输入/输出引脚，6 个模拟输入/输出。在本次倒计时电路设计中， 我们以 Arduino UNO R3 作为整体电路的控制器，通过不同引脚在不同情况下输出不同电压控制七段数码管以及蜂鸣器，以此实现倒计时功能

1. 七段数码管
   1. 数码管构成

7 段数码管一般由 8 个发光二极管组成，其中由 7 个细长的发光二极管组成数字显示，另外一个圆形的发光二极管显示小数点。

当发光二极管导通时，相应的一个点或一个笔画发光。控制相应的二极管导通，就显示出各种字符，尽管显示的字符形状有些失真，能显示的字符数量也有限，但其控制简单，使有也方便。发光二极管的阳极连在一起的称为共阳极数码管，阴极连在一起的称为共阴极数码管。

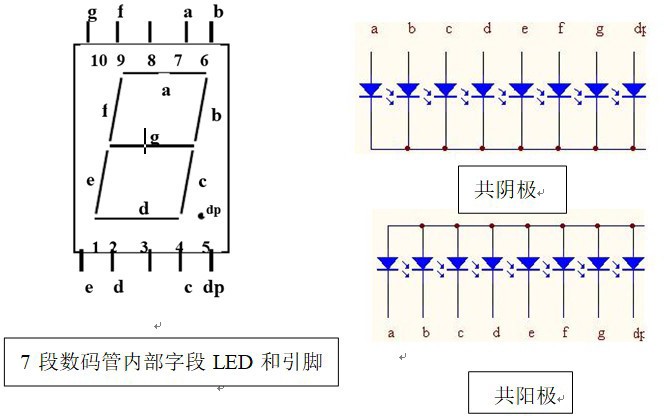


图 1: 七段数码管结构

* 1. 驱动方式

七段数码管具有静态驱动和动态驱动两种方式。

静态驱动方式中每一个发光二极管均有一个输出引脚控制，采用此种驱动方式编程实现更加简单， 数码管亮度也更高，缺点则是它需要的管脚较多，由于 arduino 的输出管脚数目有限，我们不采用此种驱动方式，而是选用动态驱动。

动态驱动则是利用了发光二极管的余辉效应以及人的视觉暂存现象。在此种驱动方式中，我们将所有数码管的同名端连在一起，在同一时刻，施加在所有数码管上的控制信号均相同，但同一时刻只有一个数码管处于导通状态。通过控制 COM 端，我们令所有数码管轮流点亮，当扫描速度足够快时，看上去就像所有数码管同步显示了一样。

1. 蜂鸣器

蜂鸣器按是否有驱动分为有源蜂鸣器和无源蜂鸣器两种。此处的“源”指的是震荡源而非电源，有源蜂鸣器只需要我们施加一个电压给他就可以响，无源蜂鸣器则需要用一个特定频率的方波对他进行驱动。为了设计简单，此次实验我们选用了有源的蜂鸣器。

1. 旋转编码器

旋转编码器是一种数字器件，它有着两个输出 A 和 B，当旋转时，输出 A、B 会有变化，通过变化的不同我们可以判断出旋转编码器是顺时针旋转还是逆时针旋转，如此便可利用旋转编码器控制倒计时的时间

