

浙江大学

本科实验报告

课程名称： 电子电路设计实验 II

姓 名： 王若鹏

学 院： 信息与电子工程学院

专 业： 电子科学与技术

学 号： 3170105582

指导教师： 叶险峰 李锡华 施红军

选课时间： 周二第 9,10 节

2019 年 4 月 30 日

浙江大学实验报告

专业: 电子科学与技术
姓名: 王若鹏
学号: 3170105582
日期: 2019.4.30
地点: 东 4-216

课程名称: 电子电路设计实验 II 指导老师: 李锡华 叶险峰 施红军 成绩: _____

实验名称: 多功能数字时钟的设计与制作 实验类型: 设计实验

同组学生姓名: 于颖

一、实验目的和任务

1.1 实验目的

- (1) 学习掌握用 Arduino UNO 设计数字时钟;
- (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 DS1302 时钟芯片和 LCD1602 液晶显示屏的使用。

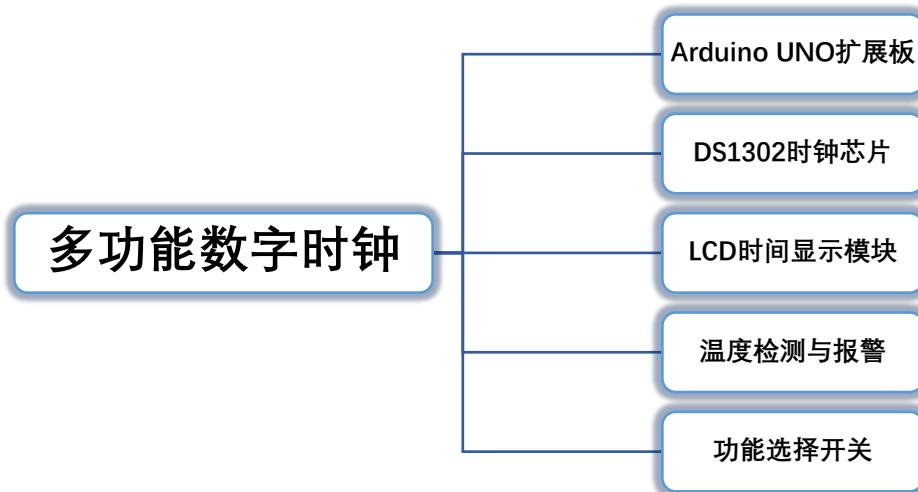
1.2 实验任务

- (1) 根据实验室提供的元器件与实验目的设计合理的电路。
- (2) 根据元器件在面包板上搭建电路。
- (3) 编写Arduino代码驱动电路，并调试。
- (4) 根据原理图设计 PCB 板图，设计出 Arduino UNO 扩展板。

二、实验原理和设计

2.1 设计思路

多功能数字时钟包括五大模块：分别是 Arduino UNO 板、DS1302 时钟芯片、LCD 时间显示模块、温度检测与报警模块、功能选择开关，其模块结构图如下：



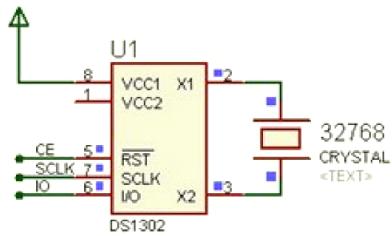
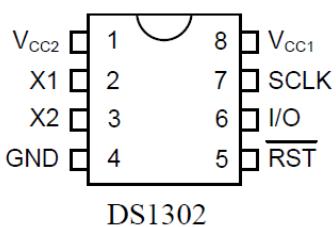
2.2 模块原理与分析

2.2.1 Arduino UNO 扩展板



本实验采用 Arduino UNO R3 扩展板作为开发平台，用于整体运行多功能数字时钟的代码。通过读取信号，实现人机交互。

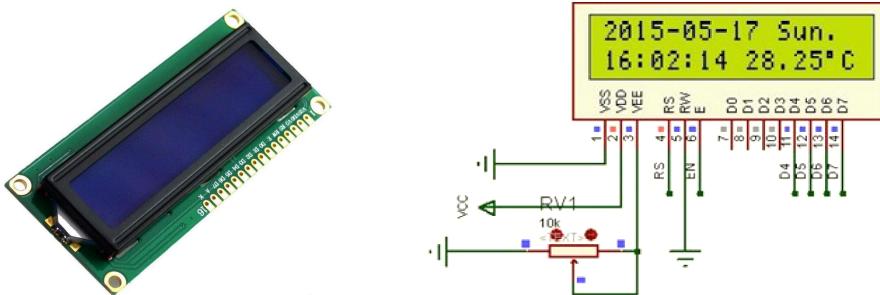
2.2.2 DS1302 时钟芯片



DS1302 芯片引脚如上图所示。1、8 口接电源，2、3 口接 32.768kHz 的晶振，4 口接地，5 口为重置端，6 口为数据输入输出端，7 口产生时钟信号。

