#include <avr/io.h>

#include <avr/delay.h>

#include "DS1302.h"

#include "spi.h"

//读写操作位

#define CMD\_READ 0x01

#define CMD\_WRITE 0x00

//时钟 命令

#define CMD\_SECOND 0x80 //秒

#define CMD\_MINUTE 0x82 //分

#define CMD\_HOUR 0x84 //时

#define CMD\_DAY 0x86 //日

#define CMD\_MONTH 0x88 //月

#define CMD\_WEEK 0x8A //星期 DATE

#define CMD\_YEAR 0x8C //年

#define CMD\_CONTROL 0x8E //控制(写保护)

#define CMD\_TRICKLE\_CHARGE 0x90 //涓流充电

#define CMD\_CLOCK\_BURST 0xBE //连续读取

//时钟配置常量

#define CFG\_CLOCK\_HALT 0x80 //停止时钟控制位 SECOND bit7

#define CFG\_12\_24 0x80 //12/24小时值选择位 HOUR bit7

#define CFG\_AM\_PM 0x20 //AP/PM位 HOUR bit5

#define CFG\_PROTECT 0x80 //写保护控制位 CONTROL bit7

#define CFG\_UNPROTECT 0x00 //写保护控制位 CONTROL bit7

//涓流充电控制常量

#define CFG\_TC\_D1R2 0xA5 //high 1 Diode +2K Resistors

#define CFG\_TC\_D2R8 0xAB //low 2 Diodes+8K Resistors

#define CFG\_TC\_DISABLED 0x00 //Disabled(TCS<>1010 or DS=00 or RS=00)

//RAM 命令

#define CMD\_RAM\_BASE 0xC0 //RAM0~RAM30<<1 地址需左移一位

#define CMD\_RAM\_BURST 0xFE //连续读取

//M16管脚定义

#define DS1302\_CE 1 //PA1 必须设为输出 ----DS1302 pin5

//宏定义

#define EnDS1302 PORTA|= (1<<DS1302\_CE) // 置CE为高电平，使能SPI接口

#define DiDS1302 PORTA&=~(1<<DS1302\_CE) // 置CE为低电平，关闭SPI接口

unsigned char sWEEK[]="Wr 1 2 3 4 5 6Su";

unsigned char CURDATE[7];

/\*-----------------------------------------------------------------------

Single\_SPI : 使用SPI接口读写数据到DS1302

输入参数： command: 命令选择;

wdata: 写入的数据;

返回值 rdata: 读回的数据;

(SPI的收发是同时进行的)

-----------------------------------------------------------------------\*/

unsigned char Single\_SPI(unsigned char command,unsigned char wdata)

{

unsigned char rdata;

EnDS1302; // 使能LCD

\_delay\_us(10);

spi\_rw(command); // 传送命令到SPI寄存器

\_delay\_us(10);

rdata = spi\_rw(wdata); // 传送数据到SPI寄存器

DiDS1302; // 关断LCD SPI通讯

\_delay\_us(10);

return rdata;

}

/\*-----------------------------------------------------------------------

Burst\_SPI : 使用SPI接口读写一批数据到DS1302

输入参数： CMD: 选择Clock/RAM，选择读写

lenth: 数据长度

pwdata: 写入数据的缓冲区;

prdata: 读回数据的缓冲区;

(SPI的收发是同时进行的)

读CLOCK的数据从SECOND寄存器开始，到CONTROL寄存器，最大8字节

写CLOCK的数据必须一次写完8个字节

读写RAM的数据从 RAM0地址开始，到RAM30地址,最大31字节

-----------------------------------------------------------------------\*/

void Burst\_SPI(unsigned char CMD,unsigned char len,

unsigned char \*pwdata,unsigned char \*prdata)

{

EnDS1302; // 使能SPI通讯

\_delay\_us(10);

spi\_rw(CMD); // 传送命令到SPI寄存器

\_delay\_us(10);

while (len--)

{

\*prdata++=spi\_rw(\*pwdata++); // 传送数据到SPI寄存器

};

DiDS1302; // 关断SPI通讯

\_delay\_us(10);

