

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：黄嘉欣

学 院：信息与工程学院

系：信息与工程学系

专 业：信息工程

学 号：3190102060

指导教师：李锡华 叶险峰 施红军

2021 年 5 月 3 日

浙江大学实验报告

专业： 信息工程
姓名： 黄嘉欣
学号： 3190102060
日期： 2021年5月3日
地点： 东四-224

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： PCB 原理图与版图设计 实验类型： 设计性实验 同组学生姓名： 刘懿萱

一、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。

二、实验任务与要求

- ① 用 Arduino UNO 设计多功能数字时钟，要求实现功能：
 - a) 显示时间、日期和星期；
 - b) 断电保存时间；
 - c) 通过按钮设置时间、日期；
 - d) 整点响铃；
 - e) 自定义闹钟；
 - f) 显示温度；
 - g) 自定义报警温度；
 - h) 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；
 - i) 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；
- ② 设计电路，完成相应器件的选择和参数计算，制作 Arduino UNO 扩展板；
- ③ 编制与调试多功能数字时钟程序。

三、硬件模块

本项目使用 Arduino UNO 设计多功能数字时钟。其硬件框图如图 2.1 所示，下面对各部分模块进行说明。

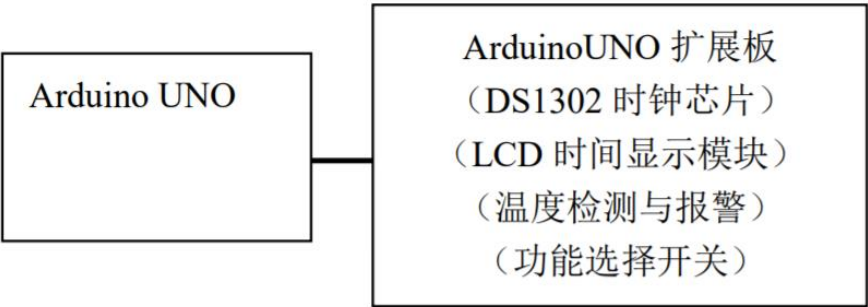


图 2.1 多功能数字时钟硬件框图

① DS1302 时钟芯片

DS1302 是一款高性能、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，一个月小于 31 天时可以自动调整，且具有闰年补偿功能。其工作电压宽达 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，且提供对后背电源进行涓细电流充电的能力。

② LCD 显示模块

本次实验使用 Arduino UNO 直接驱动 1602 液晶显示字母和数字。众所周知，LCD 显示模块的一个显著优势就是将驱动电路集成在模块当中，其具有标准，不同的生产商所生产的大多数模块都可以按照相同的方法来使用。

1602 液晶在实际项目中应用非常广泛。最初的 1602 液晶使用的是 HD44780 控制器，现在各个厂家的 1602 模块基本上都采用了与之兼容的 IC，所以在特性方面基本都是一致的。

③ 温度传感器

温度传感器种类繁多，但在应用与高精度、高可靠性方面，DS18B20 温度传感器性能比较出众。超小的体积、超低的硬件开销、超强的抗干扰能力，超高的精度和附加功能，使得 DS18B20 更受欢迎。其主要特征为：

a) 全数字温度转换及输出；

装
订
线

- b) 先进的单总线数据通信;
- c) 最高 12 位分辨率, 精度可达 ± 0.5 摄氏度;
- d) 12 位分辨率时的最大工作周期为 750 毫秒;
- e) 可选择寄生工作方式;
- f) 检测温度范围为 $-55^{\circ}\text{C} \sim +125^{\circ}\text{C}$ ($-67^{\circ}\text{F} \sim +257^{\circ}\text{F}$);
- g) 内置 EEPROM, 限温报警功能;
- h) 64 位光刻 ROM, 内置产品序列号, 方便多机挂接;
- i) 多样封装形式, 适应不同硬件系统。

四、实验原理与设计

本实验采用 Arduino UNO 设计多功能数字时钟, 主要功能部分由 Arduino 板直接驱动, 其中, LCD 显示模块用于时间、温度显示, DS18B20 温度传感器用于温度检测, DS1302 时钟芯片用于计时及时间数据储存。此项目既涉及到 Arduino UNO 与扩展板硬件设计, 也涉及到软件功能调试, 故分为以下两个部分:

① 硬件设计

实验的设计参考电路图如下 (Proteus ISIS 仿真图):