# 四、 电路及程序设计

1. 总体电路图

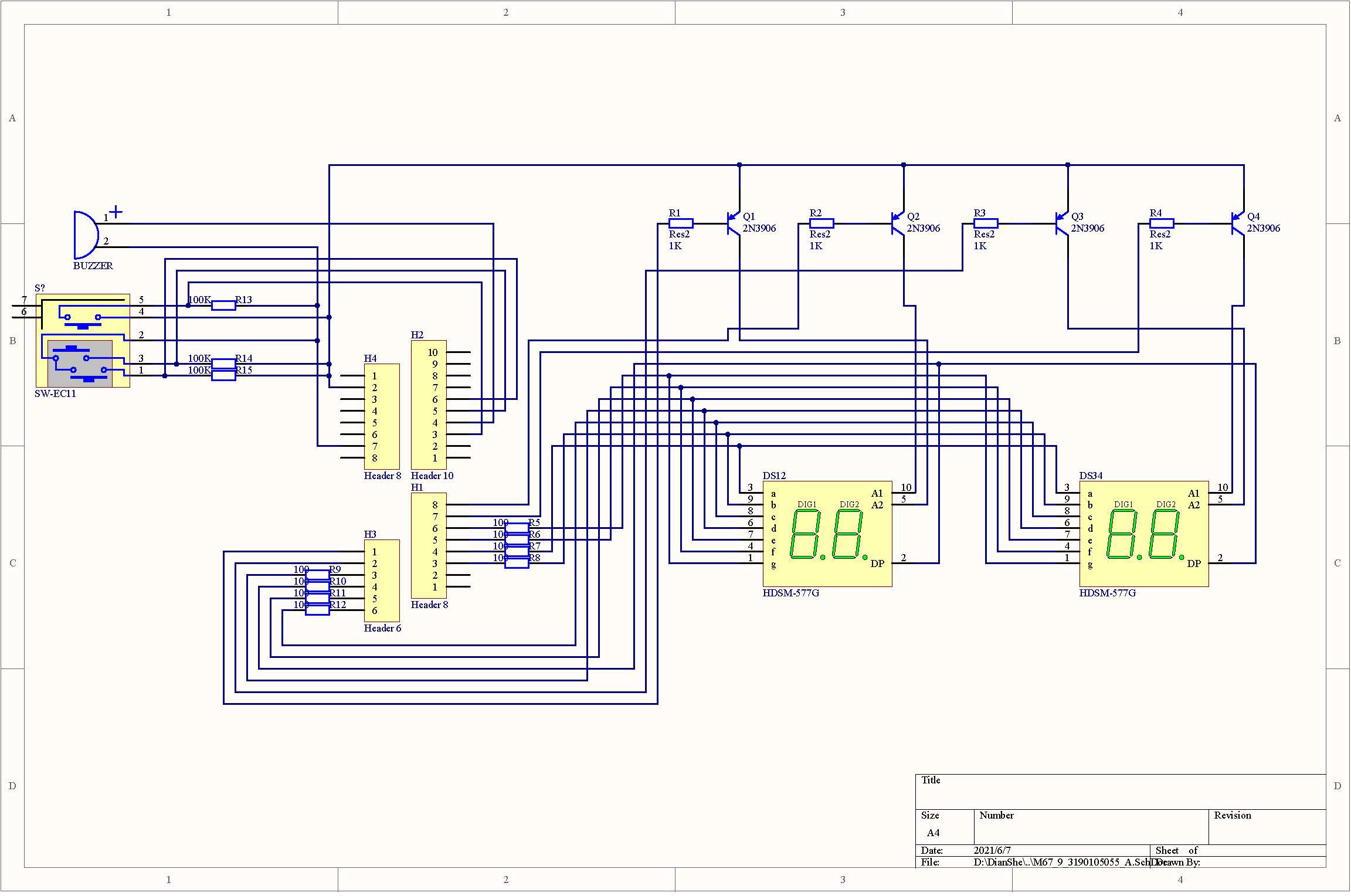


图 2: 原理图

1. 时间显示模块

在本次实验中我们选用了两个两位的七段数码管作为剩余时间显示，DS12 显示分钟，DS34 显示剩余秒数。

在线路连接上，我们将两个数码管的 1-4,6-9 号引脚一一对应连接后接到 Arduino 的输出管脚上， 如此便可将显示信号同时输出到四个七段数码管上。而控制数码管导通与否的 5、10 号引脚，我们分别将其接入四个三极管上，此处三极管作为开关管使用，通过控制三极管轮流导通即可实现四个七段数码管的轮流点亮。同时，在线路与 arduino 连接前，我们先接入了一个电阻作为保护电阻，防止出现问题。

1. 旋转编码器模块

EC11 型旋转编码器有五个引脚，其中 1、3 脚为信号 A、B，4、5 脚为开关，2 脚接地。

在转动旋转编码器时，会产生如下波形，通过观察 A、B 信号变化的过程便可判断出是顺时针还是逆时针旋转。

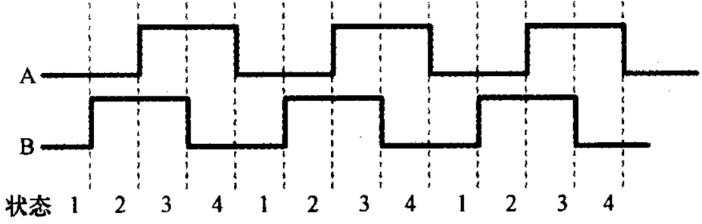


图 3: 旋转编码器波形

代码设计：

// 右旋返回为1 ， 左旋为−1, 不旋为0

int getEncoderTurn()

