

**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： | 信息与电子工程学院 |
| 系： | 信息与电子工程学系 |
| 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

**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；

装 订 线

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。

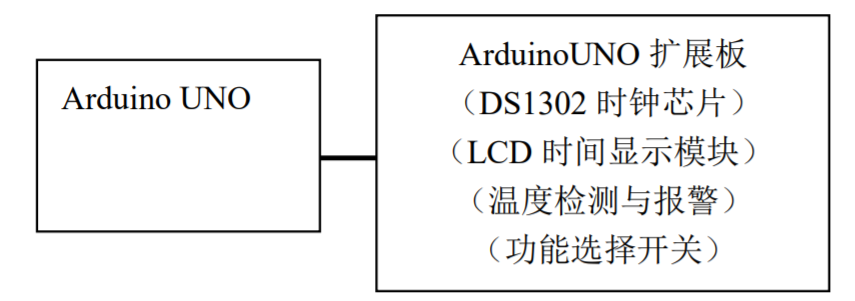


图2.1 多功能数字时钟硬件框图

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

温度传感器种类众多，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、超低的硬件开消、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；

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°C ~ +125°C (–67°F ~ +257°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)：

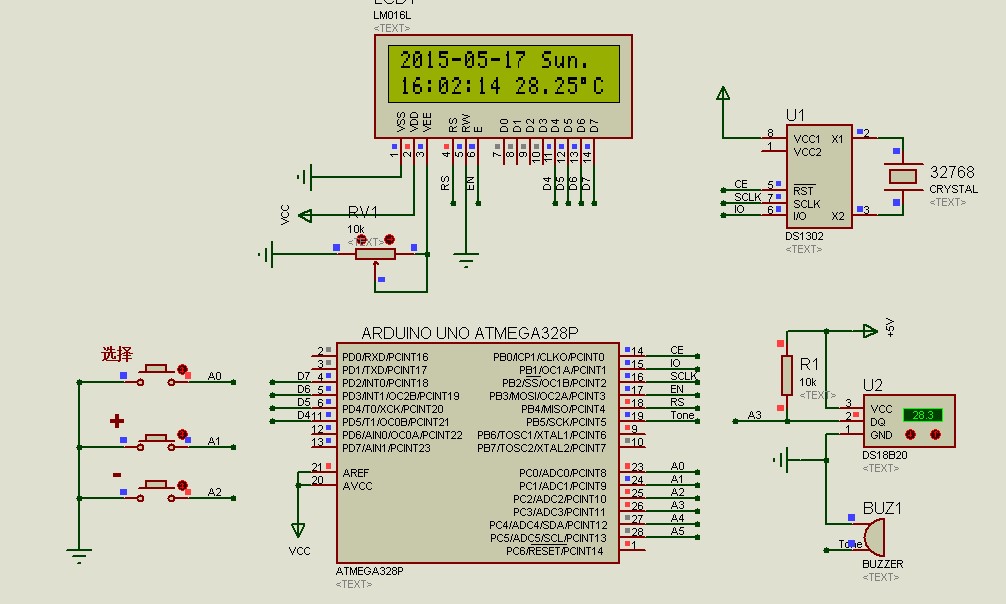


图4.1 硬件设计参考电路图

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,frequence); /\*\* 按下蜂鸣 \*/

delay(100);

noTone(Tone);

}

Display();

}

if(digitalRead(minus) == LOW){

delay(ButtonDelay);

if(digitalRead(minus) == LOW){

Time --;

tone(Tone,frequence); /\*\* 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,frequence); /\*\* 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,frequence); /\*\* 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,frequence); /\*\* 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,frequence); /\*\* 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,frequence); /\*\* 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,frequence); /\*\* 按下蜂鸣 \*/

delay(100);

noTone(Tone);

}

}

if(digitalRead(minus) == LOW){

delay(ButtonDelay);

if(digitalRead(minus) == LOW){

Temp\_Alarm --;

tone(Tone,frequence); /\*\* 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,frequence); /\*\* 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,frequence);

