****单片机课程设计

**可调式电子日历设计**

|  |  |
| --- | --- |
| **专业** | 自动化 |
| **学生姓名** | 张航 |
| **班级** | B自动化181 |
| **学号** | 180106051026 |
| **完成日期** | 2021.6.24 |

盐城工学院电气学院

**摘要：**电子日历的功能可实现年、月、日、时、分、秒和星期的显示。随着社会的发展电子日历的功能越来越强大，精确度越来越高。本次设计的电子日历采用 DS1302时钟芯片，该芯片精确度高，性能可靠。用LCD1602液晶屏，既可以显示数字又可以显示中文，方便了人们的生活。本设计总体分为硬件部分和软件部分。硬件部分可分为：时钟芯片 DS1302、STC89C52和LCD1602等。本设计非常适合家庭使用。电源采用＋5V电压供电,走时精确，界面清爽，方便人们及时知晓当前日期和时间，为人们的生活和出行提供了极大的便利，是现代家庭必备的设备之一。

**关键词：**单片机 , 电子日历， DS1302时钟芯片，LCD1602液晶屏

**目 录**

1.概述

1.1 课题背景------------------------------------------------------4

1.2 课题主要内容--------------------------------------------------4

2.系统硬件设计

2.1 STC单片机最小系统电路设计-------------------------------------5

2.2 LCD显示电路---------------------------------------------------5

2.3 DS1302时钟电路

2.31 DS1302的介绍---------------------------------------------6

2.32 DS1302时序设计-------------------------------------------7

3.软件设计

3.1 软件开发工具介绍-----------------------------------------------8

3.2 程序流程图-----------------------------------------------------8

3.3 程序代码设计

3.3.1 DS1302时钟程序-------------------------------------------8

3.3.2 LCD1602显示驱动程序--------------------------------------11

3.3.3 定时器配置------------------------------------------------13

3.3.4 主程序设置------------------------------------------------14

4.运行与调试

4.1 proteus调试图-----------------------------------------------14

4.2 实物调试图---------------------------------------------------15

5.课程设计体会---------------------------------------------------------15

参考文献---------------------------------------------------------------16

附录1：程序清单-------------------------------------------------------17

附录2：设计图纸-------------------------------------------------------23

附录3：元器件目录表---------------------------------------------------24

**1.概述**

**1.1课题背景**

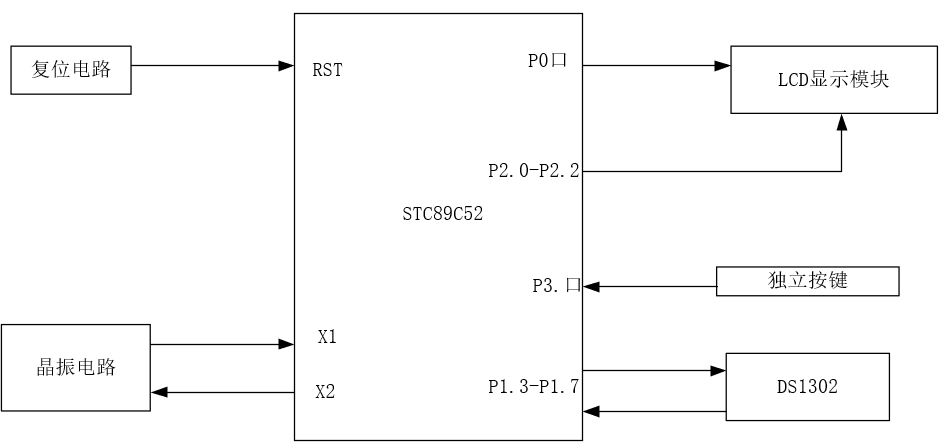
单片机运用在各种智能设备当中，比如平衡车、扫地机器人等。设备智能化是一个必然的趋势，而要实现设备智能化离不开单片机的支持，所以学习单片机是每一位自动化学子的必修之路。之前虽然已经学过单片机的理论课，但是理论学好了，不一定就能学以致用，而为了达到这个目的就要进行单片机的实践训练。可调式电子日历设计对于学习单片机十分的契合，完成这个项目需要用到单片机的定时器/计数器功能以及驱动外部模块。而这正是单片机运用的最重要的两个部分之一，学好了定时器/计数器以及懂得如何驱动外部模块，那么很多单片机的应用就能举一反三了。可调式电子日历小巧便于携带，界面清爽，一目了然，方便的知晓当前时间， 并可对时间做出修改， 实在是居家旅行日常生活之必备用品。在现今的可调式电子日历具有性能稳定、精确度高、成本低、易于产品化，以及方便、实用等特点。

**1.2课题主要内容**

设计并制作一个可调式电子日历，开机后可以在LCD液晶显示器上显示当前的年、月、日、星期、时、分、秒等信息，按下K1键，选择对年、月、日、星期、时、分、秒的设置；按下K2键，进行加设置；按下K3键，进行减设置；按下K4键，对设置进行确认。可调式电子日历，可以显示日期和时间信息，按下按键K1-K4，可设置与调整时间。