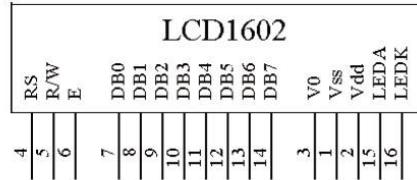
DS1302 是美国 DALLAS 公司推出的一种高性能、低功耗的实时时钟芯片，附加 31 字节静态 RAM，采用 SPI 三线接口与 CPU 进行同步通信，并可采用突发方式一次传送多个字节的时钟信号和 RAM 数据。实时时钟可提供秒、分、时、日、星期、月和年，一个月小与 31 天时可以自动调整，且具有闰年补偿功能。工作电压宽达 2.5~5.5V。采用双电源供电（主电源和备用电源），可设置备用电源充电方式，提供了对后背电源进行涓细电流充电的能力。

2.2.3 LCD 时间显示模块



该模块由 LCD1602 显示屏和一个 10k 的电位器组成。

本次实验使用 Arduino UNO 直接驱动 1602 液晶显示字母和数字，LCD 显示模块有一个优势就是将驱动电路集成在模块中。LCD 显示模块具有标准，不同的生产商所生产的大数模都可以按相同方法来使用。1602 液晶在应用中非常广泛，最初的 1602 液晶使用的是 HD44780 控制器，现在各个厂家的 1602 模块基本上都是采用了与之兼容的 IC，所以特性上基本都是一致的。



如右上图所示，1602 采用标准的 16 脚接口，其中：

第 1 脚：VSS 为地电源。

第 2 脚：VDD 接 5V 正电源。

第 3 脚：V0 为液晶显示器对比度调整端，接正电源时对比度最弱，接地电源时对比度最高，对比度过高时会产生“鬼影”，使用时可以通过一个 10K 的电位器调整对比度。

第 4 脚：RS 为寄存器选择，高电平时选择数据寄存器、低电平时选择指令寄存器。

第 5 脚：R/W 为读写信号线，高电平时进行读操作，低电平时进行写操作。当 RS 和 RW 共同为低电平时可以写入指令或者显示地址，当 RS 为低电平 RW 为高电平时可以读信号，当 RS 为高电平 RW 为低电平时可以写入数据。

第 6 脚：E 端为使能端，当 E 端由高电平跳变成低电平时，液晶模块执行命令。

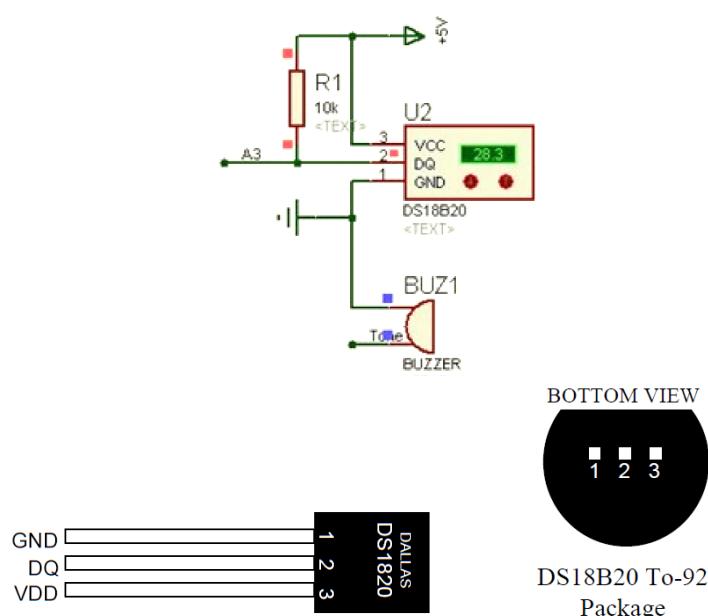
第 7~14 脚：D0~D7 为 8 位双向数据线。

第 15 脚：背光电源正极。

第 16 脚：背光电源负极。

2.2.4 温度检测与报警模块

该模块由 DS18B20 温度传感器、一个 10k 电阻、一个蜂鸣器组成。当温度达到设定值时，蜂鸣器会发出警报，以实现温度检测与报警的功能。



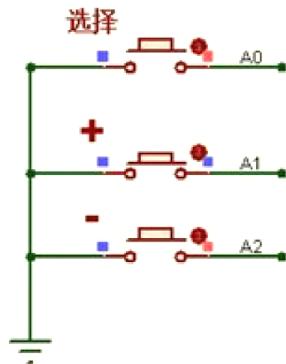
DALLAS 公司生产的 DS18B20 结构如左上图所示。1 脚接地，2 脚为数据输入输出端，3 接电源。右上图为底部视角。超小的体积，超低的硬件开销，抗干扰能力强，精度高，附加功能强，使得 DS18B20 更受欢迎。

DS18B20 的主要特征：

- 全数字温度转换及输出。
- 进先进的单总线数据通信。

- 最高 12 位分辨率，精度可达± 0.5 摄氏度。
- 12 位分辨率时的最大工作周期为 750 毫秒。
- 可选择寄生工作方式。
- 检测温度范围为 -55°C ~+125°C (-67°F ~+257°F)
- 内置 EEPROM，限温报警功能。
- 64 位光刻 ROM，内置产品序列号，方便多机挂接。
- 多样封装形式，适应不同硬件系统。