{

// return −1 ， 0 ， or +1 static int

static int

int result

int newA = int newB =

oldA=LOW;

oldB=LOW;

= 0;

digitalRead(aPin); digitalRead(bPin);

if (newA != oldA || newB != oldB)

{

// something has changed

// 当信号B 从低电平变化到高电平时， 如果A 为低电平， 则说明状态是从左到右变化， 反之则反

if(oldB == LOW && newB == HIGH){

result = oldA\*2 − 1;

}

}

oldA = newA; oldB = newB; return result;

}

1

3

5

7

9

11

13

15

17

19

21

1. 蜂鸣器模块

因为我们采用了有源蜂鸣器，所以只需要将蜂鸣器加在一个输出引脚上，控制该引脚的输出电压， 即可控制蜂鸣器的启动与否。

# 五、 主要仪器设备

Arduino UNO R3，EC11 型旋转编码器，两个两位七段数码管，四个三极管，电阻若干

六、 **PCB** 版图绘制

1. 绘制步骤
2. 绘制原理图并编译
3. 创建 PCB 文件，并将原理图同步至 PCB 图中
4. 放置各个元件，此步需要注意引脚位置的放置，不然后续可能无法与 Arduino 板连接
5. 选择层级为信号层，隐藏无关元素进行手动布线