**2.系统硬件设置**

根据课题设计的基本方案，本系统主要有如下功能电路：单片机最小系统电路，DS1302时钟电路，LCD液晶显示电路，以及独立按键。结构框图如图一所示：



图一

**2.1 STC单片机最小系统电路设计**

本设计采用STC89C52单片机作为主控芯片，作为新生代单片机在抗干扰能力更加稳定，同时它的运行速度更快，功率更低，这让用户有了更多的打算。单片机最小系统包括复位电路、晶振电路。

**A. 复位电路**

复位电路的目的是初始化主控芯片，让芯片重新开始工作。复位时，单片机重新从程序存储器中起始位置开始读程序代码，单片机运行中一旦出现死机或是程序跑飞的状况时，就需要复位。本设计中复位电路采用自动复位+手动复位的方式，系统上电，则系统自动复位，复位采用简单的阻容方式，设置RC的值可以确定复位时间。手动复位方式就是通过按键来进行操作。

复位电路本质是电容的充放电，该电路中有一个大小为10uF的电容和一个10KΩ左右的电阻，要想单片机可以稳定实现复位动作，必要使得复位引脚的高电平至少持续两个机器周期，因此选取器件的时候一定要注意参数。

**B. 晶振电路**

晶振的部分配合2个22pF的电容实现了振荡，产生的电脉冲为单片机提供时钟信号。本系统采用12MHz的晶振。

**2.2 LCD显示电路**

本系统中应用1602液晶显示器作为显示器件。由于1602具有不需要持续刷新的特性，故只需要对数据口进行一次输入，显示屏上就会保持持续点亮。因此，液晶显示器的画质更高而且不会出现闪烁。液晶显示器除了体积小、重量轻、功耗低等优点外并且还都是数字式的，这使得单片机系统的接口变得更加简单可靠，操作也更加方便。

液晶显示器的主要技术参数：

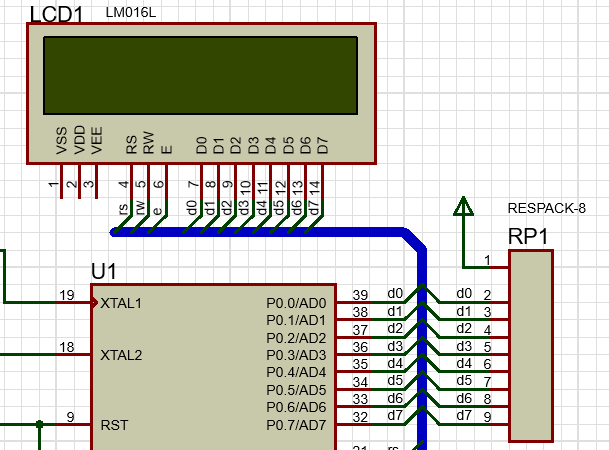
a) 显示屏显示容量：16×2个字符；

b) 器件工作电压：3.5-5.5V；

c) 器件工作电流：2.0mA(5.0V)；

d) 显示字符尺寸：2.95×3.35(W×H)mm；

1602液晶显示模块与单片机的P0口经过10K的上拉电阻直接连接，其电路如图2所示。（实际电路由于供电电流够大，所以无需接上拉电阻，并且实际电路需要接背光电源，以及在VO引脚处接一个滑动电阻用来调节对比度）



图二

**2.3 DS1302时钟电路**

**2.31 DS1302的介绍**

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。DS1302 用于数据记录，特别是对某些具有特殊意义的数据点的记录上，能实现数据与出现该数据的时间同时记录，因此广泛应用于测量系统中。



DS1302外部引脚图

是后备电源引脚，是主电源引脚，主电源关闭时，在后备电源作用下也能保持时钟的持续运行，DS1302由和两者中较大者供电，GND为接地端。



DS1302内部结构图

是复位/片选线。通过将输入驱动置高电平来启动所有的数据传送，输入有两种功能：首先，接通控制逻辑，允许地址/命令序列送入移位寄存器；其次，提供终止单字节或多字节数据的传送手段。当为高电平时，所有的数据传送被初始化，允许对DS1302进行操作。如果在传送过程中置为低电平，则会终止此次的数据传送，I/O引脚变为高阻态。上电运行时，在Vcc>2.0V之前，必须保持低电平。只有在SCLK为低电平时，才能将置为高电平。

I/O为串行数据输入输出端（双向）；SCLK（Serial clock）为时钟输入端；X1和X2是震荡源，外接32.768kHz晶振。

DS1302与单片机通信时仅需要三条引线，即复位线（Reset）、串行时钟线SCLK，输出线I/O(数据线)。通过三条引线串行访问DS1302的日期/时钟/配置/RAM等寄存器，即可以实现DS1302的所有操作。

**2.32 DS1302时序设计**

要想与DS1302通信，首先要先了解DS1302的控制字。DS1302的控制字如图三所示。



图三

控制字的最高有效位（位7）必须是逻辑 1，如果它为0，则不能把数据写入到DS1302中。位 6：如果为0，则表示存取日历时钟数据，为1表示存取 RAM数据；位5至位1（A4～A0）：指示操作单元的地址；位0（最低有效位）：如为0，表示要进行写操作，为 1表示进行读操作。

