
实时时钟设计

设计者：STCode （公众号同名）

1) 实时时钟功能介绍:

该设计利用 DS1302 获取时间数据，用 4 位 led 数码管显示模块显示当前的年月日信息，并且设计有一个按键，可以切换时间数据。

2) 使用主要器件

- 1、Arduino Uno 控制板
- 2、TM1637 4 位数码管
- 3、DS1302 时钟模块
- 4、按键
- 5、导线

3) 元件介绍

1、DS1302 时钟模块

DS1302 是 DALLAS 公司推出的涓流充电时钟芯片，内含有一个实时时钟/日历和 31 字节静态 RAM，通过简单的串行接口与单片机进行通信。实时时钟/日历电路提供秒、分、时、日、周、月、年信息，每月的天数和闰年的天数可自行调整。时钟操作可通过 AM/PM 指示决定采用 24 或 12 小时的格式。DS1302 与单片机之间能简单地采用同步串行的方式进行通信，仅需用到三个口线：1、RST 复位 2、IO 数据线 3、SCLK 串行时钟。

2、TM1637 数码管

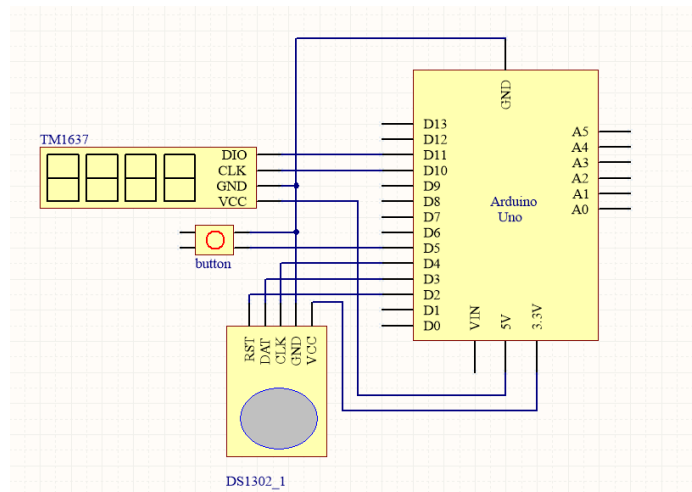
该模块是一个 12 脚的带时钟点的 4 位共阳数码管（0.36 英寸）的显示模块，驱动芯片为 TM1637，只需要两根信号线即可使用单片机控制 4 位 8 段数码管。

4) 电路设计

序号	Arduino Uno 板引脚号	DS1302 时钟模块
1	D2	RST
2	D3	DAT
3	D4	CLK
4	+3.3V	VCC
5	GND	GND

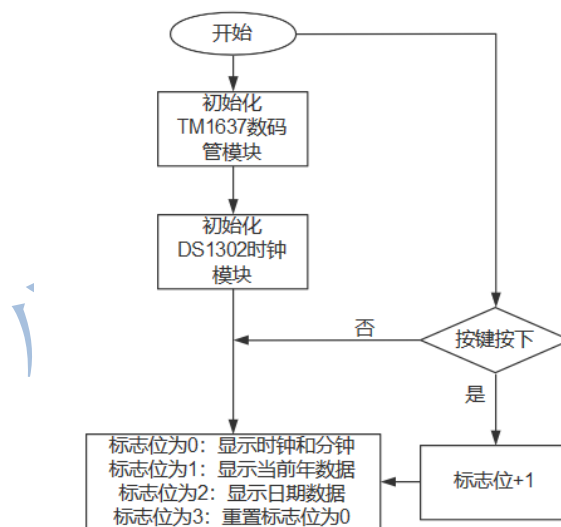
序号	Arduino Uno 板引脚号	TM1637 数码管
1	D10	CLK
2	D11	DIO
3	+5V	VCC
4	GND	GND

切换时间数据的按钮一端接 D5，另一端接 GND，整体简易接线图如下所示：



5) 程序设计

程序设计上首先是 DS1302 时钟模块获得当前的时间数据，进行数据处理后显示在搭载 TM1637 芯片的 4 位数码管上，一颗按键用来切换显示的时间数据，包括年、月、日、时、分、秒数据。程序流程如下：