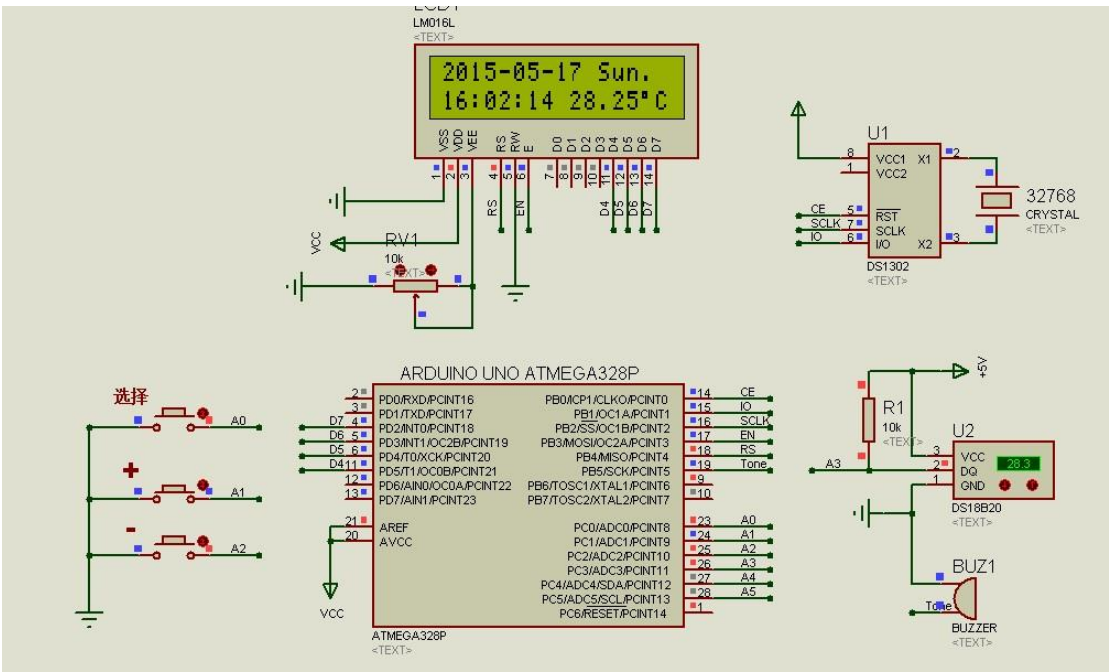


图 4.1 硬件设计参考电路图

② 软件设计 (页码: 5 ~ 15)

设计参考程序如下(有过修改,新增断电保存时间和按键蜂鸣功能):

```
/* *
 * 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>
#include <DallasTemperature.h>  //温度传感器 DS18B20 头文件
#include <EEPROM.h>             //EEPROM 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 = 0;
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 = 7, alarm_minute = 30, alarm_second = 0; //闹钟时间
double Temperatures, Temp_Alarm = 30 ;

union data
{
    int a;
    byte b[4];
};
data temp; //保存年份
```

```
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
    rtc.write_protect(false); // 关闭 DS1302 芯片写保护
    rtc.halt(false); //为 true 时 DS1302 暂停
    //YEAR = 2021; //第一次运行时的时间初始化, 此时需要将下方的 YEAR~SECOND 赋值注释掉
    //MONTH = 5; //使用此段赋值
    //DAY = 1;
    //HOUR = 0;
    //MINUTE = 0;
    //SECOND = 0;
    for (int i=0;i<4;i++){
        temp.b[i]=EEPROM.read(i);
    }
    YEAR = temp.a; //第一次时间存入 EPPROM 后, 将上段内容注释, 使用此段赋值, 重新上传
    MONTH = EEPROM.read(4); //从而具有断电时间保存功能
    DAY = EEPROM.read(5);
    HOUR = EEPROM.read(6);
    MINUTE = EEPROM.read(7);
    SECOND = EEPROM.read(8);
    if (DAY == 1){
        MONTH--;
    }
}

/** 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() {
    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);
    }

```

```
/** 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 ++;
            tone(Tone,frequency); /** 按下蜂鸣 */
            delay(100);
            noTone(Tone);
        }
        Display();
    }
    if(digitalRead(minus) == LOW){
        delay(ButtonDelay);
        if(digitalRead(minus) == LOW){
            Time --;
            tone(Tone,frequency); /** 按下蜂鸣 */
            delay(100);
            noTone(Tone);
        }
        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){
                    tone(Tone,frequency); /** 按下蜂鸣 */
                    delay(100);
                    noTone(Tone);
                    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 ++;
            tone(Tone,frequency); /** 按下蜂鸣 */
            delay(100);
            noTone(Tone);
            if(alarm_hour == 24){
                alarm_hour = 0;
            }
            FormatDisplay(0,1,alarm_hour);
        }
    }
}

if(digitalRead(minus) == LOW){
    delay(ButtonDelay);
    if(digitalRead(minus) == LOW){
        alarm_hour --;
        tone(Tone,frequency); /** 按下蜂鸣 */
        delay(100);
        noTone(Tone);
        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 ++;
            tone(Tone,frequency); /** 按下蜂鸣 */
            delay(100);
            noTone(Tone);
            if(alarm_minute == 60){
                alarm_minute = 0;
            }
            FormatDisplay(3,1,alarm_minute);
        }
    }
    if(digitalRead(minus) == LOW){
        delay(ButtonDelay);
        if(digitalRead(minus) == LOW){
            alarm_minute --;
            tone(Tone,frequency); /** 按下蜂鸣 */
            delay(100);
            noTone(Tone);
            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 ++;

