本来是在准备重庆市TI 电子设计大赛，这块launchpad的板子也是自己买的。学习过程中越来越感觉得到launchpad的强大，一边学一边做东西，做到万年历的时候花了三天时间，终于调试成功，心里那个美啊。后来想到女朋友在培训不在身边，就有了制作这个万年历的想法。

一、首先说一下硬件：

1、 launchpad开发板一块（板载MSP430G2553）

2、 LCD1602液晶屏一块（3.3V）

3、 DS1302数字时钟芯片

4、 32.768KHz晶振一枚

5、 按键两枚

6、 电位器一枚

7、 电容、导线等

二、要实现的功能：

1、 精确显示年月日时分秒

2、 闰年、二月自动调节时间

3、 两个按键控制调节时间日期等

4、 显示自定义的语句

5、 恋爱纪念日提醒，显示不一样的画面（We’ve been together for XX years！）

三、硬件电路图





四、整体思路

//设置P1口作为输出连接LCD1602的数据端D0-D7，P2.0为rs,P2.1为rw，P2.2为e(都配置为输出)作为LCD1602的控制端口。

//配置P2.3为SCLK，P2.4为SDA，P2.5为RST(都配置为输出)作为DS1302的控制端

//P2.6、P2.7接两个按键，配置为输入,下降沿中断，中断处理函数 void key\_pro(),P2、6接key1(控制光标)，P2.7接key2，控制调节

//grace配置中设置P2.0---P2.5为输出，MCLK=1MHz，关闭看门狗

//在做DS1302的硬件时应该特别注意：数据线RST、SCLK、SDA一定上拉电阻，晶振添加负载电容，连接线尽量短

先列写出主函数，是思路清晰点：

**void** **main** (**void**)

{

**CSL\_init**(); // 初始化单片机

lsd1602\_init(); // 初始化LCD1602

ds1302\_init(); //初始化DS1302

**while**(1)

{

\_DINT(); //关闭中断

data\_pro(); //从DS1302中读取并处理数据

\_EINT();

**if**(month== 0x03 && date== 0x07)//3月7日是我们的纪念日

jww(); //如果是3月7号则显示不同的画面（jww是处理函数）

update\_disbuf(); //更新LCD1602显示空间（在2553中开辟）

display(); //控制LCD1602显示时间

**if**(month== 0x03 && date== 0x07)//3月7日纪念日

\_\_delay\_cycles(1000000);

}

}

就这样很简单的思路，全部使用了子程序调用，所以看起来清晰点，下面贴出来整个函数。注释的挺多了，有什么建议或者交流的直接回复帖子。

五、程序

说一下阅读方法，先看懂了上边的主函数，然后对应主函数调用的子函数，一个一个来看，逐个看懂，就很简单。（有一个难点，就是按键处理。因为只用了两个按键，key1每按一次会有全局变量i增加一，控制LCD1602光标的显示位置，另外一个按键调节时间）

//设置P0作为输出连接LCD1602的数据端D0-D7，P2.3为SCLK，P2.4为SDA，P2.5为RST(都配置为输出)

//P2.0为rs,P2.1为rw，P2.2为e(都配置为输出)