由于仿真运行时DS1302组件总是默认从PC获取当前时钟，因而在Proteus中无法观察到这个差异。

**3．软件设计**

**3.1 软件开发工具介绍**

A.Keil C51简介。Keil C51软件是一款可以用C语言来编程的51系列兼容开发软件，和传统的汇编来说，它有更多的优势，例如它拥有更加强大的功能、更加完整的结构、可读性更强，这些优势使得它学习起来更加的容易。于此同时它还提供了数量庞大的库函数和一些开发调试工具等，通过编译完的汇编代码，更能体会出这款软件的优越性。

**3.2 程序流程图**



**3.3 程序代码设计**

**3.3.1 DS1302时钟程序**

#include <reg52.h> //52系列单片机头文件

#include <string.h>

#include<intrins.h>

#define uchar unsigned char //宏定义

#define uint unsigned int

sbit SDA=P1^0; // DS1302数据线

sbit CLK=P1^1; //DS1302时钟线

sbit RST=P1^2; //DS1302复位线

char Adjust\_Index=-1; //当前调节的时间：秒，分，时，日，

void Display\_LCD\_String(uchar p,uchar \*s);

uchar tCount=0;//一年中每个月的天数，2月的天数由年份决定

uchar MonthsDays[]={0,31,0,31,30,31,30,31,31,30,31,30,31};

uchar \*WEEK[]={"SUN","MON","TUS","WEN","THU","FRI","SAT"};//周日，周一到周六

void DS1302\_Write\_Byte(uchar x) //向DS1302写入一字节

{ uchar i;

for(i=0;i<8;i++)

{SDA=x&1;

CLK=1;

CLK=0;

x>>=1;

}

}

uchar DS1302\_Read\_Byte() //从DS1302读取一字节

{ uchar i,b,t;

for(i=0;i<8;i++)

{ b>>=1;

t=SDA;

b|=t<<7;

CLK=1;

CLK=0; }

return b/16\*10+b%16; }

uchar Read\_Data(uchar addr) //从DS1302指定位置读数据

{ uchar dat;

RST=0;

CLK=0;

RST=1;

DS1302\_Write\_Byte(addr);

dat=DS1302\_Read\_Byte();

CLK=1;

RST=0;

return dat;}

void Write\_DS1302(uchar addr,uchar dat) //向DS1302指定位置写入数据

{ CLK=0;

RST=1;

DS1302\_Write\_Byte(addr);

DS1302\_Write\_Byte(dat);

CLK=0;

RST=0; }

void SET\_DS1302() //设置时间

{ uchar i;

Write\_DS1302(0x8e,0x00);

for(i=0;i<7;i++)

{

Write\_DS1302(0x80+2\*i,(DateTime[i]/10<<4|(DateTime[i]%10)));

}

Write\_DS1302(0x8e,0x80);

}

void GetTime() //读取当前时期时间

{ uchar i;

for(i=0;i<7;i++)

{

DateTime[i]=Read\_Data(0x81+2\*i);

}

}

void Format\_DateTime(uchar d,uchar \*a) //时间和日期转换成数字字符

{

a[0]=d/10+'0';a[1]=d%10+'0';

}

uchar isLeapYear(uint y) //判断是否为闰年

{

return (y%4==0&&y%100!=0)||(y%400==0);

}

void RefreshWeekDay() //星期转换

{ if(K1==0) //选择调整对象：年，月，日，时，分，秒

{ while(K1==0);

if(Adjust\_Index==-1||Adjust\_Index==1) Adjust\_Index=7;

Adjust\_Index--;

if(Adjust\_Index==5) Adjust\_Index=4;

LCD\_DSY\_BUFFER2[13]='[';

LCD\_DSY\_BUFFER2[14]=Change\_Flag[Adjust\_Index];

LCD\_DSY\_BUFFER2[15]=']'; }

else if(K2==0) // 加

{

while (K2==0); DateTime\_Adjust(1);

}

else if (K3==0) // 减

{

while (K3==0); DateTime\_Adjust(-1);

}

else if (K4==0) // 确定

{ while(K4==0);

SET\_DS1302(); //调整后的时间写入DS1302

LCD\_DSY\_BUFFER2[13]=' ';

LCD\_DSY\_BUFFER2[14]=' ';

LCD\_DSY\_BUFFER2[15]=' ';

Adjust\_Index=-1; }

}

**3.3.2 LCD1602显示驱动程序**

对LCD1602进行控制，需要查阅对应的数据手册，各引脚功能如下图。



RS为lcd数据命令选择，RS为L则写命令 H则写数据，R/W为读写控制信号一般我们只写而不读取LCD内的数据。E为使能端口，当EN脚产生下降沿时LCD可接收信号，BLA和BLK为背光电源，如果不接电源LCD显示的信息将会看不清。

V0接入10K电阻分压可调节对比度，如果电压不合适，同样无法正常显示。送入LCD的信号分为控制信号和数据信号两种，不同信号的区分取决于RS引脚的电平，所以定义write\_com函数来实现LCD的指令操作。对数据的处理函数如下：

#include <reg52.h> //52系列单片机头文件

#include <string.h>

#include<intrins.h>

#define uchar unsigned char //宏定义

#define uint unsigned int

uchar Read\_LCD\_State() //读LCD1602状态

{ uchar state;

RS=0;

RW=1;

EN=1;

DelayMS(1);

state=P0;

