实验四 三线制实时时钟的读写

**实验目的：**

1. 掌握IIC总线的使用方式
2. 掌握实时时钟芯片DS1302的工作原理
3. 掌握三线制方式实现时钟芯片DS1302的数据读写

**实验内容：**

学习三线制传输机制，通过单片机MCU的I/O实现三线制方式读取DS1302S时钟数据并分别将年、月、日、时间信息分别显示在数码管上。

参考资料：芯片手册文档，

**实验步骤：**

1. 编写DS1302读写程序，和数码管显示程序。
2. 编写按键程序，实现不同按键显示年、月、日、时间等信息。

**实验要求：**

编写实验报告，主要包括关键步骤的实现和效果截屏，并分析实验过程中出现的问题和分析解决方法。

1. **代码实现**

**Main文件**

#include "reg51.h"

#include "LCD1602.h"

#include "DS1302.h"

#define uchar unsigned char

#define uint unsigned int

bit key\_flag1=0, key\_flag2=0;

SYSTEMTIME adjusted;

#define KEY\_MATRIX\_PORT P1 *//使用宏定义矩阵按键控制口*

uchar sec\_add=0, min\_add=0, hou\_add=0, day\_add=0, mon\_add=0, yea\_add=0;

uchar data\_alarm[7]={0};

uchar test[6] = " C";

int key\_scan()

{

int i=0;

uint temp;

P1=0xF0;

temp=P1;

**if**(temp!=0xF0)

{

i=1;

test[5]='C';

}

**else**

{

i=0;

}

**return** i;

}

uchar key\_value()

{

uint m=0, n=0, temp;

uchar value;

uchar v[3][4]={'0','1','2','6','3','4','5','7','8','9','A','B'};

P1=0xfe;temp=P1; **if**(temp!=0xfe) m=3;

P1=0xfd;temp=P1; **if**(temp!=0xfd) m=2;

P1=0xfb;temp=P1; **if**(temp!=0xfb) m=1;

P1=0xf7;temp=P1; **if**(temp!=0xf7) m=0;

P1=0x7f;temp=P1; **if**(temp!=0x7f) n=0;

P1=0xbf;temp=P1; **if**(temp!=0xbf) n=1;

P1=0xdf;temp=P1; **if**(temp!=0xdf) n=2;

value=v[n][m];

**return** value;

}

void adjust(void)

{

**if**(key\_scan()&&key\_flag1)

{

**switch**(key\_value())

{

**case** '5':adjusted.Second++;test[5]='S';**break**;

**case** '4':adjusted.Minute++;test[5]='M';**break**;

**case** '3':adjusted.Hour++;test[5]='H';**break**;

**case** '2':adjusted.Day++;test[5]='D';**break**;

**case** '1':adjusted.Month++;test[5]='M';**break**;

**case** '0':adjusted.Year++;test[5]='Y';**break**;

*//case '7':Save\_Time(&adjusted);break;*

default:test[5]='C'; **break**;

}

**if**(adjusted.Second>59)

{

adjusted.Second=adjusted.Second%60;

adjusted.Minute++;

}

**if**(adjusted.Minute>59)

{

adjusted.Minute=adjusted.Minute%60;

adjusted.Hour++;

}

**if**(adjusted.Hour>23)

{

adjusted.Hour=adjusted.Hour%24;

adjusted.Day++;

}

**if**(adjusted.Day>31)

{

adjusted.Day=adjusted.Day%31;

adjusted.Month++;

}

**if**(adjusted.Month>12)

{

adjusted.Month=adjusted.Month%12;

adjusted.Year++;

*//test[5]='O';*

}

**if**(adjusted.Year>100)

{

adjusted.Year=adjusted.Year%100;

}

Save\_Time(&adjusted);

}

}

void changing(void) interrupt 0 **using** 0

{

**if**(key\_flag1)

{

key\_flag1=0;

}

**else**

{

key\_flag1=1;

}

}

void main()

{

uint i;

uchar p1[]="D:", p2[]="T:";

SYSTEMTIME T;

EA=1;

EX0=1;

IT0=1;

EA=1;

EX1=1;

IT1=1;

init1602();

Initial\_DS1302();

**while**(1)

{

write\_com(0x80);

write\_string(p1, 2);

write\_com(0xc0);

write\_string(p2, 2);

**if**(key\_flag1==0)

{

DS1302\_GetTime(&T);

adjusted.Second=T.Second;

adjusted.Minute=T.Minute;

adjusted.Hour=T.Hour;

adjusted.Day=T.Day;

adjusted.Month=T.Month;

adjusted.Year=T.Year;