//P2.67配置为输入,下降沿中断，中断处理函数 void key\_pro(),P2。6接key1(控制光标)，P2.7接key2，控制调节（增加）

//grace配置中设置P2.345为输出，MCLK=1MHz，

//本程序将数据用ASCII表示，方便LCD1602显示

//在做DS1302的硬件时应该特别注意：数据线RST。SCLK、SDA一定上拉电阻，晶振添加负载电容，连接线尽量短

//date代表日，day代表星期几

**#include** <msp430.h>

**#include** <ti/mcu/msp430/csl/CSL.h>

//定义全局变量年月日时分秒

**char** year,month,date,hour,minute,second,n;//n代表周几

**char** love\_years; //恋爱纪念年

**char** day[7][3]={"MON","TUE","WED","THI","FRI","SAT","SUN"};

**char** i=0; //用来记录key1按下的次数

**char** dis\_buf1[16]; //lcd上排显示缓冲区

**char** dis\_buf2[16]; //lcd下排显示缓冲区

**char** weiwei1[16]={"weiwei: I love u"};

**char** weiwei2[16]={"so much! by xuxu"};

**char** love1[16]={"We've been toget"};

**char** love2[16]={"her for years"};

//LCD1602控制位

**#define** rs\_0 (P2OUT &= ~BIT0)

**#define** rs\_1 (P2OUT |= BIT0)

**#define** rw\_0 (P2OUT &= ~BIT1)

**#define** rw\_1 (P2OUT |= BIT1)

**#define** e\_0 (P2OUT &= ~BIT2)

**#define** e\_1 (P2OUT |= BIT2)

//DS1302控制位

**#define** SCLK\_0 (P2OUT &= ~BIT3)

**#define** SCLK\_1 (P2OUT |= BIT3)

**#define** SDA\_0 (P2OUT &= ~BIT4)

**#define** SDA\_1 (P2OUT |= BIT4)

**#define** RST\_0 (P2OUT &= ~BIT5)

**#define** RST\_1 (P2OUT |= BIT5)

**#define** SDA (P2IN &= BIT4)

//宏定义DS1302数据地址

**#define** write\_second 0x80

**#define** read\_second 0x81

**#define** write\_minute 0x82

**#define** read\_minute 0x83

**#define** write\_hour 0x84

**#define** read\_hour 0x85

**#define** write\_date 0x86

**#define** read\_date 0x87

**#define** write\_month 0x88

**#define** read\_month 0x89

**#define** write\_year 0x8c

**#define** read\_year 0x8d

**#define** write\_day 0x8a

**#define** read\_day 0x8b

**#define** write\_protect 0x8e

**#define** write\_power 0x90

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*以下为LCD1602的子函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//查询是否忙碌（每次输入指令前都要判断液晶模块是否处于忙的状态）

**char** **busy**(**void**)

{

**char** busyc;

rs\_0;

rw\_1;

e\_1;

\_\_delay\_cycles(1000);

P1DIR = 0x00; //P1口置为输入，读取LCD状态

busyc = P1IN & BIT7; //从P1输入是否忙碌（D7高电平表示忙碌）

P1DIR = 0xff;

e\_0;

**return** busyc;

}

//向LCD1602写字节命令

**void** **write\_command**(**char** command)

{

**while**(busy()); //判断LCD是否忙碌

rs\_0;

rw\_0;

P1OUT = command;

\_\_delay\_cycles(1); //时间间隔MIN=30nS

e\_1;

\_\_delay\_cycles(1); //MIN=150nS

e\_0;

}

//向LCD1602写数据命令

**void** **write\_data**(**char** data)

{

**while**(busy()); //判断LCD是否忙碌

rs\_1;

rw\_0;

\_\_delay\_cycles(1);

e\_1;

P1OUT=data;

\_\_delay\_cycles(1);

e\_0;

}

//LCD1602初始化函数

**void** **lsd1602\_init**()

{

e\_0;

write\_command(0x38);

\_\_delay\_cycles(15000);

write\_command(0x0c);

write\_command(0x06);

write\_command(0x01);

}

**void** **update\_disbuf**()

{

dis\_buf1[0]='2';

dis\_buf1[1]='0';

dis\_buf1[2]=year/16 + 0x30;

dis\_buf1[3]=year%16 + 0x30;

dis\_buf1[4]='-';

dis\_buf1[5]=month/16 + 0x30;

dis\_buf1[6]=month%16 + 0x30;

dis\_buf1[7]='-';

dis\_buf1[8]=date/16 + 0x30;

dis\_buf1[9]=date%16 + 0x30;

dis\_buf1[10]=' ';

dis\_buf1[11]=' ';

dis\_buf1[12]=' ';

dis\_buf1[13]=day[n-1][0];

dis\_buf1[14]=day[n-1][1];

dis\_buf1[15]=day[n-1][2];

dis\_buf2[0]=' ';

dis\_buf2[1]='j';

dis\_buf2[2]='w';

dis\_buf2[3]='w';

dis\_buf2[4]='&';

dis\_buf2[5]='z';

dis\_buf2[6]='x';

dis\_buf2[7]=' ';

dis\_buf2[8]=hour/16 + 0x30;

dis\_buf2[9]=hour%16 + 0x30;

dis\_buf2[10]=':';

dis\_buf2[11]=minute/16 + 0x30;

dis\_buf2[12]=minute%16 + 0x30;

dis\_buf2[13]=':';

dis\_buf2[14]=second/16 + 0x30;

dis\_buf2[15]=second%16 + 0x30;

}

**void** **display**()

{

**char** t;

write\_command(0x80);

**for** (t=0;t<=15;t++)

{write\_data(dis\_buf1[t]);}

write\_command(0xc0);

**for** (t=0;t<=15;t++)

{write\_data(dis\_buf2[t]);}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*以下为DS1302的子函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//向DS1302写一个字节的数据

**void** **ds1302\_Wbyte**(**char** dat)

{

**char** s,temp;

temp = dat;

**for**(s=0;s<8;s++)

{**if**(temp & 0x01) //从低位开始传输数据

{SDA\_1;}

**else**

SDA\_0;