EN=0;

DelayMS(1);

return state;

}

void LCD\_Busy\_Wait()

{

while((Read\_LCD\_State()&0x08)==0x80);

DelayMS(5);

}

void Write\_LCD\_Data(uchar dat) //向LCD1602写数据

{

LCD\_Busy\_Wait();

RS=1;

RW=0; //选择写数据模式

EN=0;

P0=dat; //将要写的数据送入数据总线

EN=1; //将使能端置给一个高脉冲

DelayMS(1);

EN=0; //将使能端置0以完成高脉冲

}

void Write\_LCD\_Command(uchar cmd)// 向LCD1602写命令

{

LCD\_Busy\_Wait();

RS=0;

RW=0;//选择写命令模式

EN=0;

P0=cmd;//将要写的命令字送入数据总线

EN=1;//将使能端置给一个高脉冲

DelayMS(1);

EN=0; //将使能端置0以完成高脉冲

}

void Init\_LCD() //LCD初始化

{ Write\_LCD\_Command(0x38); DelayMS(1);

Write\_LCD\_Command(0x01); DelayMS(1);

Write\_LCD\_Command(0x06); DelayMS(1);

Write\_LCD\_Command(0x0C); DelayMS(1);

Display\_LCD\_String(0x00," dian zi ri li ");

DelayMS(2000);

}

void DateTime\_Adjust(char x) //年月日时分秒++/--

{ switch (Adjust\_Index)

{ case 6: //年

if(x== 1&&DateTime[6]<99) DateTime[6]++;

if(x==-1&&DateTime[6]>0) DateTime[6]--;

MonthsDays[2]=isLeapYear(2000+DateTime[6])? 29:28;

if(DateTime[3]>MonthsDays[DateTime[4]])

DateTime[3]=MonthsDays[DateTime[4]];

RefreshWeekDay();

break;

case 4: //月

if(x== 1&&DateTime[4]<12) DateTime[4]++;

if(x==-1&&DateTime[4]>1) DateTime[4]--;

MonthsDays[2]=isLeapYear(2000+DateTime[6])? 29:28;

if(DateTime[3]>MonthsDays[DateTime[4]])

DateTime[3]=MonthsDays[DateTime[4]];

RefreshWeekDay();

break;

case 3: //日

MonthsDays[2]=isLeapYear(2000+DateTime[6])? 29:28;

if(x== 1&&DateTime[3]<MonthsDays[DateTime[4]]) DateTime[3]++;

if(x==-1&&DateTime[3]>0) DateTime[3]--;

RefreshWeekDay();

break;

case 2: //时

if(x== 1&&DateTime[2]<23) DateTime[2]++;

if(x==-1&&DateTime[2]>0) DateTime[2]--;

break;

case 1: //秒

if(x== 1&&DateTime[1]<59) DateTime[1]++;

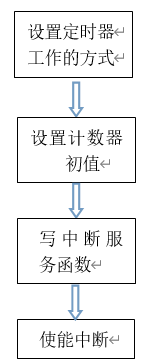
if(x==-1&&DateTime[1]>0) DateTime[1]--;

break;}

}

**3.3.3 定时器配置**

要实现电子时钟的显示与设置功能，首先要知道单片机定时器的工作原理及整个配置过程。定义一个init函数用来初始化定时器和LCD，由于时钟最小显示时间为1秒，而定时器最大定时时间为六十多毫秒，所以需要定义一个定时器溢出次数的变量，设置定时器溢出的时间为50MS，当溢出次数为20次时，时间刚好为1秒，定时器初始化见下图。



TMOD = 0x01; //定时器0工作方式1

TH0=(65536-45872)/256;//定时器0 16位计数器高8位初始计数

TL0=(65536-45872)%256;//定时器0 16位计数器低8位初始计数

EA=1; //中断使能 中端总开关

ET0=1; //定时器0 中断允许位

TR0=1; //初始中断开始启动

**3.3.4 主程序设置**

#include <reg52.h> //52系列单片机头文件

#include <string.h>

#include<intrins.h>

#define uchar unsigned char //宏定义

#define uint unsigned int

void main() //主程序

{ Init\_LCD(); //LCD初始化

IE=0x83; // 允许INT0，T0中断

IP=0x01;

IT0=0x01;

TMOD=0x01;//设置定时器0为工作方式1（M1M0为01）

TH0=-50000/256; //装初值

TL0=-50000%256;

TR0=1;

while(1)