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,frequence);

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：

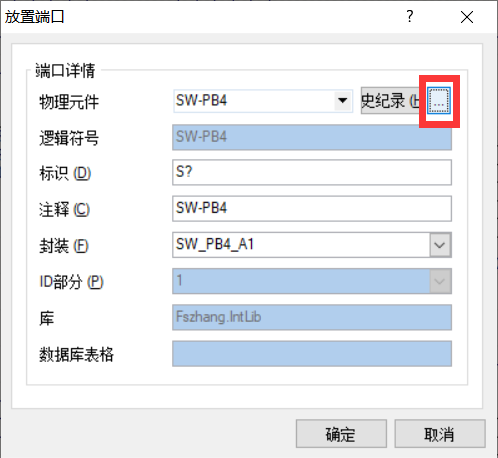


图6.1.1 放置端口

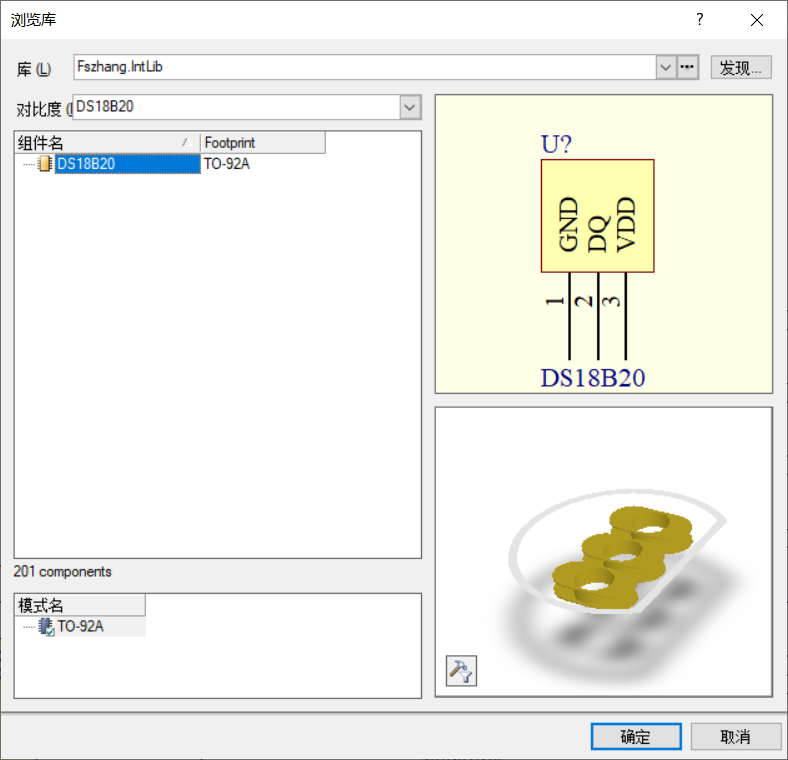