2.2.5 功能选择开关



本模块由 3 个四角功能选择开关组成。开关一边接地，一边接 Arduino 的不同控制端口，作用分别为“功能选择”“加”“减”，从而实现数字时钟的各种功能。

三、主要仪器设备

装有 Arduino IDE、AD9 的计算机一台、一只万用表、一个面包板、一个电源、一个蜂鸣器、一个 10k 电位器、1 个 10k 电阻、1 个 32768 Hz 晶振、一个 Arduino UNO 扩展板、一个 DS1302 时钟芯片、一个 LCD1602 液晶显示屏、一个 DS18B20 温度传感器、三个选择开关。

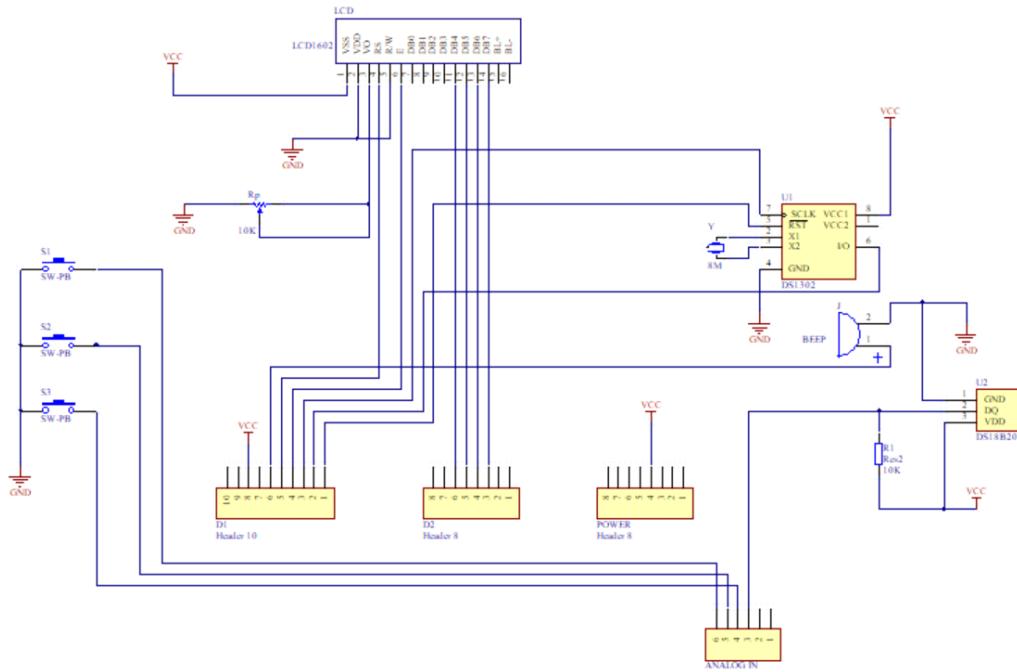
四、实验步骤与内容

4.1 实验过程

- (1) 根据要求设计电路原理图，编写代码；
- (2) 在面包板上搭建电路，进行功能调试；
- (3) 根据原理图设计 PCB 板图；
- (4) 制作 PCB 板图并焊接器件，进行功能测试。

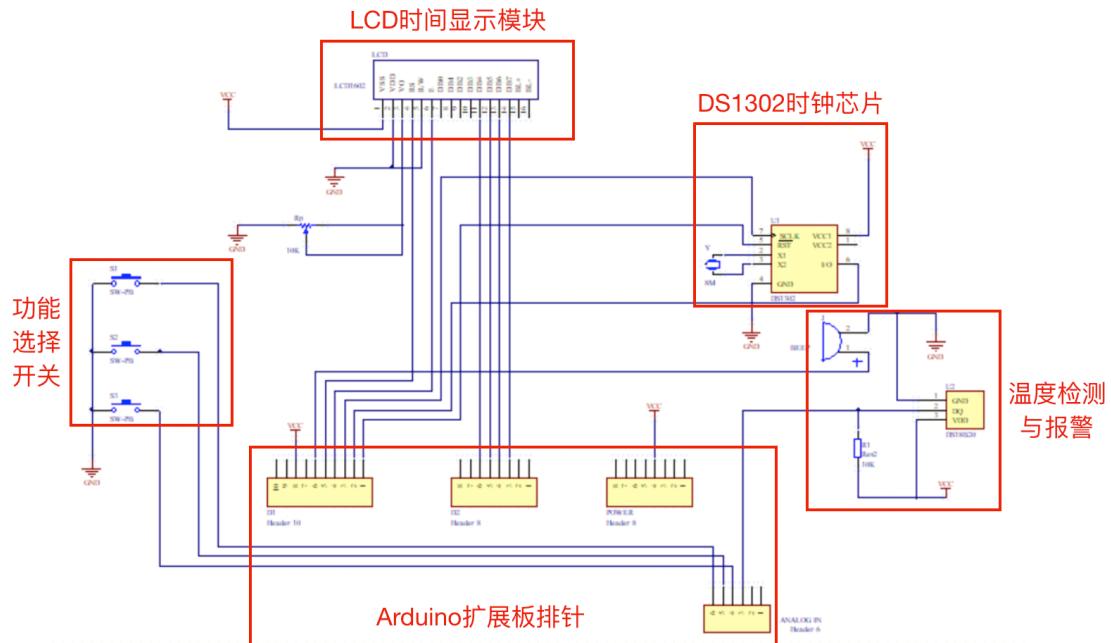
4.2 原理图设计

根据实验设计要求，使用 AD9 软件，将原理图设计如下：



(原理图设计)