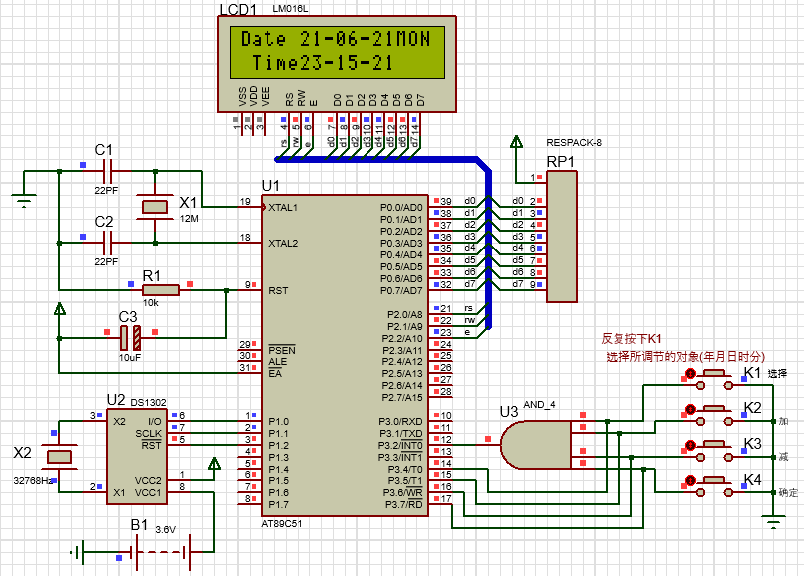
{

if(Adjust\_Index==-1) GetTime();

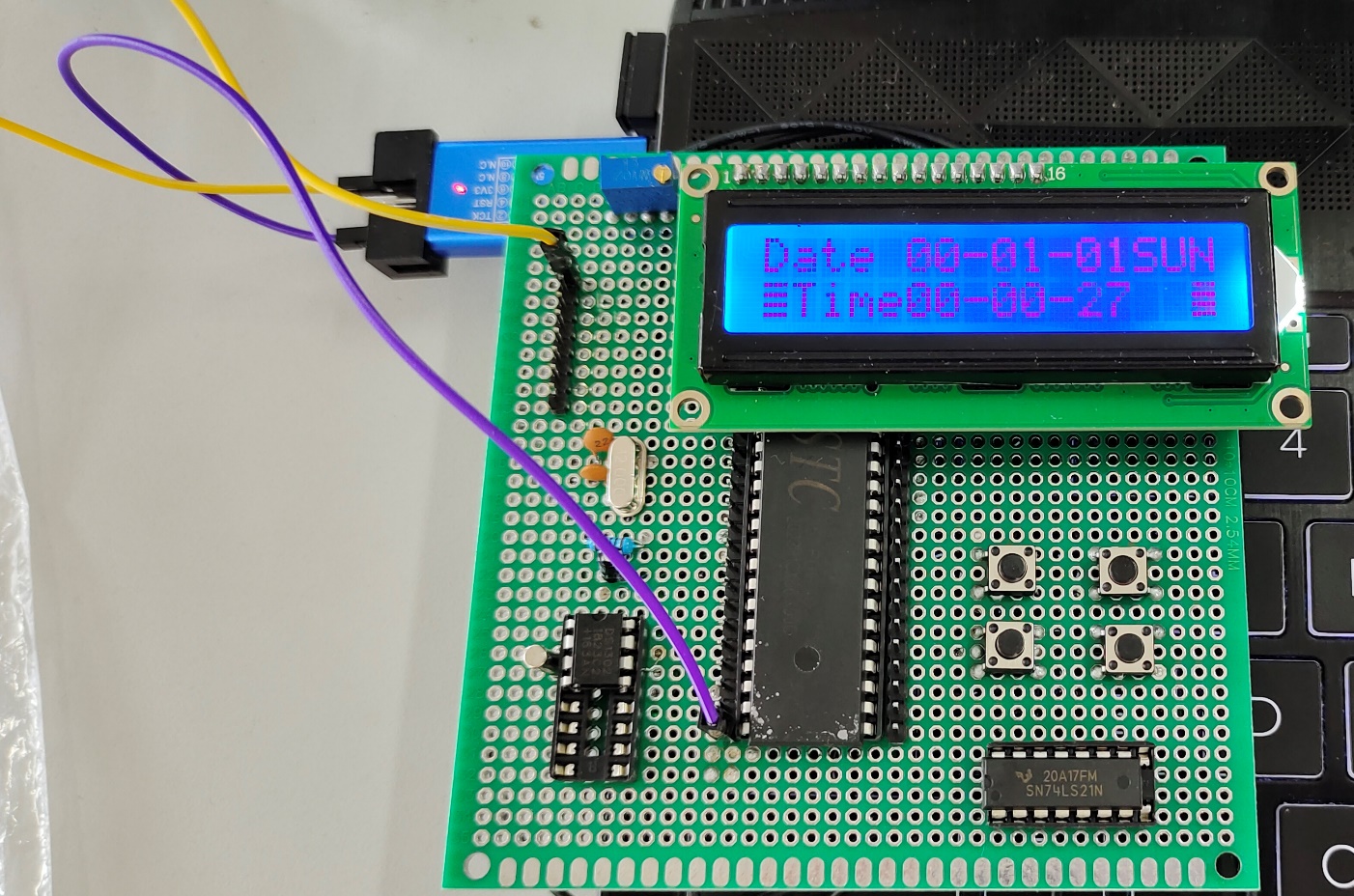
}

}

**4.运行和调试**

**4.1 proteus调试图**

**4.2 实物调试图**



**5.课程设计体会**

本次课程设计让我对51单片机有了更深的理解，通过之前半个多学期的理论学习，终于有了实践的机会，由于之前也学习过两种单片机所以上手51并不困难。虽然我学过单片机但是之前都是比较浅显的理解和学习，只知道怎么用以及如何在网上查找例子，并不清楚具体的配置过程以及单片机内部的一些电路原理。通过这次课设，我学会了如何使用基姆拉尔森公式算星期几，并且使用巧妙的方法来计算每个月的天数。看似简单的电子时钟，其实程序也并不简单，这让我知道自己还需努力，通过多做项目来提高自己的水平是一个很好的路径，对于自己未来的学习也能奠定很好的基础，希望在之后的日子里我能创造出更多有意思的作品。