## PCB 图

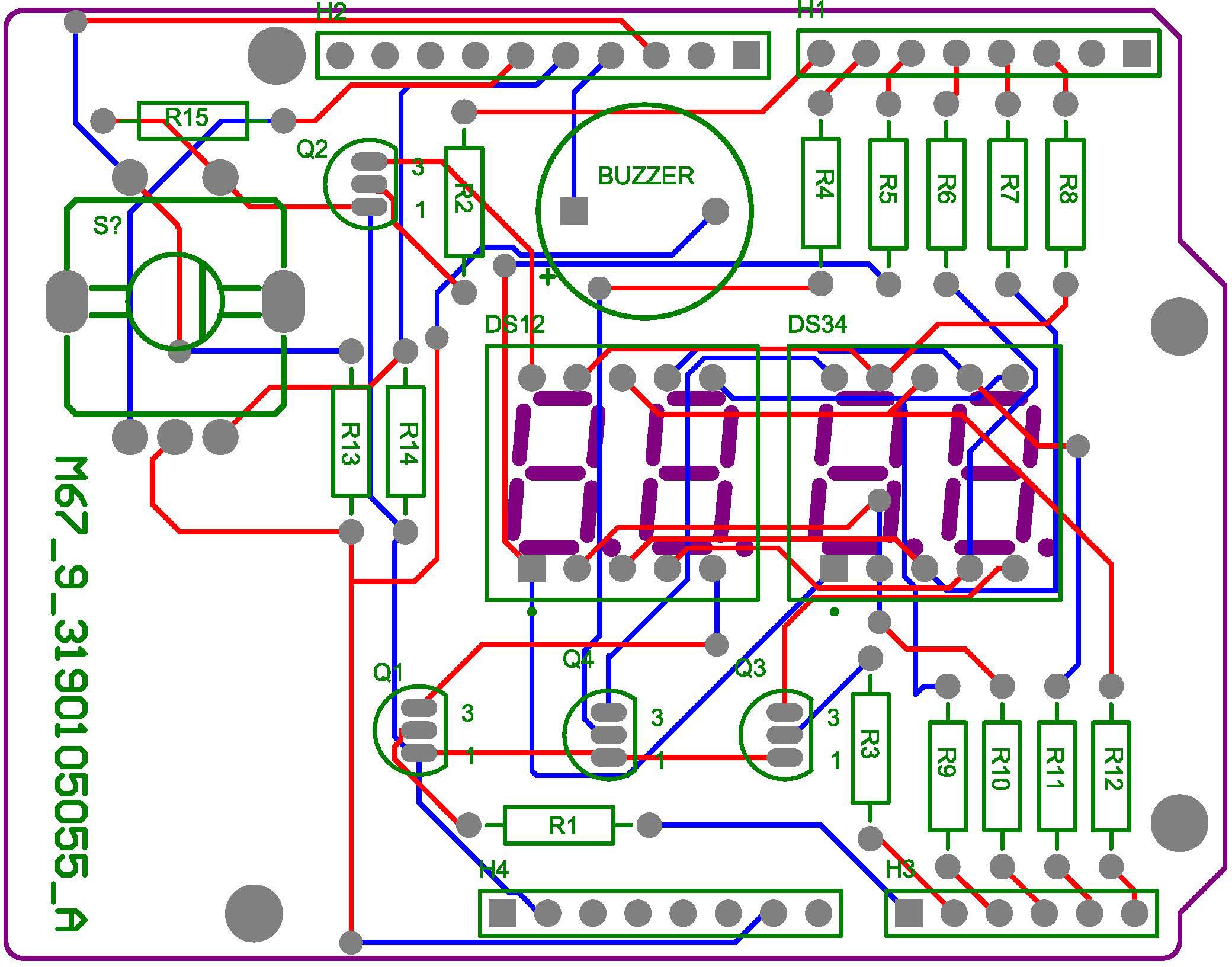


图 4: PCB 板图

# 七、 调试

1. 焊接

按照电路设计选取元件，按高度从低到高的原则将其焊接到 PCB 板上。



图 5: 焊接成果

1. 调试
   1. 将代码拷入 Arduino 之中, 数码显示如下：



图 6: 能正常显示数字

* 1. 按以下步骤测试功能：

1. 按下旋转编码器，测试是否能正常倒计时以及到时间能否报警
2. 拧动旋转编码器，测试能否正确改变设定时间
3. 设定一时间，将 Arduino 断电，然后上电测试其能否记住上次设定时间功能均正常，不足之处见下文
   1. 调试中的问题及解决方案
      1. 问题

七段数码管出现闪烁，时间切换不够流畅

* + 1. 解决方案

修改代码中以下部分，修改方案见注释

void process()

{

// 将此循环中的循环次数由100 调小， 寻找合适次数， 大概在50 左右比较合适

for (int i = 0; i < 100; i++)

{

int change = getEncoderTurn(); if (stopped)

{

changeSetTime(change);

}

else

{

updateCountingTime();

}

}

if (timerMinute == 0 && timerSecond == 0)

{

digitalWrite(buzzerPin, HIGH);

}

}

1

3

5

7

9

11

13

15

17

19

# 八、 思考题

* + - 1. 将图 **4.15** 中的数码管换成共阴的，分别说明电路和程序要做怎样的修改

电路图可以不做修改，保持原有线路不变。

程序需要改动两个地方：第一处将 *byte digits*[10][8] 中的 1 和 0 互换；二则是将 setDigit 函数中

digitalWrite 第二个参数中的不等号换为等号。

* + - 1. 考虑倒计时定时器的应用，例如如何控制定时加热，简要介绍实现的电路。

和控制蜂鸣器时间结束后长鸣原理类似。另寻一个数字输出口，将其与电阻丝一端相连，另一端接地，时间结束后，数字输出口持续输出高电平即可。

* + - 1. 如何通过修改程序，消除秒钟滴答声。

删除 updateCountingTime 函数中以下代码，或者移除 delay(10)：

2

digitalWrite(buzzerPin, HIGH); delay(10); digitalWrite(buzzerPin, LOW);

# 九、 讨论总结、体会

在前期电路设计过程中，其实也遇到了不少的问题，因为以前没有设计过 Arduino 拓展板，所以在最开始拿到题目的时候陷入了迷茫之中，不知道从何而起。在反复读了参考资料之后，才渐渐有了眉目。因为老师的资料中给了电路图，我们设计电路的时候困难较少，但是当进入面包板调试过程后，我们发现事情并不像自己想象中的那么顺利，出现了以下几个问题：

首先，七段数码管和 A4 那个 pdf 中的数码管封装不同。拿到器件之后，我们不知道怎么去接线， 但通过互联网查找到了该型号数码管的器件手册（后面发现老师给了器件手册截图），事情变得顺利起来了。