**for**(i=0; i<9; i++)

{

adjusted.DateString[i] = T.DateString[i];

adjusted.TimeString[i] = T.TimeString[i];

}

}

**else**

adjust();

DateToStr(&adjusted);

TimeToStr(&adjusted);

write\_com(0x82);

write\_string(adjusted.DateString, 8);

**if**(key\_flag1)write\_string(test, 6);

**else** write\_string(" ", 6);

write\_com(0xc2);

write\_string(adjusted.TimeString, 8);

**if**(key\_flag1)write\_string(test, 6);

**else** write\_string(" ", 6);

delay(10);

}

}

**DS1302.h**

#ifndef TIMER\_DS1302

#define TIMER\_DS1302

sbit DS1302\_CLK = P3^6;*//实时时钟时钟线引脚*

sbit DS1302\_IO = P3^4;*//实时时钟数据线引脚*

sbit DS1302\_RST = P3^5;*//实时时钟复位线引脚*

sbit ACC0 = ACC^0;*//定义ACC的最低位和最高位,在对ACC移位操作后,用于传输数据*

sbit ACC7 = ACC^7;

**typedef** **struct** SYSTEM\_TIME

{

unsigned char Second;

unsigned char Minute;

unsigned char Hour;

unsigned char Week;

unsigned char Day;

unsigned char Month;

unsigned char Year;

unsigned char DateString [9];*//用这两个字符串来放置读取的时间*

unsigned char TimeString[9];

}SYSTEMTIME; *//定义的时间类型结构体*

#define AM(X) X

#define PM(X) (X+12) *//转成24小时制*

#define DS1302\_SECOND 0x80 *//片内各位数据的地址*

#define DS1302\_MINUTE 0x82

#define DS1302\_HOUR 0x84

#define DS1302\_WEEK 0x8A

#define DS1302\_DAY 0x86

#define DS1302\_MONTH 0x88

#define DS1302\_YEAR 0x8C

#define DS1302\_RAM(X)(0xC0+(X)\*2)

*//用于计算 DS1302\_RAM 地址的宏*

*/\*\*\*\*\*\*内部指令\*\*\*★★\*\*\*★\*吉\*★\*/*

void DS1302InputByte(unsigned char d) *//实时时钟写入1B(内部函数)*

{

unsigned char i;

ACC=d;

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

{

DS1302\_IO=ACC0;

DS1302\_CLK=1; *//写数据在上升沿,且先写低位再写高位*

DS1302\_CLK=0; *//因为在前面已定义ACCO=ACC^0;以便再次利用*

ACC=ACC>>1;

}

}

*//函数功能:实时时钟读取1B(内部函数)*

unsigned char DS1302OutputByte (void)

{

unsigned char i;

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

{

ACC=ACC >>1;

*//相当于汇编中的 RRC*

ACC7=DS1302\_IO;

*//由低位到高位传播AcC7中的信息*

DS1302\_CLK=1;

*//读信息是在下降沿*

DS1302\_CLK=0;

}

**return**(ACC);

}

void Write1302(unsigned char ucAddr, unsigned char ucDa) *//ucAddr: DS1302地址,*

{

*//ucData:要写的数据*

DS1302\_RST =0;

DS1302\_CLK =0;

DS1302\_RST =1;

DS1302InputByte(ucAddr) ;

*//地址,命令*

DS1302InputByte(ucDa);

*//写1B数据*

DS1302\_CLK =1;

DS1302\_RST =0;

}

unsigned char Read1302(unsigned char ucAddr) *//读取DS1302某地址的数据*

{

unsigned char ucData;

DS1302\_RST=0;

DS1302\_CLK = 0;

DS1302\_RST =1;

DS1302InputByte(ucAddr|0x01);

*//上升沿,写地址,命令*

ucData = DS1302OutputByte();

*//下降沿,读1B数据*

DS1302\_CLK =1;

DS1302\_RST =0;

**return**(ucData) ;

*//在上升沿之后进行写操作,在下降沿之前进行读操作*

}

void DS1302\_SetProtect(bit flag) *//!是否写保护*

{

**if**(flag)

Write1302(0X8E,0x80);

**else**

Write1302(0X8E,0x00);

}

void DS1302\_SetTime (unsigned char Address,unsigned char Value)*//函数功能:设置时间*

{

DS1302\_SetProtect(0);