为了便于理解，对原理图分模块进行标注，红色方框内为五大模块，如下图：



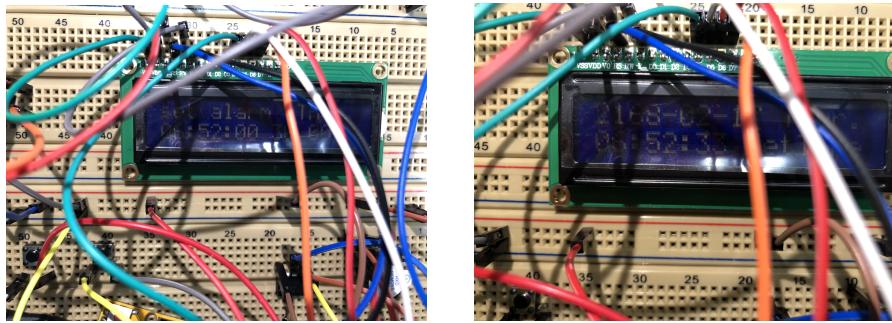
(原理图标注)

4.3 电路搭建与测试

根据原理图，在面包板上搭建电路。同时改进测试代码(详见附录)，将其烧录至Arduino板上，编译成功截图如下：

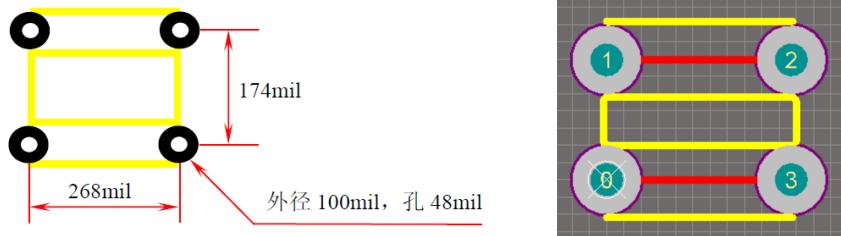


对数字时钟进行功能测试如下，显示屏亮度较暗，经过万用表测量，发现是 Arduino 电压输出不够 5V。

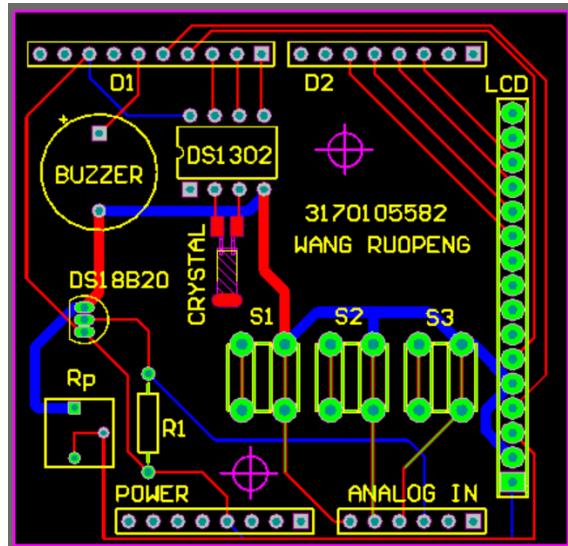


4.4 PCB 板图绘制

根据原理图绘制 PCB 板图，由于部分器件没有提供封装，需要自己设计，比如选择开关的封装按要求设计如下：

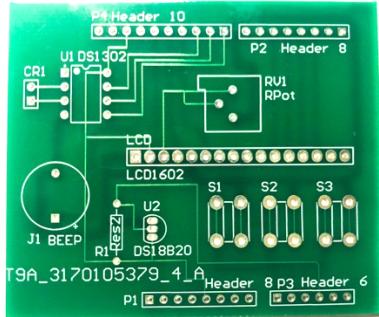


其他部件调用已有的封装，将 PCB 初步设计如下：

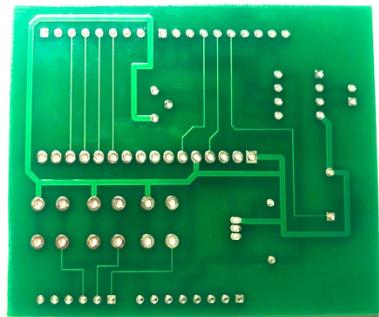


4.5 PCB 制作与元器件焊接

得到的 PCB 实物如下所示，按照原理图焊接元器件，随后进行功能调试。



(PCB 正面)



(PCB 背面)

五、实验结果与功能实现

5.1 显示时间、日期和星期

首先，我们为开机界面进行了个性化设置，即打开后会在屏幕上显示”Digital Clock”，持续 2 秒，如左图所示。正常工作的显示状态如右图所示。



(开机界面)



