

4.6 多功能数字时钟的设计与制作

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、实验设计方案

本项目使用 Arduino UNO 设计多功能数字时钟。硬件框图如图 1 所示，下面对各部分模块进行说明。

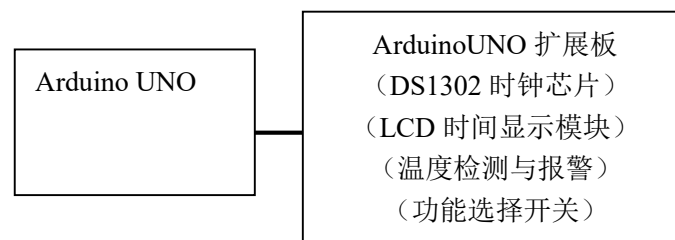


图 4.19 多功能数字时钟框图

DS1302 时钟芯片

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。详细内容，可查阅芯片的 Datasheet。

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，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多数模块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。有关 LCD1602 模块的内容，详见附录 1。

温度传感器

温度传感器的种类众多，在应用与高精度、高可靠性的场合时 DALLAS（达拉斯）公司生产的 DS18B20 温度传感器当仁不让。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。
- 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 $-55^{\circ}\text{C} \sim +125^{\circ}\text{C}$ ($-67^{\circ}\text{F} \sim +257^{\circ}\text{F}$)
- 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。
详细使用方法请查阅相关 Datasheet。

三、硬件设计

设计参考电路图如下：

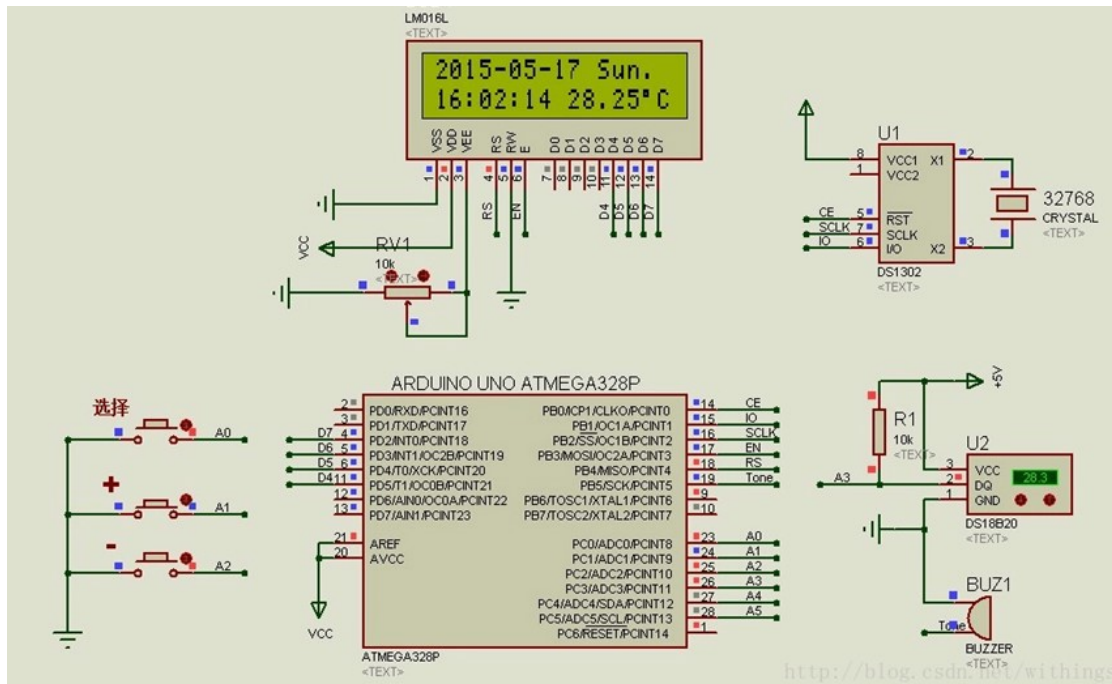


图 4.20 电路原理图

四、软件设计

设计参考程序如下：

```
/*
 * 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, frequency = 2093;
int alarm_hour = 7, alarm_minute = 30, alarm_second = 0; //闹钟时间
double Temperatures, Temp_Alarm = 30 ;

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(2015, 5, 16, 0, 2, 20, 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
}

/** 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;
        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){
        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,frequency);
        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); //断电将单片机的当前时间写到
DS1302 芯片中
    rtc.time(t);
}

```

五、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：

- (1) 显示时间、日期和星期
- (2) 断电保存时间
- (3) 通过按钮设置时间、日期
- (4) 整点响铃
- (5) 自定义闹钟
- (6) 显示温度
- (7) 自定义报警温度
- (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；
- (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。

2、设计的电路，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。

4、将制作的 Arduino UNO 扩展板与 Arduino UNO 板组装后，进行系统联调。