```

```

        tone(Tone,frequency); /** 按下蜂鸣 */
        delay(100);
        noTone(Tone);

    }
}
if(digitalRead(minus) == LOW){
    delay(ButtonDelay);
    if(digitalRead(minus) == LOW){
        Temp_Alarm --;
        tone(Tone,frequency); /** 按下蜂鸣 */
        delay(100);
        noTone(Tone);

    }
}
}

/** 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++;
                    tone(Tone,frequency); /** 按下蜂鸣 */
                    delay(100);
                    noTone(Tone);

                }
            }
            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){
        tone(Tone,frequency);
        delay(500);
        noTone(Tone);
    }
}

/** 闹钟 指定时间蜂鸣 */
void Clock_Alarm(){
    if(hour == alarm_hour && minute == alarm_minute && second == alarm_second){
        tone(Tone,frequency);
        delay(5000);
        noTone(Tone);
    }
}

/** 获取 DS18B20 温度 */
void GetTemperatures(){
    sensors.requestTemperatures(); // Send the command to get temperatures
    Temperatures = sensors.getTempCByIndex(0);
    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(500);
        noTone(Tone);
    }
}

void loop() {
    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);
    rtc.time(t);
    temp.a = year;
    for (int i=0;i<4;i++){
        EEPROM.write(i,temp.b[i]);
    }
    EEPROM.write(4, month);
    EEPROM.write(5, day);
    EEPROM.write(6, hour);
    EEPROM.write(7, minute);
    EEPROM.write(8, second);
}
```

五、实验设备

- ① Arduino UNO R3
- ② 面包板
- ③ DS1302 时钟芯片、LCD1602 液晶显示屏、DS18B20 温度传感器
- ④ 开关、导线、电阻、蜂鸣器等

六、实验步骤与设计结果

- ① 原理图设计

根据参考电路图和代码, 结合 Arduino UNO R3 实物, 我们可以确定各个功能模块

与 Arduino UNO R3 排针接口的对应连线关系为：

表 6.1 功能模块与 Arduino 接口

功能模块 Pin	Arduino Pin	功能模块 Pin	Arduino Pin	功能模块 Pin	Arduino Pin
LCD RS	12	LCD R/W	GND	DS1302 SCLK	10
LCD E	11	LCD VSS	GND	SW-PB4 A0	A0
LCD DB4	5	LCD VCC	5V	SW-PB4 A1	A1
LCD DB5	4	BUZ Tone	13	SW-PB4 A2	A2
LCD DB6	3	DS1302 RST	8	DS18B20 DQ	A3
LCD DB7	2	DS1302 I/O	9		

其余引脚由电气标识对应相连。

设计步骤：

- a) 放置元件，合理布局：如图，点击在 Altium Designer 菜单 “放置(P)->器件(P)”，在弹框中选择 “...”，从安装的元件库中选择所需要的元件（注意封装的正确），命名并放置到画布的合适位置：