图6.1.2 选择元件

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：

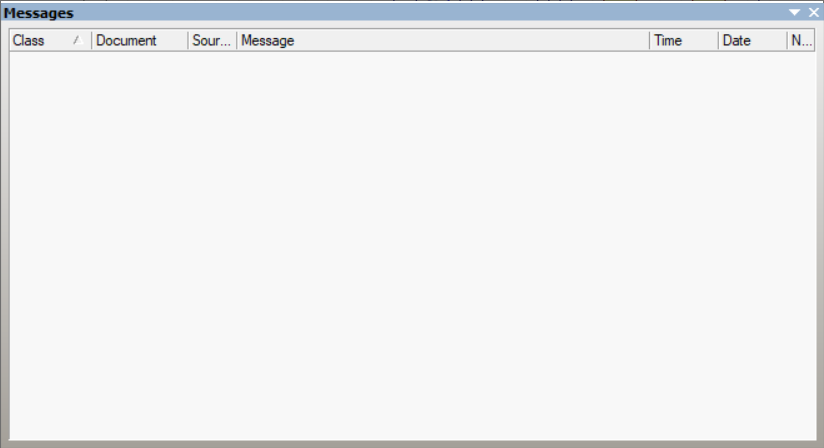


图6.1.4 编译信息

最终得到的原理图如下图所示：

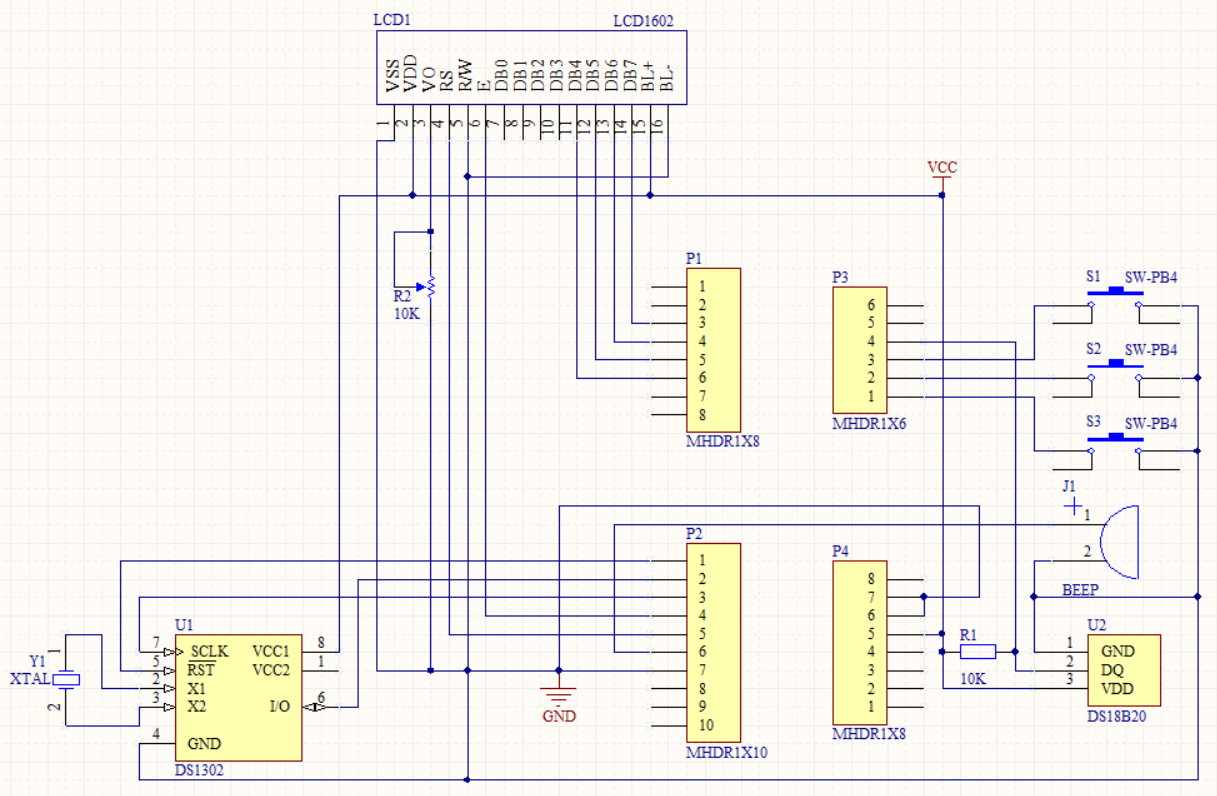


图6.1.5 电路原理图