**参考文献**

【1】 刘瑞新 . 单片机原理及应用教程 . 北京 . 机械工业出版社 ,2003

【2】 李群芳 .单片微型计算机与接口技术 .电子工业出版 ,2012

【3】 审庆阳 .郭庭杰 .单片机的实践与应用 .清华大学出版 ,2002

【4】 沙占友 .王彦朋 .孟志永 .单片机外围电路设计 .北京 .电子工业出版社 ,2003

【5】张兰红，陆广平，仓思雨. 单片机课程设计仿真与实践指导[M].北京：机械工业出版社，2018

**附录1：程序清单**

#include <reg52.h> //52系列单片机头文件

#include <string.h>

#include<intrins.h>

#define uchar unsigned char //宏定义

#define uint unsigned int

sbit SDA=P1^0; // DS1302数据线

sbit CLK=P1^1; //DS1302时钟线

sbit RST=P1^2; //DS1302复位线

sbit RS=P2^0; //数据命令选择端（H/L）

sbit RW=P2^1; //读写选择端（H/L）

sbit EN=P2^2; //使能信号

sbit K1=P3^4; // 选择

sbit K2=P3^5; // 加

sbit K3=P3^6; // 减

sbit K4=P3^7; // 确定

void Display\_LCD\_String(uchar p,uchar \*s);

uchar tCount=0;//一年中每个月的天数，2月的天数由年份决定

uchar MonthsDays[]={0,31,0,31,30,31,30,31,31,30,31,30,31};

uchar \*WEEK[]={"SUN","MON","TUS","WEN","THU","FRI","SAT"};//周日，周一到周六

uchar LCD\_DSY\_BUFFER1[]={"Date 00-00-00 "}; //LCD显示缓冲

uchar LCD\_DSY\_BUFFER2[]={" Time00-00-00 "};

uchar DateTime[7]; //所读取的日期时间

char Adjust\_Index=-1; //当前调节的时间：秒，分，时，日，

uchar Change\_Flag[]="-MHDM-Y";

uchar Read\_LCD\_State();

void LCD\_Busy\_Wait();

void Write\_LCD\_Data(uchar dat);

void Write\_LCD\_Command(uchar cmd);

void Init\_LCD();

void Set\_LCD\_POS(uchar p);

void DelayMS(uint x) //延时程序

{ uchar i;

while(x--) for(i=0;i<120;i++); }

uchar Read\_LCD\_State() //读LCD1602状态

{ uchar state;

RS=0;

RW=1;

EN=1;

DelayMS(1);

state=P0;

EN=0;

DelayMS(1);

return state;

}

void LCD\_Busy\_Wait()

{

while((Read\_LCD\_State()&0x08)==0x80);

DelayMS(5);

}

void Write\_LCD\_Data(uchar dat) //向LCD1602写数据

{

LCD\_Busy\_Wait();

RS=1;

RW=0; //选择写数据模式

EN=0;

P0=dat; //将要写的数据送入数据总线

EN=1; //将使能端置给一个高脉冲

DelayMS(1);

EN=0; //将使能端置0以完成高脉冲

}

void Write\_LCD\_Command(uchar cmd)// 向LCD1602写命令

{

LCD\_Busy\_Wait();

RS=0;

RW=0;//选择写命令模式

EN=0;

P0=cmd;//将要写的命令字送入数据总线

EN=1;//将使能端置给一个高脉冲

DelayMS(1);

EN=0; //将使能端置0以完成高脉冲

}

void Init\_LCD() //LCD初始化

{ Write\_LCD\_Command(0x38); DelayMS(1);

Write\_LCD\_Command(0x01); DelayMS(1);

Write\_LCD\_Command(0x06); DelayMS(1);

Write\_LCD\_Command(0x0C); DelayMS(1);

Display\_LCD\_String(0x00," dian zi ri li ");

DelayMS(2000);

}

void Set\_LCD\_POS(uchar p)

{

Write\_LCD\_Command(p|0x80);

}

void Display\_LCD\_String(uchar p,uchar \*s)

{ uchar i;

Set\_LCD\_POS(p);

for(i=0;i<16;i++)

{ Write\_LCD\_Data(s[i]);

DelayMS(1);}

}

void DS1302\_Write\_Byte(uchar x) //向DS1302写入一字节

{ uchar i;

for(i=0;i<8;i++)

{SDA=x&1;

CLK=1;

CLK=0;

x>>=1;

}

}

uchar DS1302\_Read\_Byte() //从DS1302读取一字节

{ uchar i,b,t;

for(i=0;i<8;i++)

{ b>>=1;

t=SDA;

b|=t<<7;

CLK=1;

CLK=0; }

return b/16\*10+b%16; }

uchar Read\_Data(uchar addr) //从DS1302指定位置读数据

{ uchar dat;

RST=0;

CLK=0;

RST=1;