其次，旋转编码器给我留下了深刻的印象，因为不懂旋转编码器的工作原理，我们就照猫画虎地按照图上接线，最后在测试功能的时候发现拧动编码器无法改变计时的多少。在课后，我将板子连到自己电脑进行串口调试后发现拧动编码器并不会令 A，B 信号改变，所以我大胆推测是线路接错了。在网上学习了旋转编码器的工作原理之后，我发现自己将 1，3 号脚直接接地了，这也就导致了 A，B 信号输入始终为低电平。将其改接高电平之后，成功地解决了问题。

在设计电路并进行面包板调试时，因为电子电路设计实验二是开放实验，因为以前没有做过类似的实验，出现了各种各样的问题。但是在解决问题的过程中，我也很明显地感受到了自己的成长。实践是最能够检验自己所学的途径，在实践的过程中，每解决一个问题都能够获得一股巨大的成就感，这种成就感也能激励着我继续学习下去。同时，在实践中才能够更加透彻地理解各个器件地使用方法，就如数码管的驱动方式，不进行实验始终觉得动态驱动很虚无飘渺，但实验后才会发现这的确能够正确地运行。

调试过程中，除了焊接三极管时需要注意不要将三个电极焊在一起了，焊接总体而言问题不大。在调试过程中，数码管出现了闪烁的情况，极其影响观感，在研究了代码之后，发现是 process 函数中的循环次数过大，但是也不能改的过小，改得太小也会导致数码显示管出现模糊的情况，最终在反复尝试之后将循环次数确定在了 50 次，使得模糊和闪烁之间取得了一个平衡。

在这次的实验设计工作中，我学到的更多的是电路设计的方法，跟着老师给的大致方案，一步一步地将一个能够实现具体功能的电路构建好，并在面包板上实现。虽然其中设计的部分偏少，更多地是跟着老师的方案把东西做出来，但是这个过程也让我管中窥豹，知道了电路设计的大致流程，也在一定程度上培养了对于电路设计的兴趣，以及自己的动手能力。其次，在这个过程中也学会了 AD 软件的基本操作，懂得了利用 AD 软件绘制 PCB 版图的过程。

1 #include <EEPROM.h>

Listing 1: 完整代码

3 // 驱动数码管显示的引脚定义， 0−13 为数字引脚,14−19 为模拟引脚int segmentPins[] = {3, 2, 19, 16, 18, 4, 5, 17};

5 // 用于驱动四个数码管， 控制其开关int displayPins[] = {15,6,14,7};

7 // 时间序列

int times[] = {5, 10, 15, 20, 30, 45, 100, 130, 200, 230, 300, 400, 500,

600, 700, 800, 900, 1000, 1500, 2000,

9 3000

};

11 int numTimes = 20;

byte selectedTimeIndex;

13 int timerMinute; int timerSecond;

15 // 蜂鸣器开关

int buzzerPin = 11;

17 // 旋转编码器三个按钮

int aPin = 12;

19 int bPin = 13;

int buttonPin = 10;

21

boolean stopped = true; // 初始为stopped

23 byte digits[10][8] = {

//a b c d e f g ．

25 {1, 1, 1, 1, 1, 1, 0, 0}, //0

{0, 1, 1, 0, 0, 0, 0, 0}, //1

27 {1, 1, 0, 1, 1, 0, 1, 0}, //2

{1, 1, 1, 1, 0, 0, 1, 0}, //3

29 {0, 1, 1, 0, 0, 1, 1, 0}, //4

{1, 0, 1, 1, 0, 1, 1, 0}, //5

31 {1, 0, 1, 1, 1, 1, 1, 0}, //6

{1, 1, 1, 0, 0, 0, 0, 0}, //7

33

{1, 1, 1, 1, 1, 1, 1, 0}, //8

35 {1, 1, 1, 1, 0, 1, 1, 0} //9

};

37

void setup()

39 {

for (int i = 0; i < 8; i++)