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：

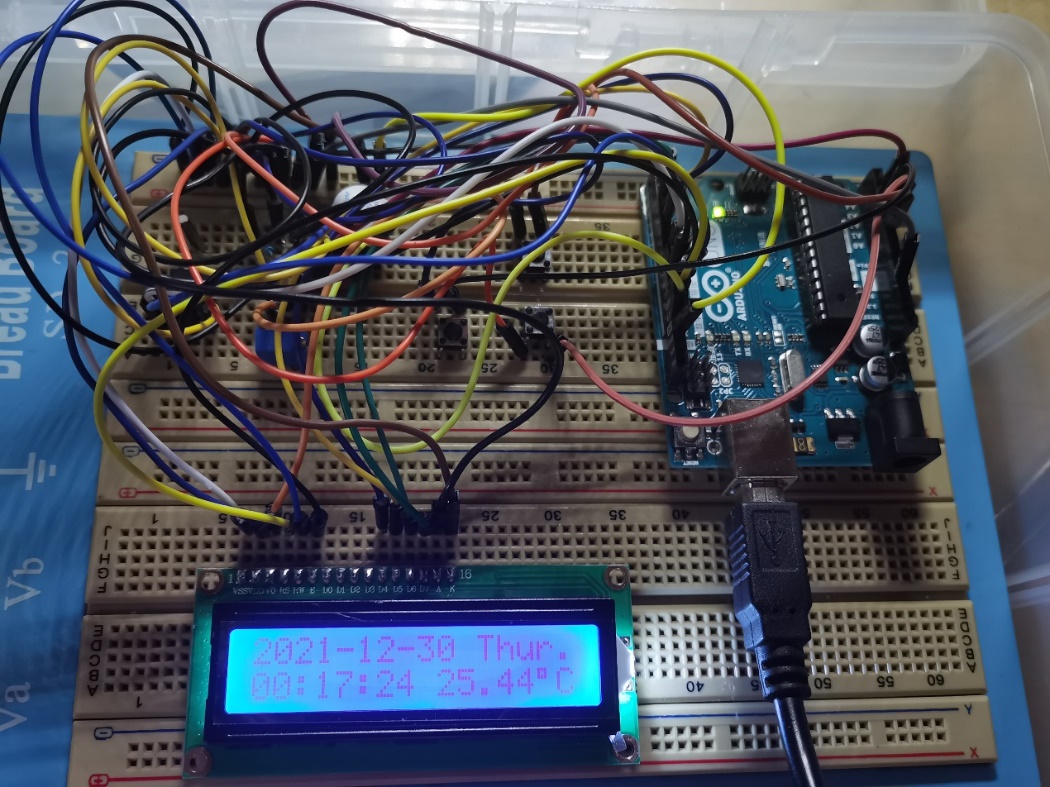


图6.2.1 面包板实物图



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：

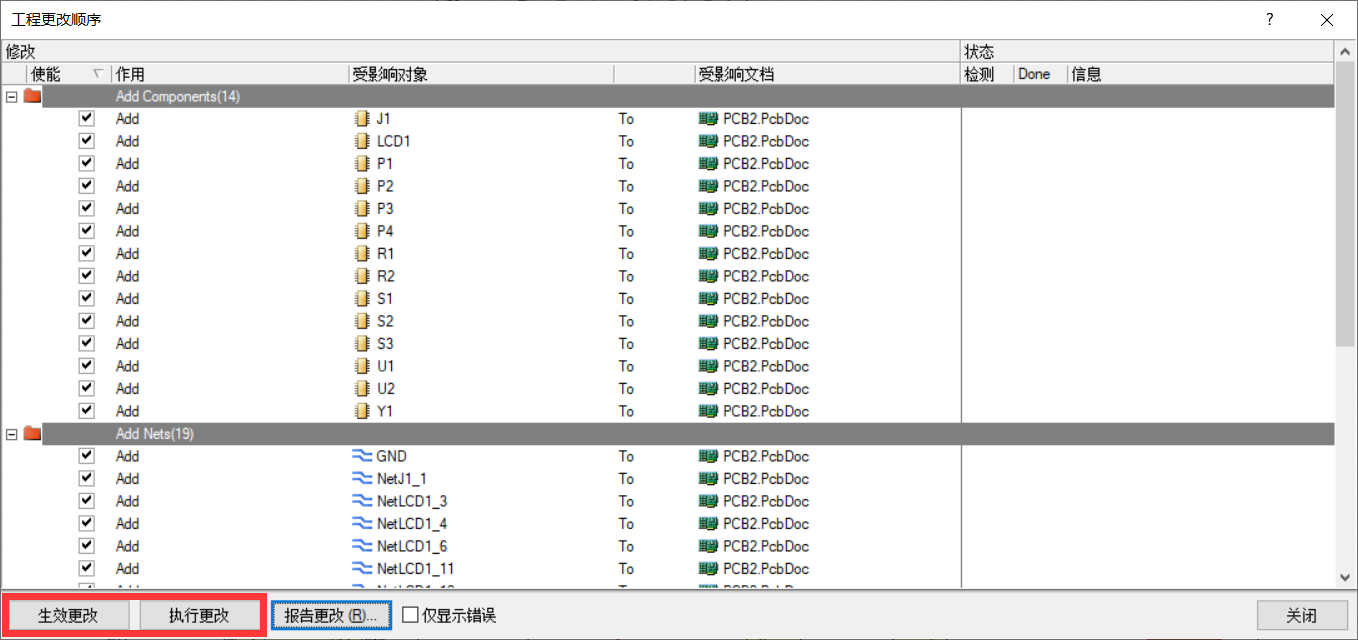