}

//检测DS1302

unsigned char check\_RTC(void)

{

unsigned char i;

Single\_SPI(CMD\_CONTROL|CMD\_WRITE,CFG\_UNPROTECT);//写允许

Single\_SPI(CMD\_RAM\_BASE|CMD\_WRITE,0xA5); //写RAM0

Single\_SPI(CMD\_CONTROL|CMD\_WRITE,CFG\_PROTECT); //写保护

i=Single\_SPI(CMD\_RAM\_BASE|CMD\_READ,0); //读取RAM0

if(i==0xA5) return 1;

else return 0;

}

//初始化DS1302

void init\_RTC(char \*init\_data)

{

unsigned char temp[8];

DiDS1302;

DDRA|= (1<<DS1302\_CE);

Single\_SPI(CMD\_CONTROL|CMD\_WRITE,CFG\_UNPROTECT); //写允许

Burst\_SPI(CMD\_CLOCK\_BURST|CMD\_WRITE,8,init\_data,&temp[0]); //初始化时间

Single\_SPI(CMD\_TRICKLE\_CHARGE,CFG\_TC\_D1R2); //涓流充电选择

Single\_SPI(CMD\_CONTROL|CMD\_WRITE,CFG\_PROTECT); //写保护

}

//压缩BCD码转换成ASCII码

void BCDtoASCII(unsigned char cBCD,unsigned char \*pstr)

{

unsigned char i,j;

i=cBCD>>4;

j=cBCD&0x0F;

\*pstr++=i+0x30;

\*pstr=j+0x30;

}

//BCD码0~7转换成汉字'一'~'日'

void decodeWEEK(unsigned char WEEK,unsigned char \*pstr)

{

unsigned char i;

i=WEEK;

i&=0x07;

i=i<<1;

\*pstr++=sWEEK[i++];

\*pstr=sWEEK[i];

}

//读取当前时间并解释

void read\_date(char \*s\_cur\_date)

{

Burst\_SPI(CMD\_CLOCK\_BURST|CMD\_READ,7,0,CURDATE); //读取当前时间

BCDtoASCII(CURDATE[6],s\_cur\_date + 2); //年 ，00~10

BCDtoASCII(CURDATE[4],s\_cur\_date + 6); //月 ，01~12

BCDtoASCII(CURDATE[3],s\_cur\_date + 10); //日 ，01~31(28/30/31)

BCDtoASCII(CURDATE[2],s\_cur\_date + 14); //时 ，00~23

BCDtoASCII(CURDATE[1],s\_cur\_date + 18); //分 ，00~59

BCDtoASCII(CURDATE[0],s\_cur\_date + 22); //秒 ，00~59

decodeWEEK(CURDATE[5],s\_cur\_date + 31); //星期 ，1~7 一~日

}

/\*-----------------------------------------------------------------------

delay\_nms :长延时函数(GCC内部延时函数的延时过短)

输入参数: t :延时时间 ms

-----------------------------------------------------------------------\*/

void delay\_nms(unsigned int t)

{

while (t--)

\_delay\_ms(1);

}

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

\*\*作品：吴鉴鹰单片机驱动12864大字万年历设计和温度显示

\*\*申明：转载请标明作品来源 知识产权归作者本人所有！

\*\* 这一讲的主要内容: LCD12664液晶显示原理

\*\* 功能描述:单片机驱动12864大字万年历设计和温度显示

\*\* 输　入:

\*\* 液晶：Proteus元件库—AMPIRE128X64(无字库)

\*\* 输　出:

\*\*

\*\* 全局变量:

\*\* 调用模块:

\*\*

\*\* 作　者: 吴鉴鹰

\*\* 日　期: 14.07.2

\*\*作者所属单位：鉴鹰电子工作室

\*\*交流QQ群: 第一群： 332164870 第二群：194314772 联系QQ:1123942529

\*\*申明：转载请标明作品来源 知识产权归作者吴鉴鹰所有！

备注：有什么错误的地方，欢迎各大读者指正

O(∩\_∩)O~

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

#include <at89x51.h>

#include "1302.h"