DS1302\_Write\_Byte(addr);

dat=DS1302\_Read\_Byte();

CLK=1;

RST=0;

return dat;}

void Write\_DS1302(uchar addr,uchar dat) //向DS1302指定位置写入数据

{ CLK=0;

RST=1;

DS1302\_Write\_Byte(addr);

DS1302\_Write\_Byte(dat);

CLK=0;

RST=0; }

void SET\_DS1302() //设置时间

{ uchar i;

Write\_DS1302(0x8e,0x00);

for(i=0;i<7;i++)

{

Write\_DS1302(0x80+2\*i,(DateTime[i]/10<<4|(DateTime[i]%10)));

}

Write\_DS1302(0x8e,0x80);

}

void GetTime() //读取当前时期时间

{ uchar i;

for(i=0;i<7;i++)

{

DateTime[i]=Read\_Data(0x81+2\*i);

}

}

void Format\_DateTime(uchar d,uchar \*a) //时间和日期转换成数字字符

{

a[0]=d/10+'0';a[1]=d%10+'0';

}

uchar isLeapYear(uint y) //判断是否为闰年

{

return (y%4==0&&y%100!=0)||(y%400==0);

}

void RefreshWeekDay() //星期转换

{ uint i,d,w=5;

for(i=2000;i<2000+DateTime[6];i++)

{ d=isLeapYear(i) ? 366 : 365;

w=(w+d)%7; }

d=0;

for(i=1;i<DateTime[4];i++) d+=MonthsDays[i];

d+=DateTime[3];

DateTime[5]=(w+d)%7+1;

}

void DateTime\_Adjust(char x) //年月日时分秒++/--

{ switch (Adjust\_Index)

{ case 6: //年

if(x== 1&&DateTime[6]<99) DateTime[6]++;

if(x==-1&&DateTime[6]>0) DateTime[6]--;

MonthsDays[2]=isLeapYear(2000+DateTime[6])? 29:28;

if(DateTime[3]>MonthsDays[DateTime[4]])

DateTime[3]=MonthsDays[DateTime[4]];

RefreshWeekDay();

break;

case 4: //月

if(x== 1&&DateTime[4]<12) DateTime[4]++;

if(x==-1&&DateTime[4]>1) DateTime[4]--;

MonthsDays[2]=isLeapYear(2000+DateTime[6])? 29:28;

if(DateTime[3]>MonthsDays[DateTime[4]])

DateTime[3]=MonthsDays[DateTime[4]];

RefreshWeekDay();

break;

case 3: //日

MonthsDays[2]=isLeapYear(2000+DateTime[6])? 29:28;

if(x== 1&&DateTime[3]<MonthsDays[DateTime[4]]) DateTime[3]++;

if(x==-1&&DateTime[3]>0) DateTime[3]--;

RefreshWeekDay();

break;

case 2: //时

if(x== 1&&DateTime[2]<23) DateTime[2]++;

if(x==-1&&DateTime[2]>0) DateTime[2]--;

break;

case 1: //秒

if(x== 1&&DateTime[1]<59) DateTime[1]++;

if(x==-1&&DateTime[1]>0) DateTime[1]--;

break;}

}

void T0\_INT() interrupt 1 //定时器0中断，定时器每秒刷新LCD显示

{ TH0=-50000/256;

TL0=-50000%256;

if(++tCount!=2) return;

tCount=0;

Format\_DateTime(DateTime[6],LCD\_DSY\_BUFFER1+5);

Format\_DateTime(DateTime[4],LCD\_DSY\_BUFFER1+8);

Format\_DateTime(DateTime[3],LCD\_DSY\_BUFFER1+11);

strcpy(LCD\_DSY\_BUFFER1+13,WEEK[DateTime[5]-1]);

Format\_DateTime(DateTime[2],LCD\_DSY\_BUFFER2+5);

Format\_DateTime(DateTime[1],LCD\_DSY\_BUFFER2+8);

Format\_DateTime(DateTime[0],LCD\_DSY\_BUFFER2+11);

Display\_LCD\_String(0x00,LCD\_DSY\_BUFFER1);

Display\_LCD\_String(0x40,LCD\_DSY\_BUFFER2); }

void EX\_INT0() interrupt 0 //键盘中断

{ if(K1==0) //选择调整对象：年，月，日，时，分，秒

{ while(K1==0);

if(Adjust\_Index==-1||Adjust\_Index==1) Adjust\_Index=7;

Adjust\_Index--;

if(Adjust\_Index==5) Adjust\_Index=4;

LCD\_DSY\_BUFFER2[13]='[';

LCD\_DSY\_BUFFER2[14]=Change\_Flag[Adjust\_Index];

LCD\_DSY\_BUFFER2[15]=']'; }

else if(K2==0) // 加

{

while (K2==0); DateTime\_Adjust(1);

}

else if (K3==0) // 减

{

while (K3==0); DateTime\_Adjust(-1);

}

else if (K4==0) // 确定

{ while(K4==0);

SET\_DS1302(); //调整后的时间写入DS1302

LCD\_DSY\_BUFFER2[13]=' ';

LCD\_DSY\_BUFFER2[14]=' ';

LCD\_DSY\_BUFFER2[15]=' ';