SCLK\_1; //上升沿锁存数据

temp = temp>>1; //数据右移一位为下次输出做准备

SCLK\_0; //为下一次做准备

}

}

//从DS1302读取8位数据

**char** **ds1302\_Rbyte**()

{

**char** u,temp;

P2DIR &= ~BIT4;

**for**(u=8; u>0;u--)

{

temp=temp>>1;

temp += (P2IN&BIT4)<<3;

SCLK\_1;

SCLK\_0;

}

P2DIR |= BIT4;

**return**(temp);

}

//向DS1302写入地址然后读取数据

**char** **ds1302\_Read**(**char** cmd)

{

**char** dat;

RST\_0; //初始化RST为低

SCLK\_0; //SLK=0

RST\_1; //开始传输数据

ds1302\_Wbyte(cmd); //传输命令字，写入要读取的时间、日期地址

dat = ds1302\_Rbyte(); //读取八位数据

RST\_0; //结束数据传输

SCLK\_0; //拉高时钟线

**return**(dat);

}

//向DS1302写入地址后写入数据

**void** **ds1302\_Write**(**char** cmd,**char** dat)

{

RST\_0;

SCLK\_0; //在RST的上升沿，SCLK必须为0

RST\_1; //开始传输数据

ds1302\_Wbyte(cmd); //写入要修改的地址

ds1302\_Wbyte(dat); //写入数据

SCLK\_1; //拉高时钟线

RST\_0; //拉低RST

}

//初始化DS1302的程序

**void** **ds1302\_init**(**void**)

{

ds1302\_Write(write\_protect,0x00);//关闭写保护

ds1302\_Write(write\_second,0x00); //初始化秒为00，不暂停时钟(BIT7)

ds1302\_Write(write\_minute,0x59); //初始化分为00

ds1302\_Write(write\_hour,0x23); //初始化时为23，设置为24小时制(BIT7)

ds1302\_Write(write\_date,0x01); //初始化日期为01

ds1302\_Write(write\_month,0x01); //初始化月为01

ds1302\_Write(write\_year,0x12); //初始化年为(20)12

ds1302\_Write(write\_day,0x07); //周日

ds1302\_Write(write\_power,0x0a5); //一个二极管压降，电阻4K

ds1302\_Write(write\_protect,0x80);//打开写保护

}

//DS1302数据处理函数，此函数将个位、十位分别用char表示，方便用数码管显示

**void** **data\_pro**(**void**)

{

ds1302\_Write(write\_protect,0x00);//关闭写保护

date = ds1302\_Read(read\_date); //读取日

**if**(date>0x29 && month==0x02) //如果调时时2月超过29

{ds1302\_Write(write\_date,0x01);

date = ds1302\_Read(read\_date);

ds1302\_Write(write\_month,month + 1);

}

month = ds1302\_Read(read\_month);//读取月

year = ds1302\_Read(read\_year); //读取年

hour = ds1302\_Read(read\_hour); //读取时

hour = hour & 0x3f; //屏蔽掉前两位

minute = ds1302\_Read(read\_minute);//读取分

second = ds1302\_Read(read\_second);//读取秒

n = ds1302\_Read(read\_day);//读取星期

ds1302\_Write(write\_protect,0x80);//打开写保护

}

//按键中断处理程序(使用grace配置)

**void** **key\_pro**(**void**)

{

**char** j; //用于复制自定义的显示计数（16个）

\_\_delay\_cycles(8000); //延时去抖动

**if**(!(P2IN & BIT6)) //判断是否有按键按下

{

i++;

**switch**(i)

{

**case** 1:

write\_command(0x83); //显示光标到年

write\_command(0x0f);

**break**;

**case** 2:

write\_command(0x86); //显示光标到月

write\_command(0x0f);

**break**;

**case** 3:

write\_command(0x89); //显示光标到日

write\_command(0x0f);

**break**;

**case** 4:

write\_command(0x8f); //显示光标到周

write\_command(0x0f);

**break**;

**case** 5:

write\_command(0xc9); //显示光标到时

write\_command(0x0f);

**break**;

**case** 6:

write\_command(0xcc); //显示光标到分

write\_command(0x0f);

**break**;

**case** 7:

write\_command(0xcf); //显示光标到秒

write\_command(0x0f);