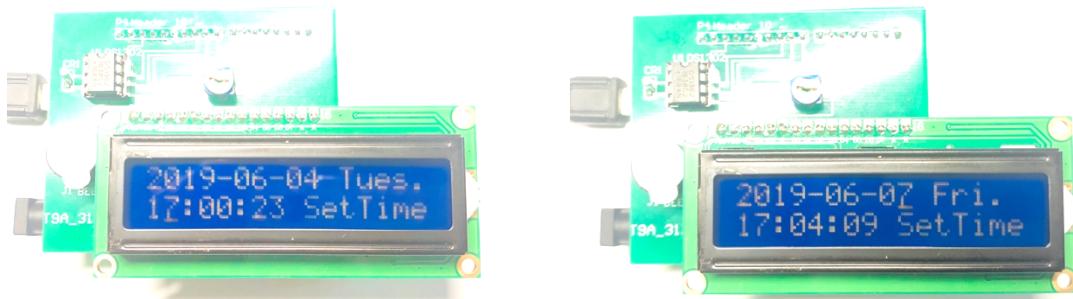
(正常工作状态)

5.2 断电保存时间

经过不断调试，我们发现很难在断电之前将时间数据写入 DS1302 芯片，于是只好在 IDE 里通过修改代码初始化开机时间。

5.3 通过按钮设置时间、日期

按最左边的按钮 S1 进入设置时间模式 SetTime，再次按 S1 会实现各个数字之间的切换，再按 S2/S3 实现 +/- 功能。由此，通过按钮来设置时间、日期的功能得以实现。下图分别是对时间和日期的设置。

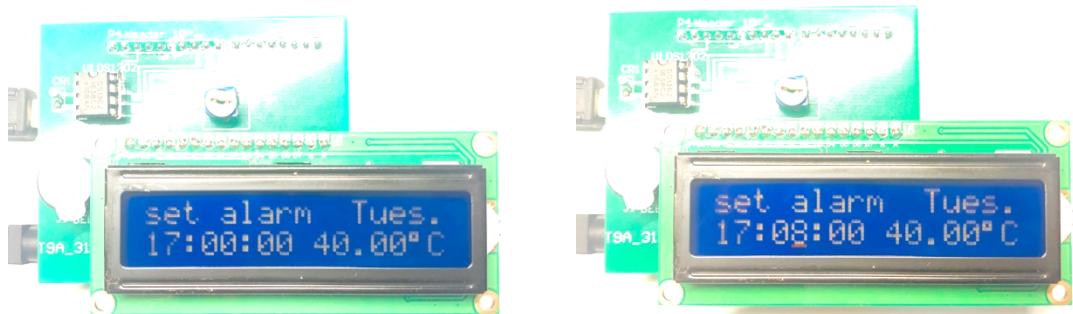


5.4 整点响铃

基本的整点响铃的功能成功实现，同时，为了增加用户体验，我们对代码进行了个性化修改，实现了“几点响几下”的功能，如 7 点响 7 下，16 点响 4 下。

5.5 自定义闹钟

同时按+/-键进入闹钟和报警温度设置功能；按 S1 按钮对各个数字进行修改，实现自定义闹钟的功能。如下图，将默认的 17:00 调为 17:08。



5.6 显示温度

如 5.1 图所示，温度数值显示正常，且能够随环境而不断变化。

5.7 自定义报警温度

同时按+/-键进入闹钟和报警温度设置功能；按 S1 按钮对温度值进行修改，实现自定义报警温度的功能。如下图，将默认的 40°C 调为 42°C。



5.8 按键功能:按选择键进入设置时间功能；同时按+/-键进入闹钟和报警温度设置功能。该功能已实现。

5.9 再按选择键光标跳动，光标跳到哪，当前的参数即可通过加减键修改。

该功能已实现

5.10 结果：所有功能均已通过老师验收。

六、总结与体会

经过上学期的模拟电路和本学期的数字电路相关课程的学习，我对电路设计有了初步的认识。但很遗憾一直没有机会亲手设计一个电子系统，而本学期的电子设计实验 II 恰好给了我这个机会。常言道要“学以致用”，有了理论的知识，就需要把它们应用于工程实践中去，这样才能体现出来它们的价值。

首先在选题的时候，由于上学期恰好上过电子工程训练这门工程实践性极强的课，对 Arduino 的应用还记忆犹新，于是主要偏向于和 Arduino 相关的课题。经过和组内另一位同学的讨论，我们决定选择多功能数字时钟。一是功能多样却不复杂，二是可以通过代码进行测试。总之，这道题虽具有一定的挑战性，但也在我能力范围之内。

确定选题之后，我们根据给的参考原理图，根据老师提供的器件设计了多功能数字时钟的原理图。老师提供了 Arduino 扩展板、LCD 显示屏、蜂鸣器、DS1302 等元器件。自己在网上下载了相关器件的 datasheet，这个过程让我有了电子工程师的感觉。因为在未来的工作中，没有人会告诉你这个器件的特性是什么，需要自己去搜寻，网络就是一个很好的平台。在查询资料的过程中，我发现了许多专业好网站，如电子发烧友网、Alldatasheet 等，上面有非常全面的信息，各个公司的电子元件应有尽有。找到需要的资料后，我再根据各个元件的电气特性，设计出了具体电路和功能模块。同时在 IDE 上改进给的参考测试代码。