|  |  |  |
| --- | --- | --- |
| 41  43 | {  pinMode(segmentPins[i], OUTPUT);  } | |
| 45 | for (int i = 0; i < 4; i++)  { | |
|  | pinMode(displayPins[i], OUTPUT); | |
| 47 |  | } |
| 49 |  | pinMode(buzzerPin, OUTPUT);  pinMode(buttonPin, INPUT); |
| 51 |  | pinMode(aPin, INPUT);  pinMode(bPin, INPUT); |
| 53 |  | selectedTimeIndex = EEPROM.read(0); // 读取设置的时间  timerMinute = times[selectedTimeIndex] / 100; // 分 |
| 55 |  | timerSecond = times[selectedTimeIndex] % 100; // 秒  //Serial.begin(9600); |
| 57 | } |  |
| 59 | void  { | loop() |
| 61 |  | if (digitalRead(buttonPin)) // 旋转编码器开关按下  { |
| 63 |  | stopped = !stopped; //false  digitalWrite(buzzerPin, LOW); // 蜂鸣器low |
|  |  | while (digitalRead(buttonPin)){}; // 一直写入保留旋转编码器设定的时 |
|  | 间， 除非编码器按钮按下切换到倒计时模式 | |
| 65 |  | EEPROM.write(0, selectedTimeIndex); |
| 67 |  | }  updateDisplay(); // 更新数码管显示的时间 |
| 69 | } |  |
| 71 | void  { | updateDisplay() //mmss |
| 73 |  | int minsecs = timerMinute \* 100 + timerSecond; // 总时间int v = minsecs; |
| 75 |  | for (int i = 0; i < 4; i++)  { |
| 77 |  | int digit = v % 10;  setDigit(i); |
| 79 |  | setSegments(digit);  v = v / 10; |
| 81 |  | process();  } |

83

setDigit(5); // all digits off to prevent uneven illumination

}

void process()

{

for (int i = 0; i < 50; i++) //tweak this number between flicker and blur

85

87

|  |  |  |  |
| --- | --- | --- | --- |
| 89  91 | { | int change = getEncoderTurn(); if (stopped) | |
| 93 |  | {  changeSetTime(change); | |
| 95 |  | }  else | |
| 97 |  | {  updateCountingTime(); | |
| 99 | }  } | | |
| 101  103  105 | if (timerMinute == 0 && timerSecond == 0)  {  digitalWrite(buzzerPin, HIGH);  }  } | | |
| 107 | void changeSetTime(int change)  { | |  |
| 109 | selectedTimeIndex += change;  if (selectedTimeIndex < 0) | |  |
| 111 | {  selectedTimeIndex = numTimes; | |  |
| 113 | }  else if (selectedTimeIndex > numTimes) | |  |
| 115 | {  selectedTimeIndex = 0; | |  |
| 117 | }  timerMinute = times[selectedTimeIndex] | | / 100; |
| 119 | timerSecond = times[selectedTimeIndex]  } | | % 100; |
| 121 | void updateCountingTime() | |  |
| 123 | {  static unsigned long lastMillis; | |  |
|  | unsigned long m = millis(); | |  |

125

if (m > (lastMillis + 1000) && (timerSecond

{

digitalWrite(buzzerPin, HIGH); delay(10); digitalWrite(buzzerPin, LOW); if (timerSecond == 0)

{

timerSecond = 59; timerMinute−−;

}

else

{

timerSecond−−;

}

lastMillis = m;

}

> 0 ||

timerMinute > 0))

}

void setDigit(int digit)

{

for (int i = 0; i < 4; i++)

{

digitalWrite(displayPins[i], (digit != i)); // 写入0 时数码管导通

}

127

129

131

133

135

137

139

141

143

145

147

149

|  |  |  |  |
| --- | --- | --- | --- |
| 151 | } | | |
| 153 | void setSegments(int n) | |  |
| 155 | {  for (int i = 0; i < 8; i++) | |  |
| 157 | {  digitalWrite(segmentPins[i], | | !digits[n][i]); |
|  | } | | |
| 159 | } |  | |
| 161 | // 右 | 旋返回为1 ， 左旋为−1, 不旋为0 | |
| 163 | int  { | getEncoderTurn() | |
| 165 |  | // return −1 ， 0 ， or +1  static int oldA=LOW; | |
| 167 |  | static int oldB=LOW;  int result = 0; | |

|  |  |  |
| --- | --- | --- |
|  | int newA = digitalRead(aPin); | |
| 169 | //Serial.println(newA); | |
|  | int newB = digitalRead(bPin); | |
| 171 | Serial.println(newB); | |
|  | if (newA != oldA || newB != oldB) | |
| 173 | { | |
|  | // something has changed | |
| 175 | if(oldB == LOW && newB == HIGH){ | |
| 177 | result  } | = oldA\*2 − 1; |
| 179  181 | }  oldA = newA; oldB = newB; return result;  } | |