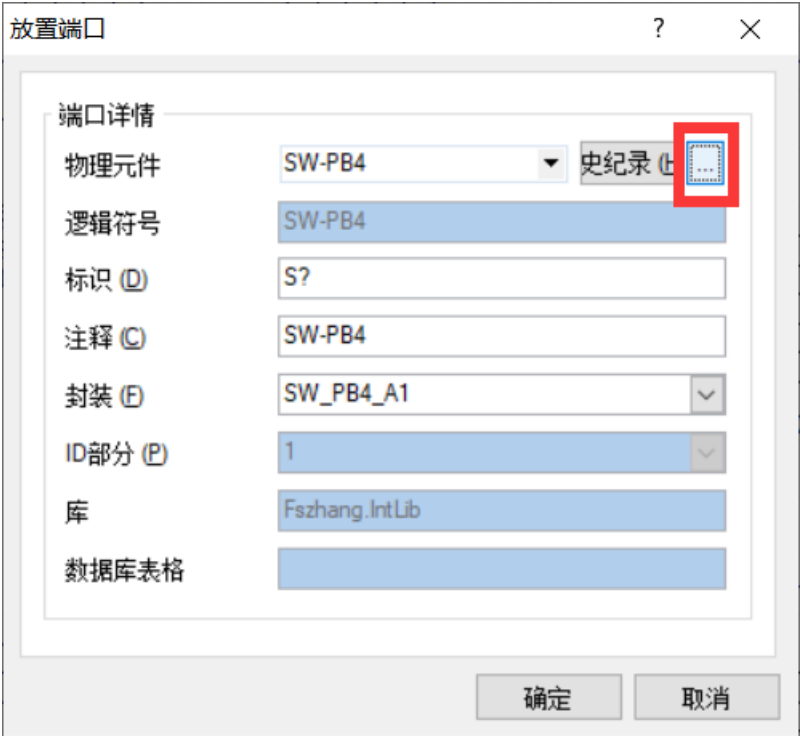


图 6.1.1 放置端口

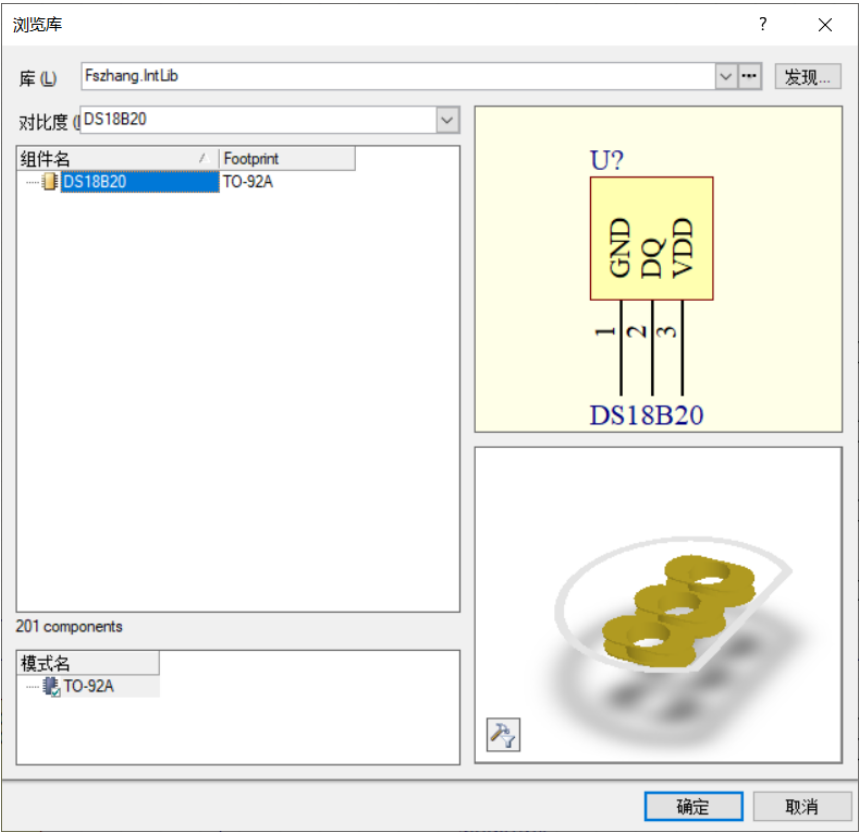


图 6.1.2 选择元件

b) 连线：所有元件放置完成后，根据管脚之间的对应连接关系进行连线。其中，元件管脚与 Arduino 排针接口的对应关系如表 6.1 所示。在菜单中选择“放置线”，如下，光标将变为十字形，此时可以选择连线的起点、转折点和终点，从而将具有连接关系的管脚连接起来；



图 6.1.3 放置线

c) 编译与修改：在菜单中选择“工程(C)->Compile Document *.SchDoc”，编译文件，并在下方“System->Messages”中查看编译信息。若显示有 Error，则需要对对应部分进行修正，直到没有报错为止，如图所示：