原理图和代码设计完毕，接下来要做的就是在面包板上搭建电路并进行测试。由于上学期上过信息与电子工程导论这门课，当时的大作业就是在面包板上搭建加法器，所以我对面包板的特性比较熟悉，插线插器件也相对熟练。用了不到一节课的时间，我就完成了搭建工作，还检查了一遍。然而，由于部分杜邦线针头不结实，有时会出现插线脱落的现象，就需要再对着原理图插线，相对费时。在测试过程中，刚开始出现了 bug，最后发现是由于把选择开关接错造成的，启示我以后搭建电路时要仔细认真，确定好元件内部的电连接方式后再插板，以免后期 debug 过于繁琐，这样还可以节约很多时间出来对系统进行优化。总体上测试算是比较成功，但有一个问题就是显示屏亮度较暗。我们用万用表进行了测量，发现是 Arduino 板上的 5V 输出口只输出了 3V 左右的电压。有时还会出现屏幕上数字闪烁的情况，经过排查确定了是因为面包板接触不良造成的。

最后该画 PCB 板图了，这是我第一次画 PCB，有点生疏。在老师讲解过后，没有立刻掌握，还要不断地回顾上课时录的视频才慢慢学会。画出来时还是非常有成就感的。

夏学期一开始，我们提交了 PCB 板图，并在一周后拿到了成品。看到 PCB 的那一刻，心情还是十分激动的，毕竟是自己设计的第一块电路板。我们将所有元器件焊接上去后，接通了电源，发现数字时钟能够正常显示，没有出现春学期的数字闪烁情况，由此可见当时确实是由于面包板接触不良造成的。由于原理图绘制正确，少走了很多弯路，因此在接下来的几周时间里，我们主要是在调试代码，找出了一些 bug，不断优化设计，最终实现了多功能数字时钟的各种功能。

这是我利用电子电路知识完成的第一个项目，收获颇丰，非常感谢老师们能给予我这个机会。在这个过程中，不仅巩固了我的电路知识，也让我学习了新的技能，如绘制 PCB、调试 Arduino 等等。

附录：测试代码

```
/*
 * LCD RS pin to digital pin 12
 * LCD Enable pin to digital pin 11
 * LCD D4 pin to digital pin 5
 * LCD D5 pin to digital pin 4
 * LCD D6 pin to digital pin 3
 * LCD D7 pin to digital pin 2
 * LCD R/W pin to ground
 * LCD VSS pin to ground
 * LCD VCC pin to 5V
 */

#include <DS1302.h>
#include <LiquidCrystal.h> //LCD1602 显示头文件
#include <OneWire.h> //温度传感器 DS18B20 头文件
#include <DallasTemperature.h> //温度传感器 DS18B20 头文件
#define ONE_WIRE_BUS A3 //DS18B20 信号端口
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
#define choose A0 //选择端口
#define add A1 //加
#define minus A2 //减
#define Tone 13 //蜂鸣器端口
uint8_t CE_PIN = 8; //DS1302 RST 端口
uint8_t IO_PIN = 9; //DS1302 DAT 端口
uint8_t SCLK_PIN = 10; //DS1302 CLK 端口
DS1302 rtc(CE_PIN, IO_PIN, SCLK_PIN); //创建 DS1302 对象
unsigned long seconds;
int s = 0, m = 0, h = 0, d = 0, mon = 0, y = 0; //时间进位
int second = 0, minute = 0, hour = 0, day = 0, month = 0, year = 0; //当前时间
int SECOND = 0, MINUTE = 0, HOUR = 0, DAY = 0, MONTH = 0, YEAR = 0; //初始时间
int chose = 0, alarm_choose = 0 ,ButtonDelay = 10, frequence = 2093;
int alarm_hour = 16, alarm_minute = 0, alarm_second = 0; //闹钟时间
double Temperatures, Temp_Alarm = 40 ;
void setup(){
    for(int i = 2;i <= 13;i++){
        pinMode(i,OUTPUT);
    }
    digitalWrite(add, HIGH);
    digitalWrite(minus, HIGH);
    digitalWrite(choose, HIGH);
    lcd.begin(16, 2); //初始化 LCD1602
```

```

sensors.begin(); //初始化温度传感器 DS18B20
//Time t(2019, 5, 28, 17, 0, 0, 1); rtc.time(t); //设置 DS1302 芯片初始时间
set(rtc.year(), rtc.month(), rtc.date(), rtc.hour(), rtc.minutes(), rtc.seconds()); //通电读取 DS1302
芯片的时间
rtc.write_protect(false); // 关闭 DS1302 芯片写保护
rtc.halt(false); //为 true 时 DS1302 暂停
lcd.clear();
lcd.print(" Digital Clock");
delay(2000);
lcd.clear();
}
/** 格式化输出 */
void FormatDisplay(int col, int row,int num){
lcd.setCursor(col, row);
if(num < 10) lcd.print("0");
lcd.print(num);
}
/** 计算时间 */
void time() {

int SECOND = 45, MINUTE = 59, HOUR = 16, DAY = 4, MONTH = 6, YEAR = 2019;

second = (SECOND + seconds)%60; //计算秒
m = (SECOND + seconds)/60; //分钟进位
FormatDisplay(6,1,second);

minute = (MINUTE + m)%60; //计算分钟
h = (MINUTE + m)/60; //小时进位
FormatDisplay(3,1,minute);

hour = (HOUR + h)%24; //计算小时
d = (HOUR + h)/24; //天数进位
FormatDisplay(0,1,hour);

lcd.setCursor(2, 1);
lcd.print(":");
lcd.setCursor(5, 1);
lcd.print(":");
}
/** 根据年月计算当月天数 */
int Days(int year, int month){
int days = 0;
if (month != 2){
switch(month){