**break**;

**default**: **break**;

}

}

**if**(i>7)

{

i=0;

write\_command(0x0c); //关闭光标显示

}

P2IFG=0; //软件复位标志

**while**(!(P2IN & BIT6))

{

ds1302\_Write(write\_protect,0x00);//关闭写保护

**while**(!(P2IN & BIT7)) //如果按键2被按下

{

**switch**(i) //根据i的值来确定某变量进行加1

{**case** 0: //显示自定义的语句

**for**(j=0;j<16;j++)

{

dis\_buf1[j] = weiwei1[j];

dis\_buf2[j] = weiwei2[j];

display();

}

**break**;

**case** 1: //此事处理全局变量year

year++; //year加1

**if**(year>0x99) //如果year大于99则置0

year=0;

**if**((year & 0x0f)>0x09) //如果year低四位大于9（第四位BCD码表示个位），十位加1，个位置0

{year +=0x10;

year &=0xf0;

}

ds1302\_Write(write\_year,year);//将变化后的year写入DS1302

year = ds1302\_Read(read\_year);//读出来year显示（使调节year时同步变化，为了人性化）

update\_disbuf(); //更新显示字符

**while**(!(P2IN & BIT7))

display();

**break**;

**case** 2: //此时处理全局变量month

month++;

**if**(month>0x12)

month=1;

**if**((month & 0x0f)>0x09)

{month +=0x10;

month &=0xf0;

}

ds1302\_Write(write\_month,month);

month = ds1302\_Read(read\_month);

update\_disbuf(); //更新显示字符

**while**(!(P2IN & BIT7))

display();

**break**;

**case** 3: //此时处理全局变量date

date++;

**if**( (date>0x29 && month==0x02)

|| (month==0x04 || month==0x06 || month==0x09 || month==0x11)&& date>0x30

|| date>0x31 )

date=0x01;

**if**((date & 0x0f)>0x09)

{date +=0x10;

date &=0xf0;

}

ds1302\_Write(write\_date,date);

date = ds1302\_Read(read\_date);

update\_disbuf(); //更新显示字符

**while**(!(P2IN & BIT7))

display();

**break**;

**case** 4: //此时处理全局变量day

n++;

**if**(n>0x07)

n=1;

ds1302\_Write(write\_day,n);

n = ds1302\_Read(read\_day);

update\_disbuf(); //更新显示字符

**while**(!(P2IN & BIT7))

display();

**break**;

**case** 5: //此时处理全局变量hour

hour++;

**if**(hour>0x23)

hour=0;

**if**((hour & 0x0f)>0x09)

{hour +=0x10;

hour &=0xf0;

}

ds1302\_Write(write\_hour,hour);

hour = ds1302\_Read(read\_hour);

update\_disbuf(); //更新显示字符

**while**(!(P2IN & BIT7))

display();

**break**;

**case** 6: //此时处理全局变量minute

minute++;

**if**(minute>0x59)

minute=0;

**if**((minute & 0x0f)>0x09)

{minute +=0x10;

minute &=0xf0;

}

ds1302\_Write(write\_minute,minute);

minute = ds1302\_Read(read\_minute);

update\_disbuf(); //更新显示字符

**while**(!(P2IN & BIT7))

display();

**break**;

**case** 7: //此时处理全局变量second

second++;

**if**(second>0x59)

second=0;

**if**((second & 0x0f)>0x09)

{second +=0x10;

second &=0xf0;

}

ds1302\_Write(write\_second,second);

second = ds1302\_Read(read\_second);

update\_disbuf(); //更新显示字符

**while**(!(P2IN & BIT7))

display();

**break**;

**default**: **break**;

}

}

ds1302\_Write(write\_protect,0x80);//打开写保护

}

}

//纪念日处理程序

**void** **jww**(**void**)

{**char** ww\_xx;

**for**(ww\_xx=0;ww\_xx<16;ww\_xx++)

{

dis\_buf1[ww\_xx] = love1[ww\_xx];

dis\_buf2[ww\_xx] = love2[ww\_xx];

}

love\_years= year - 0x11; //减去2011年

dis\_buf2[8]=love\_years/16 + 0x30;

dis\_buf2[9]=love\_years%16 + 0x30;

display();

\_DINT();

\_\_delay\_cycles(2000000); //延时显示出来

\_EINT();

}

//主函数

**void** **main** (**void**)

{

**CSL\_init**(); // 初始化单片机

lsd1602\_init(); // 初始化LCD1602

ds1302\_init(); //初始化DS1302

**while**(1)

{

\_DINT();

data\_pro(); //读取并处理数据

\_EINT();

**if**(month== 0x03 && date== 0x07)//3月7日纪念日

jww();

update\_disbuf(); //更新显示字符

display(); //显示时间

**if**(month== 0x03 && date== 0x07)//3月7日纪念日

\_\_delay\_cycles(1000000);

}

}