图6.3.1 导入原理图数据

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)：

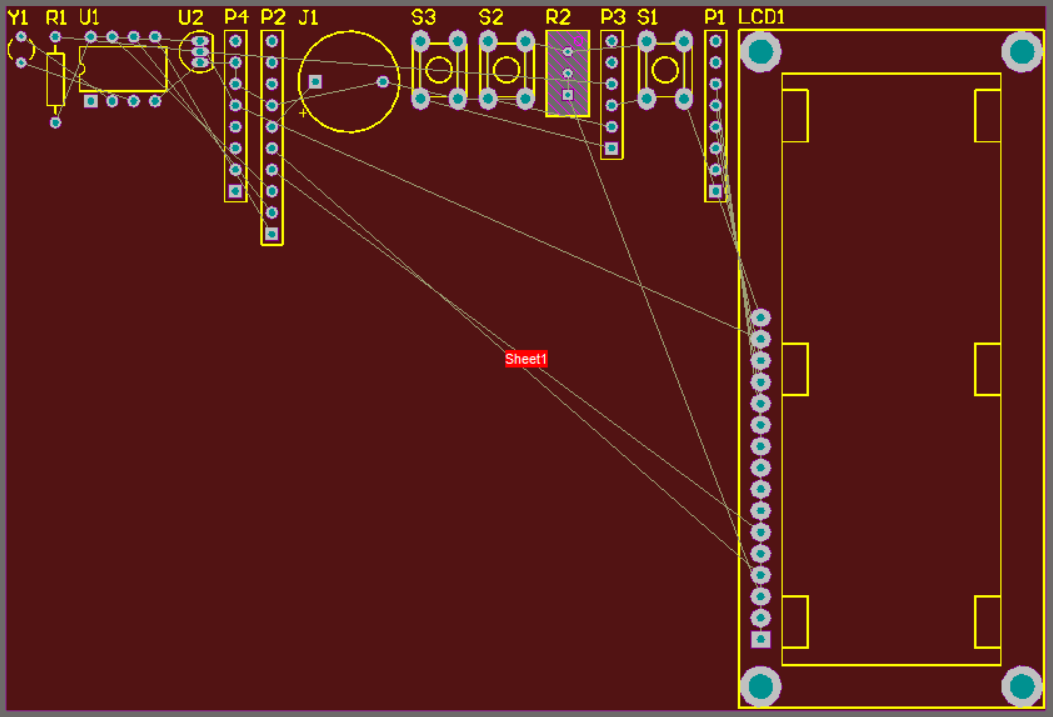


图6.3.2 PCB元件封装

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：