```

```

case 1: case 3: case 5: case 7: case 8: case 10: case 12: days = 31; break;
case 4: case 6: case 9: case 11: days = 30; break;
}
}else{ //闰年
if(year % 4 == 0 && year % 100 != 0 || year % 400 == 0){
days = 29;
}
else{
days = 28;
}
}
return days;
}
/** 计算当月天数 */
void Day(){
int days = Days(year,month);
int days_up;
if(month == 1){
days_up = Days(year - 1, 12);
}
else{
days_up = Days(year, month - 1);
}
day = (DAY + d) % days;
if(day == 0){
day = days;
}
if((DAY + d) == days + 1 ){
DAY -= days;
mon++;
}
if((DAY + d) == 0){
DAY += days_up;
mon--;
}
FormatDisplay(8,0,day);
}
/** 计算月份 */
void Month(){
month = (MONTH + mon) % 12;
if(month == 0){
month = 12;
}
y = (MONTH + mon - 1) / 12;
}

```

```

FormatDisplay(5,0,month);
lcd.setCursor(7, 0);
lcd.print('-');
}
/** 计算年份 */
void Year(){
    year = ( YEAR + y ) % 9999;
    if(year == 0){
        year = 9999;
    }
    lcd.setCursor(0, 0);
    if(year < 1000){
        lcd.print("0");
    }
    if(year < 100){
        lcd.print("0");
    }
    if(year < 10){
        lcd.print("0");
    }
    lcd.print(year);
    lcd.setCursor(4, 0);
    lcd.print('-');
}
/** 根据年月日计算星期几 */
void Week(int y,int m, int d){
    if(m == 1){
        m = 13;
    }
    if(m == 2){
        m = 14;
    }
    int week = (d+2*m+3*(m+1)/5+y+y/4-y/100+y/400)%7+1;
    String weekstr = "";
    switch(week){
        case 1: weekstr = "Mon. "; break;
        case 2: weekstr = "Tues. "; break;
        case 3: weekstr = "Wed. "; break;
        case 4: weekstr = "Thur. "; break;
        case 5: weekstr = "Fri. "; break;
        case 6: weekstr = "Sat. "; break;
        case 7: weekstr = "Sun. "; break;
    }
    lcd.setCursor(11, 0);
}