6) 编译和测试

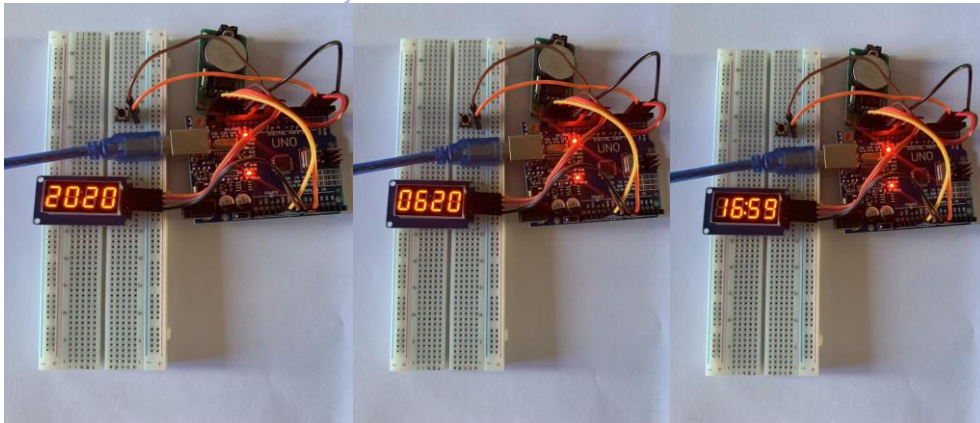
1、编译结果：

```
sketch_jun20a | Arduino 1.8.5
文件 编辑 项目 工具 帮助
sketch_jun20a $
41 tm1637.point(PPOINT_OFF); //关闭冒号
42 tm1637.display(0,tim.mon/10); //显示月
43 tm1637.display(1,tim.mon%10);
44 tm1637.display(2,tim.date/10); //显示日期
45 tm1637.display(3,tim.date%10);
46 }
47 }
48
49 void key_scan() //按键扫描子函数
50 {
51     if (key == 1) { //如果按键按下
        ...
    }
}

编译完成。

项目使用了 4552 字节，占用了 (14%) 程序存储空间。最大为 32256 字节。
全局变量使用了 352 字节，(17%) 的动态内存，余留 1696 字节局部变量。最大为 2048 字节。
```

2、测试截图和图片



7) 程序清单

```
#include <DS1302.h> //DS1302 头文件
#include <Wire.h> //Wire 头文件
#include "TM1637.h" //数码管头文件
#define CLK 10 //数码管 CLK 接口
```

```
#define DIO 11          //数码管 DIO 接口
TM1637 tm1637(CLK,DIO); //创建一个对象
DS1302 rtc(2, 3, 4); //RTC 的 RST,DAT,CLK
int key=5;             //按键引脚 5
int flag1=0;           //标志位 flag1 初始值为 0
void setup()
{
    tm1637.init();
    tm1637.set(BRIGHT_TYPICAL); //BRIGHT_TYPICAL = 2,BRIGHT_DARKEST = 0,BR
    IGHTEST = 7;
    Serial.begin(9600); //设置串口波特率
    pinMode(key,INPUT_PULLUP); //设置按键引脚为上拉电阻模式
    initRTCTime();       //初始化时钟
}

void loop()            //主函数
{
    key_scan();         //按键扫描子函数
    Time tim = rtc.time(); //获取时间数据
    if(flag1==0)        //当标志位为 0，显示时钟和分钟
    {
        tm1637.point(POINT_ON); //显示冒号
        tm1637.display(0,tim.hr/10); //显示时钟
        tm1637.display(1,tim.hr%10);
        tm1637.display(2,tim.min/10); //显示分钟
        tm1637.display(3,tim.min%10);
    }
    else if(flag1==1)    //当标志位为 1
    {
        tm1637.point(POINT_OFF); //关闭冒号
        tm1637.display(0,tim.yr/1000); //显示年
        tm1637.display(1,tim.yr/100%10);
        tm1637.display(2,tim.yr%100/10);
        tm1637.display(3,tim.yr%100%10);
    }
    else if(flag1==2)    //当标志位为 2
    {
        tm1637.point(POINT_OFF); //关闭冒号
        tm1637.display(0,tim.mon/10); //显示月
        tm1637.display(1,tim.mon%10);
        tm1637.display(2,tim.date/10); //显示日期
        tm1637.display(3,tim.date%10);
    }
}
```

```
void key_scan()    //按键扫描子函数
{
    if(digitalRead(key)==LOW)    //假如按键按下
    {
        delay(20);                //延时 20ms
        if(digitalRead(key)==LOW)    //当按键确实按下
        {
            flag1++;                //标志位+1
            if(flag1==3) flag1=0;    //当标志位为 3，归 0
            while(digitalRead(key)==LOW){};
        }
    }
}

void initRTCTime(void)//初始化时钟子函数
{
    rtc.writeProtect(false);    //关闭写保护
    rtc.halt(false);            //清除时钟停止标志
    Time TIME(2020, 6, 20, 16, 58, 30, 7); //建立时间对象
    rtc.time(TIME);//向 DS1302 设置时间数据
}
```