Write1302(Address, ((Value/10)<<4 | (Value%10)));*//将十进制数转换为BCD码*

}

*//在 DS1302中的与日历、时钟相关的寄存器存放的数据必须为BCD码形式*

void DS1302\_GetTime (SYSTEMTIME \*Time)

{

unsigned char ReadValue;

ReadValue = Read1302(DS1302\_SECOND);

Time->Second = ((ReadValue&0x70)>>4)\*10 + (ReadValue&0x0F);*//将BCD码转换为十进制数,此处为结构体操作*

ReadValue = Read1302(DS1302\_MINUTE);

Time->Minute =((ReadValue&0x70)>>4)\*10 +(ReadValue&0x0F);

ReadValue = Read1302(DS1302\_HOUR);

Time->Hour =((ReadValue&0x70)>>4)\*10+(ReadValue&0x0F);

ReadValue = Read1302(DS1302\_DAY);

Time->Day =((ReadValue&0x70)>>4)\*10 +(ReadValue&0x0F);

ReadValue = Read1302(DS1302\_WEEK);

Time->Week =((ReadValue&0x70)>>4)\*10 + (ReadValue&0x0F);

ReadValue = Read1302(DS1302\_MONTH);

Time->Month = ((ReadValue&0x70)>>4)\*10 +(ReadValue&0x0F);

ReadValue = Read1302(DS1302\_YEAR);

Time->Year = ((ReadValue&0x70)>>4)\*10 +(ReadValue&0x0F);

}

void DateToStr (SYSTEMTIME\* Time)

{

*//将十进制数转换为液晶显示的ASCII 值，即变为字符型，此函数为年月日信息*

Time->DateString[0]=Time->Year/10 +'0';

Time->DateString[1]=Time->Year%10 +'0';

Time->DateString[2]='-';

Time->DateString[3]=Time->Month/10 +'0';

Time->DateString[4]=Time->Month%10 +'0';

Time->DateString[5]='-';

Time->DateString[6]=Time->Day/10 +'0';

Time->DateString[7]=Time->Day%10 +'0';

Time->DateString[8]='\0';

}

void TimeToStr(SYSTEMTIME\* Time)

{

*//将十进制数转换为液晶显示的ASCII值，此处为时间信息*

Time->TimeString[0]=Time->Hour/10 +'0';

Time->TimeString[1]=Time->Hour%10 +'0';

Time->TimeString[2]=':';

Time->TimeString[3]=Time->Minute/10 + '0';

Time->TimeString[4]=Time->Minute%10+'0';

Time->TimeString[5]=':';

Time->TimeString[6]=Time->Second/10+ '0';

Time->TimeString[7]=Time->Second%10 +'0';

Time->DateString[8]='\0';

}

void Initial\_DS1302 (void)

{

unsigned char Second;

Second=Read1302(DS1302\_SECOND);

**if**(Second&0x80)*//初始化时间(*

DS1302\_SetTime(DS1302\_SECOND,0);

DS1302\_SetTime(DS1302\_MINUTE,0);

DS1302\_SetTime(DS1302\_YEAR,21);

DS1302\_SetTime(DS1302\_MONTH,12);

DS1302\_SetTime(DS1302\_DAY,17);

}

**LCD1602.h**

#ifndef LCD\_CHAR\_1602\_2005\_4\_9

#define LCD\_CHAR\_1602\_2005\_4\_9

#define uchar unsigned char

#define uint unsigned int

sbit lcdrs = P2^6;

sbit lcdrw = P2^5;

sbit lcden = P2^7;

void delay (uint z)

{

uint x,y;

**for**(x=z;x>0 ; x--)

**for**(y=110; y>0;y--);

}

void write\_com(uchar com)

{

lcdrw=0;

lcdrs=0;

P0=com;

delay(5);

lcden=1;

delay(5);

lcden=0;

}

void write\_data(uchar date)

{

lcdrw=0;

lcdrs=1;

P0=date;

delay(5);

lcden=1;

delay(5);

lcden=0;

}

void init1602()

{

lcdrw=0;

lcden=0;

write\_com(0x3C);

write\_com(0x0C);

write\_com(0x06);

write\_com(0x01);

write\_com(0x80);

}

void write\_string(uchar \*pp, uint n)

{

int i;

**for**(i=0; i<n; i++)

{

write\_data(pp[i]);

}

}

#endif

1. **实现效果**

**显示日期和时间**

一些电子设备

描述已自动生成

**修改月份**

电子器材

中度可信度描述已自动生成

电子仪器

中度可信度描述已自动生成