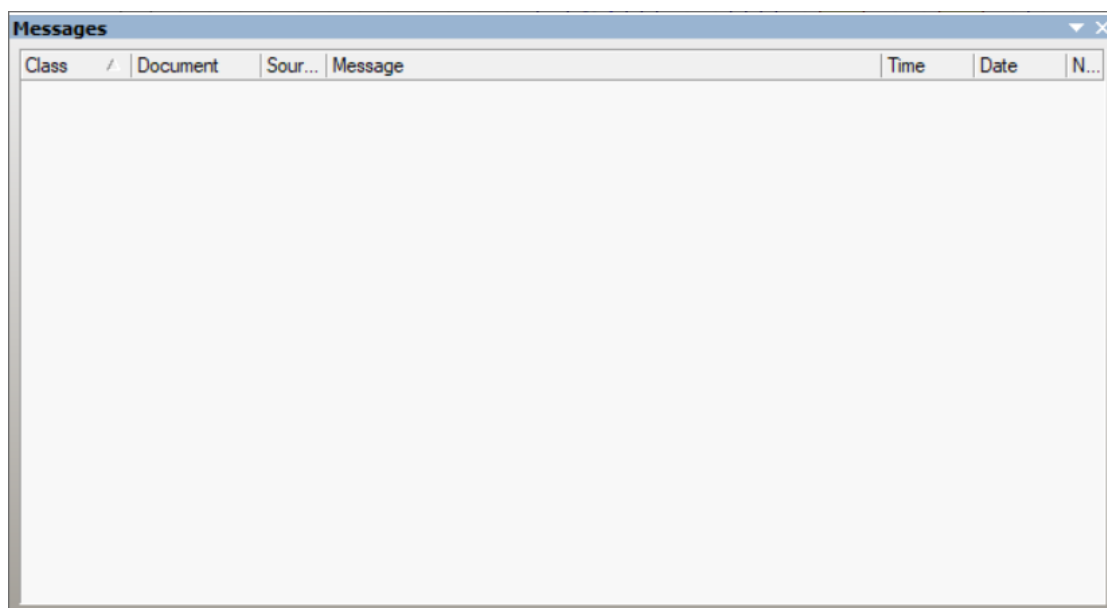


图 6.1.4 编译信息

最终得到的原理图如下图所示：

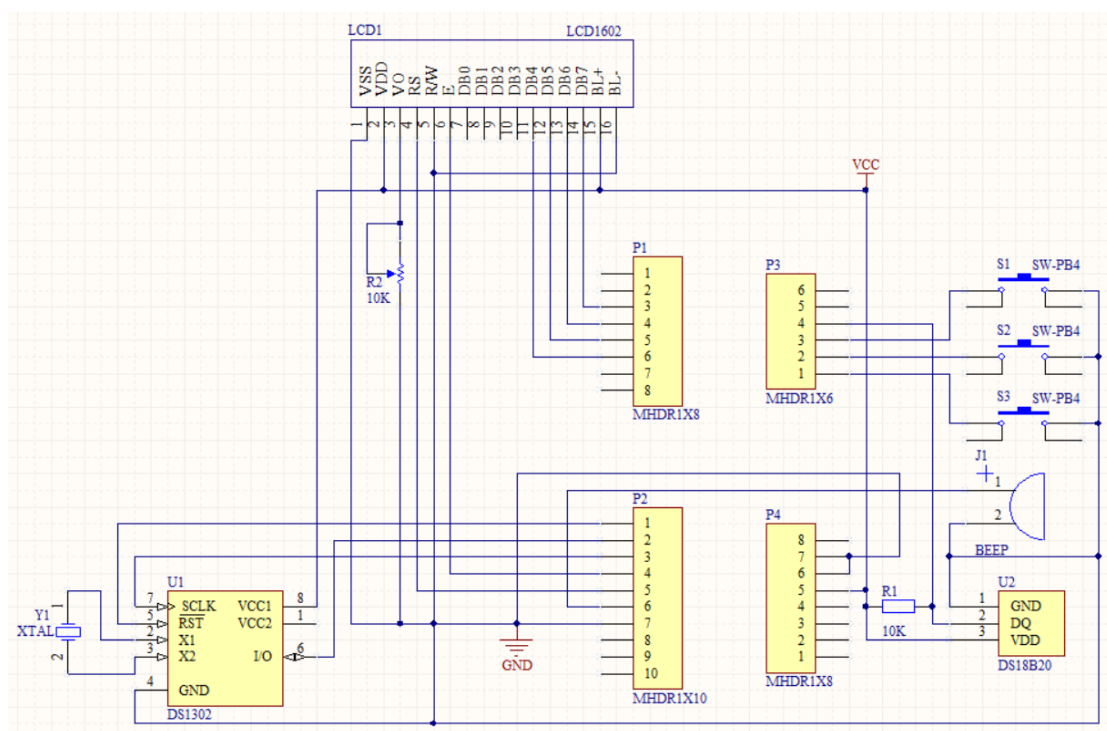


图 6.1.5 电路原理图

所用核心器件为：Arduino UNO R3、DS1302 时钟芯片、LCD1602 液晶显示屏、DS18B20 温度传感器、蜂鸣器、晶振、电阻、开关。

② 面包板调试

根据①中设计出的电路原理图,我们可以在面包板上进行连线调试。在合理布局、正确接线的基础上,我们对给出的参考代码进行了学习,并做出了一些功能上的修改和添加(见 4.2)。通过 Arduino IDE 上传程序,我们在面包板上实现了多功能数字时钟的功能调试,满足预期要求。下图为运行中的电路实物图:

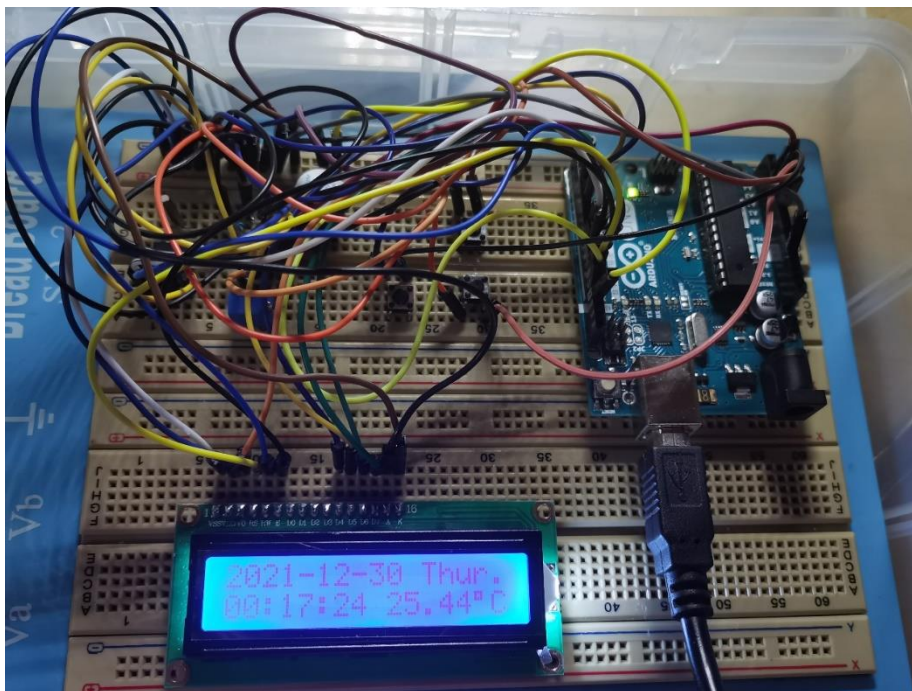


图 6.2.1 面包板实物图



图 6.2.2 设置时间



图 6.2.3 设置闹钟和报警温度

③ PCB 版图设计

由②可知，①中电路设计正确，故可以开始 PCB 版图绘制。其步骤如下：

- a) 导入原理图数据：在工程下新建 PCB 文件并保存，点击菜单“设计(D)->Update PCB document”，在弹出窗口中选择“生效更改”、“执行更改”，如图所示：