```

```
lcd.print(weekstr);
}
/** 显示时间、日期、星期 */
void Display() {
    time();
    Day();
    Month();
    Year();
    Week(year,month,day);
}
/** 显示光标 */
void DisplayCursor(int rol, int row) {
    lcd.setCursor(rol, row);
    lcd.cursor();
    delay(100);
    lcd.noCursor();
    delay(100);
}
/** 设置初始时间 */
void set(int y, int mon, int d, int h, int m, int s){
    YEAR = y;
    MONTH = mon;
    DAY = d;
    HOUR = h;
    MINUTE = m;
    SECOND = s;
}
/** 通过按键设置时间 */
void Set_Time(int rol, int row, int &Time){
    DisplayCursor(rol, row);
    if(digitalRead(add) == LOW){
        delay(ButtonDelay);
        if(digitalRead(add) == LOW){
            Time++;
        }
        Display();
    }
    if(digitalRead(minus) == LOW){
        delay(ButtonDelay);
        if(digitalRead(minus) == LOW){
            Time--;
        }
        Display();
    }
}
```

```
}

/** 按键选择 */

void Set_Clock(){
    if(digitalRead(choose)==LOW){
        lcd.setCursor(9, 1);
        lcd.print("SetTime");
        while(1){
            if(digitalRead(choose) == LOW){
                delay(ButtonDelay);
                if(digitalRead(choose) == LOW){
                    chose++;
                }
            }
            seconds = millis()/1000;
            Display();
            if(chose == 1){
                Set_Time(1, 1, HOUR); //SetHour
            }else if(chose == 2){
                Set_Time(4, 1, MINUTE); //SetMinute
            }else if(chose == 3){
                Set_Time(7, 1, SECOND); //SetSecond
            }else if(chose == 4){
                Set_Time(9, 0, DAY); //SetDay
            }else if(chose == 5){
                Set_Time(6, 0, MONTH); // SetMonth
            }else if(chose == 6){
                Set_Time(3, 0, YEAR); //SetYear
            }else if(chose >= 7) {
                chose = 0;
                break;
            }
        }
    }
}

/** 设置闹钟小时 */

void Set_Alarm_Hour(){
    DisplayCursor(1, 1);
    if(digitalRead(add) == LOW){
        delay(ButtonDelay);
        if(digitalRead(add) == LOW){
            alarm_hour++;
            if(alarm_hour == 24){
                alarm_hour = 0;
            }
        }
    }
}
```

```
FormatDisplay(0,1,alarm_hour);
}
}
if(digitalRead(minus) == LOW){
delay(ButtonDelay);
if(digitalRead(minus) == LOW){
alarm_hour--;
if(alarm_hour == -1){
alarm_hour = 23;
}
FormatDisplay(0,1,alarm_hour);
}
}
}
/** 设置闹钟分钟 */
void Set_Alarm_Minute(){
DisplayCursor(4, 1);
if(digitalRead(add) == LOW) {
delay(ButtonDelay);
if(digitalRead(add) == LOW){
alarm_minute++;
if(alarm_minute == 60){
alarm_minute = 0;
}
FormatDisplay(3,1,alarm_minute);
}
}
if(digitalRead(minus) == LOW){
delay(ButtonDelay);
if(digitalRead(minus) == LOW){
alarm_minute--;
if(alarm_minute == -1){
alarm_minute = 59;
}
FormatDisplay(3,1,alarm_minute);
}
}
}
/** 设置报警温度 */
void Set_Alarm_Temp(){
DisplayCursor(10, 1);
if(digitalRead(add) == LOW) {
delay(ButtonDelay);
if(digitalRead(add) == LOW){
```

```

Temp_Alarm++;
}
}
if(digitalRead(minus) == LOW){
delay(ButtonDelay);
if(digitalRead(minus) == LOW){
Temp_Alarm--;
}
}
}
/** 进入报警设置 */
void Set_Alarm(){
if(digitalRead(add) == LOW && digitalRead(minus) == LOW){
alarm_hour = hour;
alarm_minute = minute;
//alarm_choose = 1;
lcd.setCursor(0, 0);
lcd.print("set alarm ");
lcd.setCursor(6, 1);
lcd.print("00"); //闹钟秒数
while(1){
if(digitalRead(choose) == LOW){
delay(ButtonDelay);
if(digitalRead(choose) == LOW){
alarm_choose++;
}
}
lcd.setCursor(9, 1);
lcd.print(Temp_Alarm);
lcd.setCursor(14, 1);
lcd.print((char)223); //显示 o 符号
lcd.setCursor(15, 1);
lcd.print("C"); //显示字母 C
if(alarm_choose == 1){
Set_Alarm_Hour();
} else if(alarm_choose == 2){
Set_Alarm_Minute();
} else if(alarm_choose == 3){
Set_Alarm_Temp();
} else if(alarm_choose >= 4){
alarm_choose = 0;
break;
}
}
}

```

```

        }
    }

/** 正点蜂鸣 */
void Point_Time_Alarm(){
    if(minute == 0 && second == 0){
        for(int i=0; i<hour%12; i++){
            tone(Tone,frequence);
            delay(500);
            noTone(Tone);
            delay(300);
        }
    }
}

/** 闹钟 指定时间蜂鸣 */
void Clock_Alarm(){
    if(hour == alarm_hour && minute == alarm_minute && second == alarm_second){
        for(int i=0; i<5; i++){
            tone(Tone,frequence); delay(500);
            noTone(Tone); delay(500);
        }
    }
}

/** 获取 DS18B20 温度 */
void GetTemperatures(){
    sensors.requestTemperatures(); // Send the command to get temperatures
    Temperatures = sensors.getTempCByIndex(0)-5;
    lcd.setCursor(9, 1);
    lcd.print(Temperatures); //获取温度
    lcd.setCursor(14, 1);
    lcd.print((char)223); //显示 o 符号
    lcd.setCursor(15, 1);
    lcd.print("C"); //显示字母 C
}

/** 超过指定温度报警 */
void Temperatures_Alarm(){
    if(Temperatures >= Temp_Alarm){
        tone(Tone,frequence);
        delay(2000);
        noTone(Tone);
        delay(500);
    }
}

int flag=0;

```

```
void loop() {
    if(flag==0){
        set(2019,6,4,16,0,0);
        //set(rtc.year(), rtc.month(), rtc.date(), rtc.hour(), rtc.minutes(), rtc.seconds());
        flag=1;
    }
    seconds = millis()/1000; //获取单片机当前运行时间
    Display(); //显示时间
    Set_Clock(); //设置时间
    Set_Alarm(); //设置闹钟
    Point_Time_Alarm(); //正点蜂鸣
    Clock_Alarm(); //闹钟时间蜂鸣
    GetTemperatures(); //获取 DS18B20 温度
    Temperatures_Alarm(); //超过指定温度报警
    Time t(year, month, day, hour, minute, second, 1); //断电将单片机的当前时间写到 DS1302 芯片中
    rtc.time(t);
}
```