Adjust\_Index=-1; }

}

void main() //主程序

{ Init\_LCD(); //LCD初始化

IE=0x83; // 允许INT0，T0中断

IP=0x01;

IT0=0x01;

TMOD=0x01;//设置定时器0为工作方式1（M1M0为01）

TH0=-50000/256; //装初值

TL0=-50000%256;

TR0=1;

while(1)

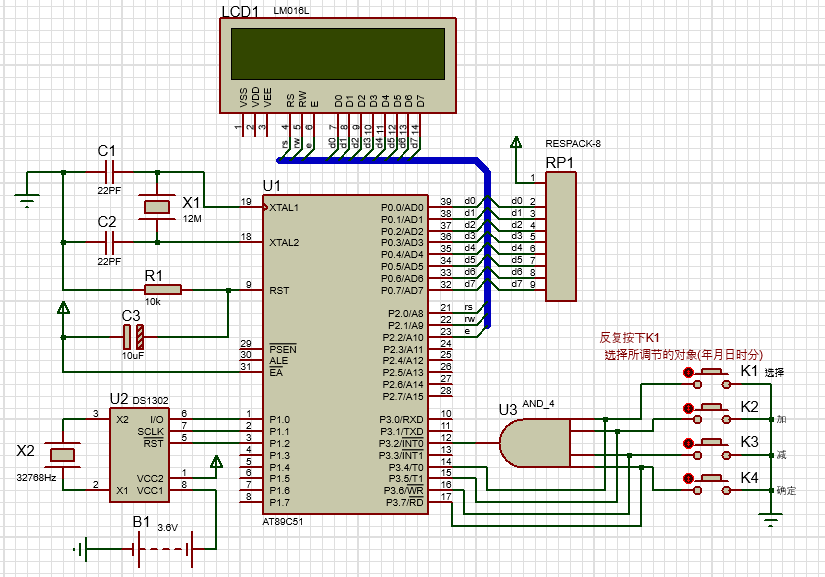
{

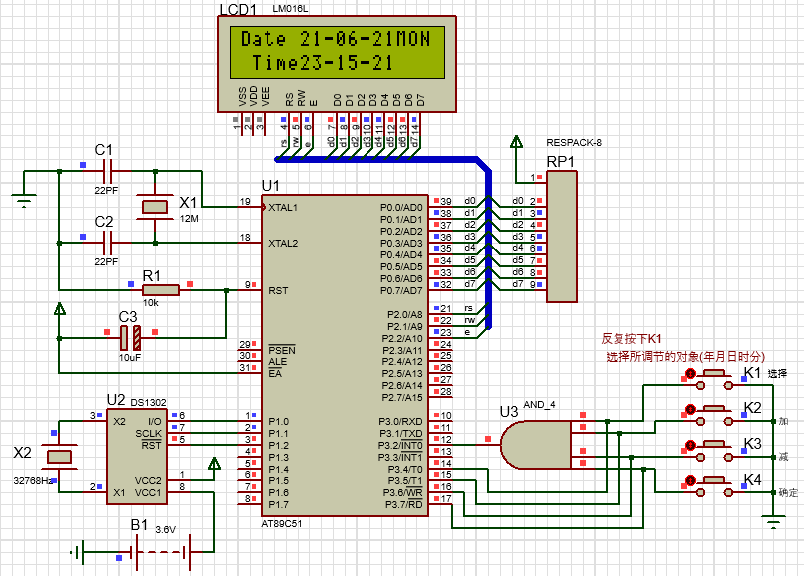
if(Adjust\_Index==-1) GetTime();

}

}

**附录2：设计图纸**





**附录3：元器件目录表**

表1.单片机最小系统元件清单

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 型 号（名称） | 数量 | 作用 | 标注 |
| 1 | STC89C52 | 1 | 单片机 | U1 |
| 2 | 40脚锁座 | 1 | 单片机座子 |  |
| 3 | 22pF瓷片电容 | 2 | 晶振电路用 | C2,C1 |
| 4 | 12MHZ晶振 | 1 | 晶振 | X1 |
| 5 | 10uF电解电容 | 1 | 上电复位 | C3 |
| 6 | 10K电阻 | 1 | 复位电路 | R1 |
| 7 | 4脚按键 | 1 | 按下，单片机实现复位功能 | K5 |

表2.DS1302时钟电路元件清单

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 型 号（名称） | 数量 | 作用 | 标注 |
| 1 | DS1302时钟芯片 | 1 | 时钟计时 | U2 |
| 2 | 32.768kHz晶振 | 1 | 晶振 | X2 |
| 3 | 3.6V电池模块 | 1 | 后备电源 | B1 |

表3.LCD1602液晶显示元件清单

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 型 号（名称） | 数量 | 作用 | 标注 |
| 1 | LCD1602 | 1 | 显示日历 | LM016L |
| 2 | 16脚单排母座 | 1 | 显示屏底座 |  |
| 3 | 9脚排阻 | 1 | 保护电路 | RP1 |

表4.独立按键元件清单

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 型 号（名称） | 数量 | 作用 | 标注 |
| 1 | 74LS21P | 1 | 4输入双与门 | U3 |
| 2 | 4脚按键 | 4 | 实现选择、加、减、确定功能 | K1、K2、K3、K4 |