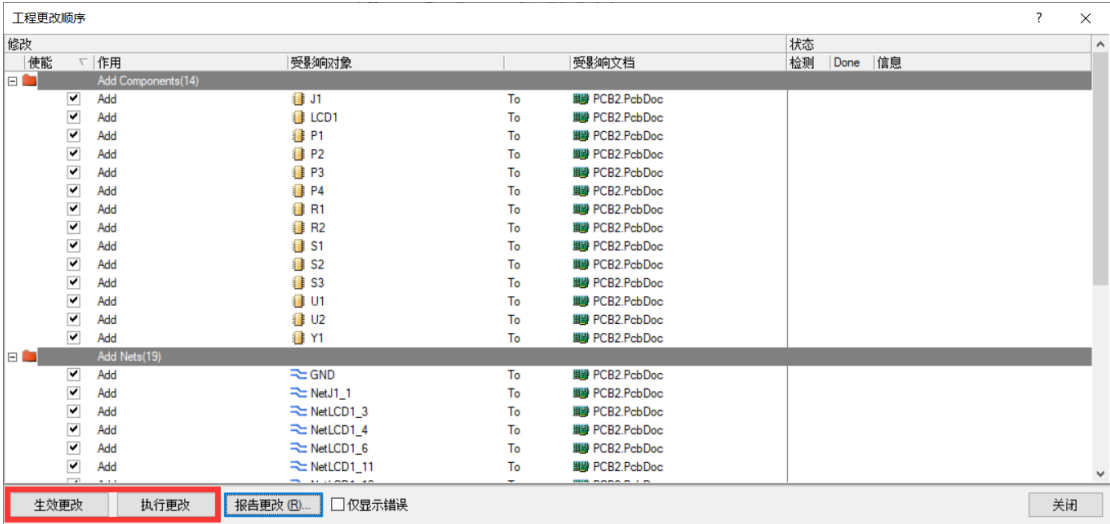


图 6.3.1 导入原理图数据

此时 PCB 文件中会出现所用元件的封装，如图(注意：导入原理图数据前需要选择正确的元件封装)：

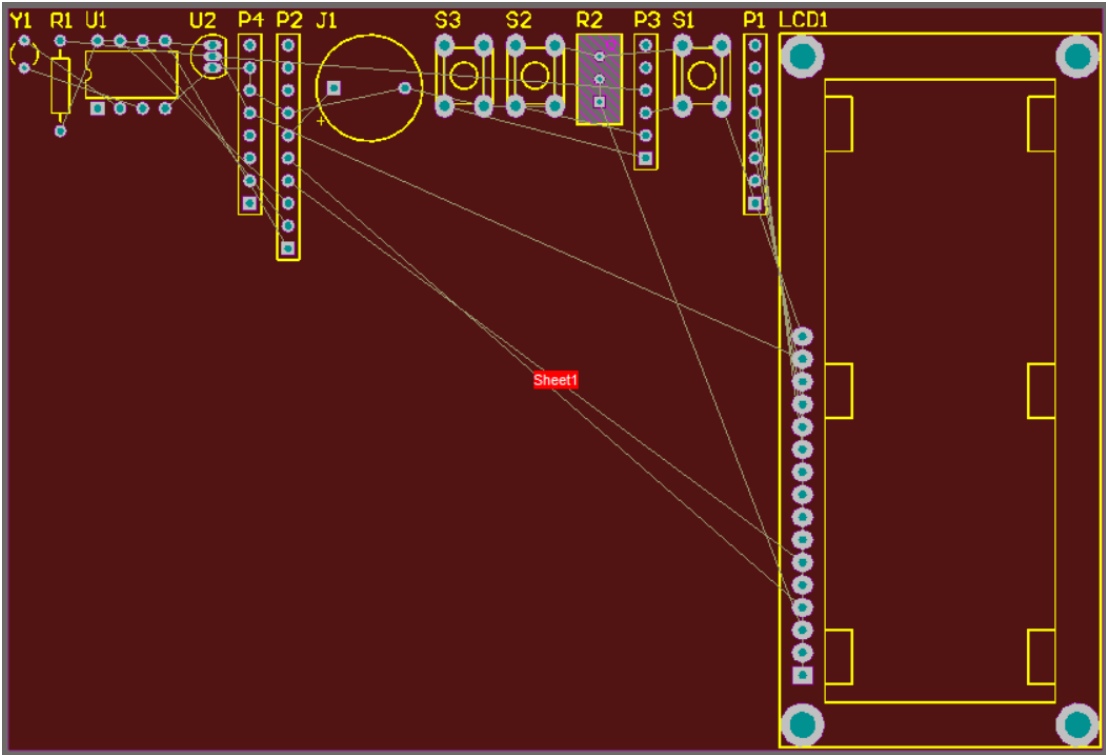


图 6.3.2 PCB 元件封装

b) 设置 PCB 板大小、形状等：将板参数选项设置完成后，根据 Arduino UNO R3 板的排针布局，确定出扩展板排针的相对位置，在菜单中选择“放置(P)->禁止布线->线径”，画出扩展板的外形，如图：