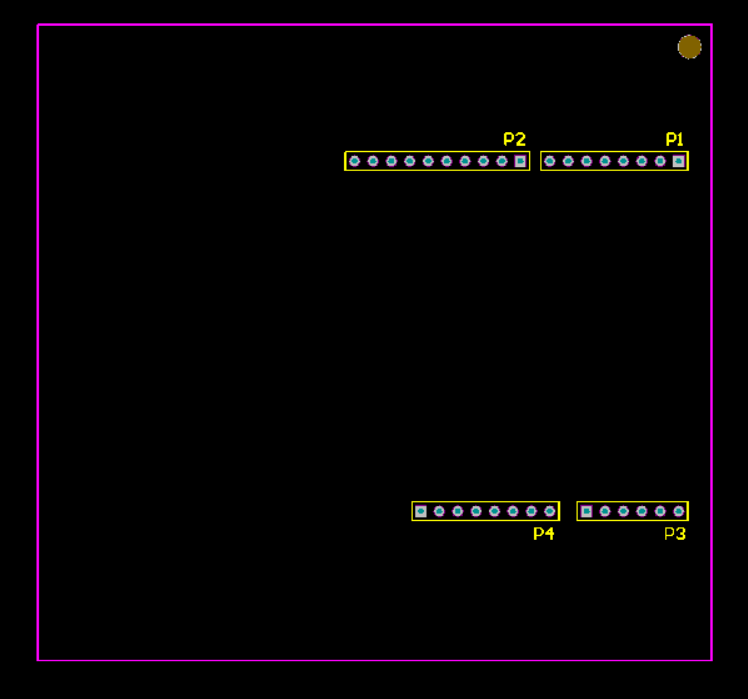


图6.3.3 PCB板形状

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：

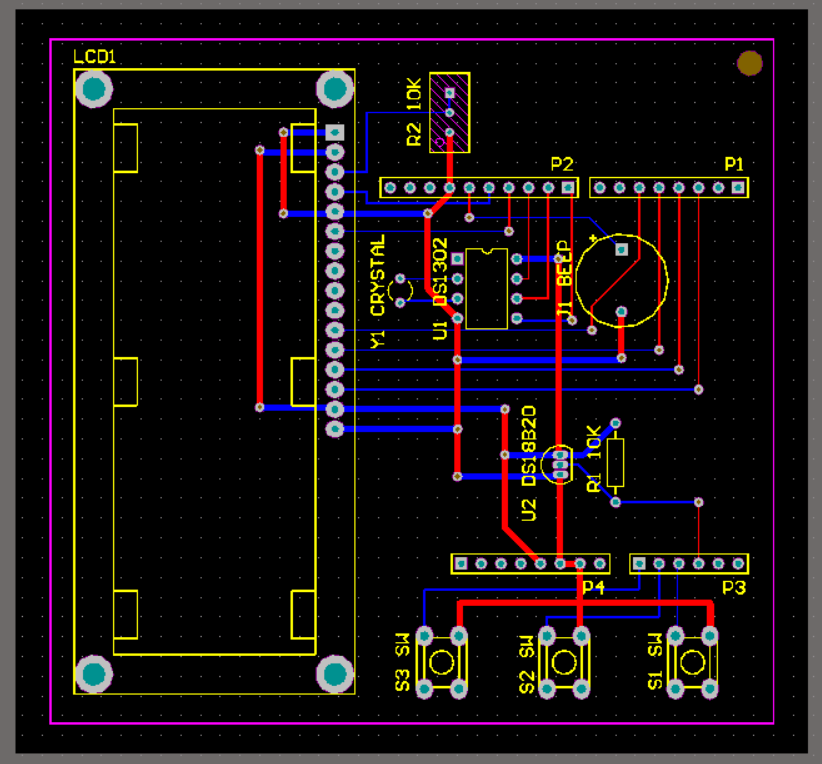


图6.3 PCB版图

**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。