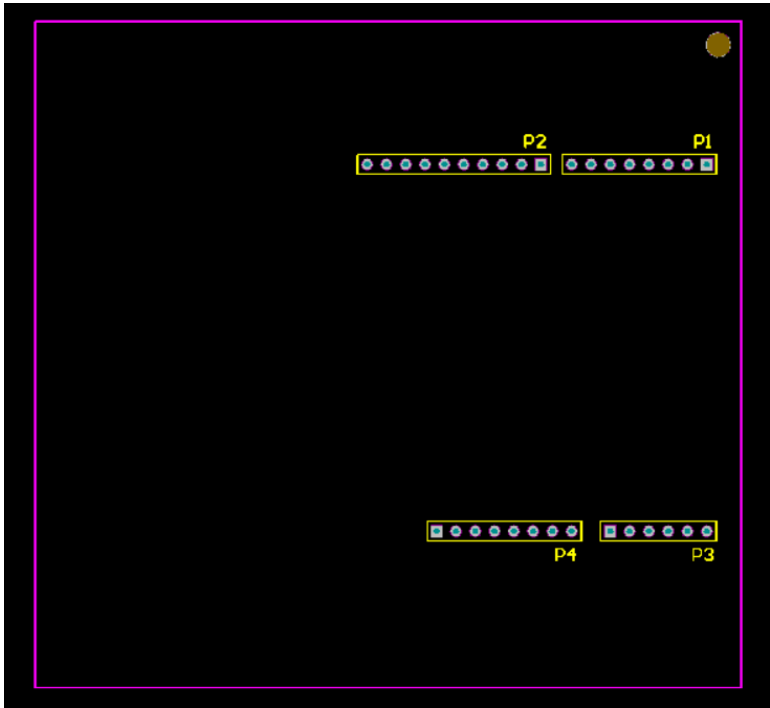


图 6.3.3 PCB 板形状

据此可以留出部分边距，选择“设计(D)->板子形状->按照选择对象定义”，裁剪出相应大小的 PCB 板；

- c) 合理放置元件(布局)、布线：将各元件放置在 PCB 板上合适的位置，尽可能使飞线短且交叉少，同时元件布局紧凑整洁；完成后，点击“放置(P)->Interactive Routing”，开始布线。在布线过程中，为了提高布通率和抗干扰能力，使电路联通更加整洁有序，我们选择了双面布线。最终得到的 PCB 版图如下：

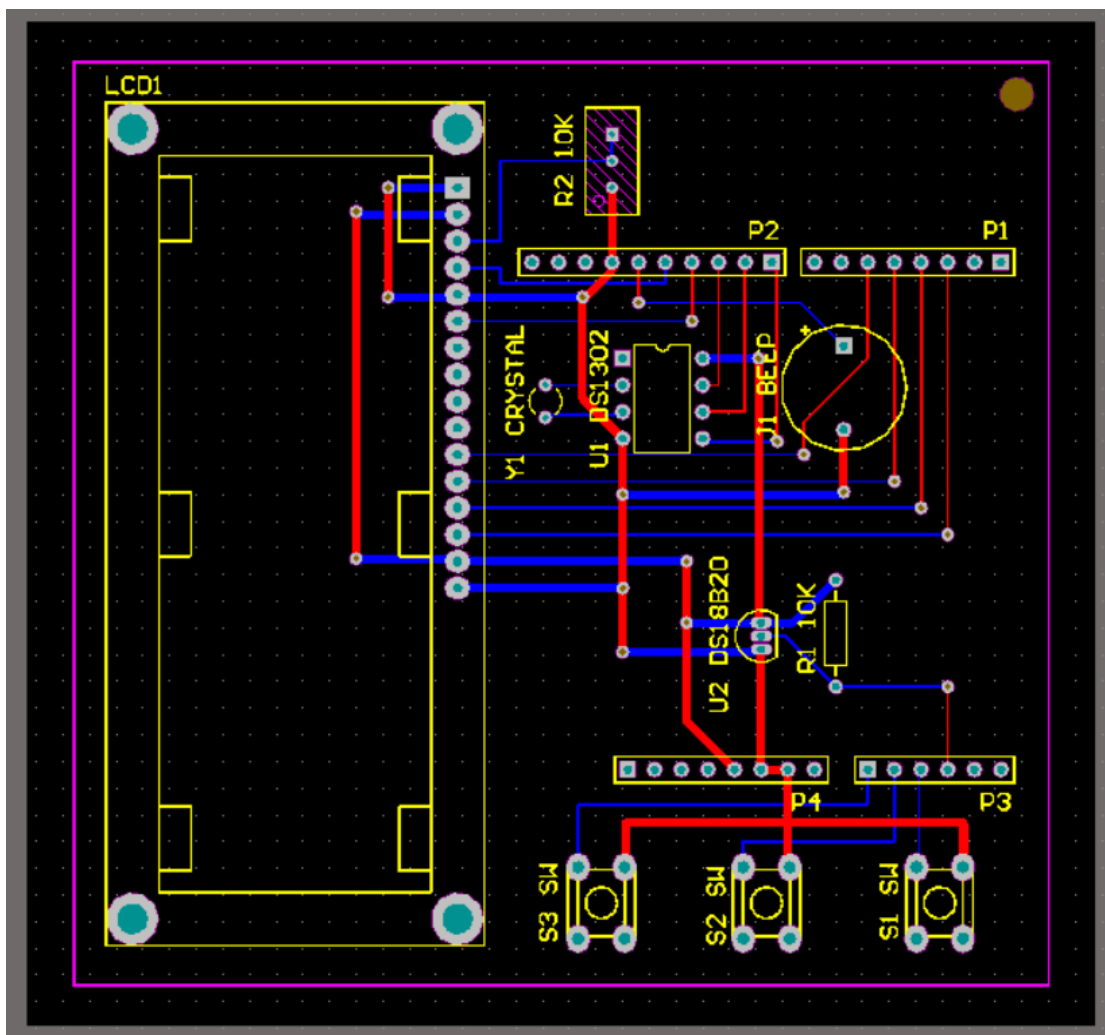


图 6.3 PCB 版图

七、总结与体会

此次实验，我们利用 Altium Designer 对 PCB 原理图进行了设计，并在此基础上完成了面包板的调试实验与 PCB 版图的绘制，既锻炼了我们的设计技能，也让我们对 AD 软件有了一个很好的学习与掌握。除此之外，在对参考代码的学习过程当中，我们也有了

解了 Arduino 编程的基本语法与事项,能够自主对代码进行修改和提高,受益匪浅。

在实验过程当中,我们也遇到过一些困难。在最初原理图设计之时,由于并未去真正了解 Arduino UNO R3 板的结构,我们的原理图连接主要以参考图的连接方法为主。直到开始面包板调试,拿到实物 Arduino 开发板,我们才意识到了此前原理图设计中的错误。所幸,在前面两节课中,我们对各个功能模块的连接方法以及 AD 9 软件的使用有了一个较好的了解,因而比较快速地完成了原理图的修改工作。除此之外,在面包板的调试过程当中,我们也进一步地对元件的管脚和连接进行了熟悉。因为是第一次使用面包板进行调试,我们对其内部结构不甚了解,且各个实物元件的管脚并未标出,需要我们自行上网搜索、查找。但正是在这个过程当中,我们回顾了电子工程训练与电子电路设计实验 I 的相关内容,回顾了如何区分元件的正负方向等,也学会了正确连接面包板进行调试。通过资料查阅,我们发现 DS1302 只有在双电源供电的情况下才能实现断电时间保存,而 Arduino 开发板内置的 EEPROM 可作为永久性储存器件,保存时间信息。因此,为了实现设计要求,我们对 Arduino 代码进行了修改,在每次循环时将当前时间存入 EEPROM,而在每次电路上电时从 EEPROM 读取时间。由于 EEPROM 以字节为单位存储数据,为了正确保存年份,我们采用了共用体结构,以四个字节保存数据,避免了处理年份信息时可能引起的错误。当然,我们也尝试过在硬件电路中添加电容(时钟芯片供电端口并联电容),在断电后为 DS1302 供电。此方法可以使断电后时间继续变化,但坚持时间不长(约几分钟),因此未使用该方案。经过三节课的尝试和修改,最终得到的电路能够正常实现所有功能,大大加强了我们的充实感与获得感。

在绘制 PCB 版图的过程中,虽然我们最初感觉不知道该如何下手,但经过老师的讲解,渐渐有了一些门路。一方面,如何将扩展板的排针接口与 Arduino 板对齐,是我们首先需要考虑的问题。通过将 Arduino UNO R3 的 PCB 版图复制到扩展板 PCB 版图中,对应固定好四组排针的相对位置,我们比较巧妙、快速地解决了这一困扰。当然,由于最初封装选择错误,我们第一次得到的排针封装远小于 Arduino 板,但这也恰恰提醒了我们,设计过程中,每一步都要做到严谨、细心。在布局的过程中,为了使飞线尽可能短且交叉少,我们花了较长时间对元件位置进行调整,既做到了整体的美观简洁、也确保了功能的高效实现。这个过程虽然略有繁琐,但也向我们完整地展示了 PCB 设计工作的全部流程,对我们以后的专业生活和工作都有着很大的借鉴和启发意义。

总的来说,正是在一次又一次的纠错、一次又一次的实践过程当中,我们的专业素养才会不断地培养和提高。通过这次设计实验,我们很好地提高了自己的自学能力和规

划能力,既学会了 Altium Designer 软件的使用,也掌握了 PCB 和 Arduino 扩展板的设计方法,更了解了如何利用 Arduino UNO 设计数字时钟,以及各种功能模块的使用方法。“万事开头难”,但永远不要被开头给难住。俗话说,“师傅领进门,修行靠个人”,在实践的过程中去探索、理解,边做边学、边学边做,渐渐就能够领悟要领,而这也是对我们学习能力的一种锻炼和考验。面对夏学期的电路装配、调试与验收实验,我现在充满着期待。相信,这一系列的设计与装配实验,能够让我们在专业核心素养的道路上,不断提高。