/\*#define uchar unsigned char

#define uint unsigned int //宏定义

#define Ds1302\_Second\_Addr 0x80

#define Ds1302\_Minute\_Addr 0x82

#define Ds1302\_Hour\_Addr 0x84

#define Ds1302\_Day\_Addr 0x86

#define Ds1302\_Week\_Addr 0x8a

#define Ds1302\_Month\_Addr 0x88

#define Ds1302\_Year\_Addr 0x8c \*/

/\*\*端口定义\*\*/

sbit IO=P3^7;

sbit CLK=P3^6;

sbit RST=P3^5;

sbit ACC0=ACC^0;

sbit ACC7=ACC^7;

char datastring[13]; ////存放时间的数组

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

\*\* 函数名称: void Ds1302\_Write\_Fu(uchar dat)

\*\* 功能描述: 向ds1302中写入数据的函数

\*\* 输　入: uchar dat（向ds1302写入的数）

\*\*

\*\*

\*\* 输　出:

\*\*

\*\* 全局变量:

\*\* 调用模块:

\*\*

\*\* 作　者: 吴鉴鹰

\*\* 日　期: 14.07.02

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

void Ds1302\_Write\_Fu(uchar dat)

{

uchar i,j;

j=dat;

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

{

IO=j&0x01;

CLK=1;

CLK=0;

j>>=1;

}

}

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

\*\* 函数名称: uchar Ds1302\_Read\_Fu()

\*\* 功能描述: 从实时时钟读出一个字节

\*\* 输　入:

\*\*

\*\*

\*\* 输　出:

\*\*

\*\* 全局变量:

\*\* 调用模块:

\*\*

\*\* 作　者: 吴鉴鹰

\*\* 日　期: 14.07.02

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

uchar Ds1302\_Read\_Fu()

{

uchar i;

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

{

ACC=ACC>>1;

ACC7=IO;

CLK=1;

CLK=0;

}

return(ACC);

}

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

\*\* 函数名称: void Ds1302\_Dat\_Wr(uchar addr,uchar dat)

\*\* 功能描述: 按照地址向ds1302写入数据

\*\* 输　入: addr为要写数据的地址,dat为要写的数据

\*\*

\*\*

\*\* 输　出:

\*\*

\*\* 全局变量:

\*\* 调用模块:

\*\*

\*\* 作　者: 吴鉴鹰

\*\* 日　期: 14.07.02

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

void Ds1302\_Dat\_Wr(uchar addr,uchar dat)

{

RST=0;

CLK=0;

RST=1;

Ds1302\_Write\_Fu(addr);

Ds1302\_Write\_Fu(dat);

CLK=1;

RST=0;

}

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

\*\* 函数名称: uchar Ds1302\_Dat\_Rd(uchar addr)

\*\* 功能描述: 按照地址从ds1302读出数据

\*\* 输　入: uchar addr为要读的数据的地址

\*\*

\*\*

\*\* 输　出: return(dat)读出的数据

\*\*

\*\* 全局变量:

\*\* 调用模块:

\*\*

\*\* 作　者: 吴鉴鹰

\*\* 日　期: 14.07.02

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

uchar Ds1302\_Dat\_Rd(uchar addr)

{

uchar dat;

RST=0;

CLK=0;

RST=1;

Ds1302\_Write\_Fu(addr|0x01);

dat=Ds1302\_Read\_Fu();

CLK=1;

RST=0;

return(dat);

}

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

\*\* 函数名称: void Ds1302\_Protect\_Fu(bit flagg)

\*\* 功能描述: 是否写保护

\*\* 输　入: bit flagg（是否写保护的标志位）

\*\*

\*\*

\*\* 输　出:

\*\*

\*\* 全局变量:

\*\* 调用模块:

\*\*

\*\* 作　者: 吴鉴鹰

\*\* 日　期: 14.07.02

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

void Ds1302\_Protect\_Fu(bit flagg)

{

if(flagg)

Ds1302\_Dat\_Wr(0x8e,0x10); //禁止写

else

Ds1302\_Dat\_Wr(0x8e,0x00); //允许写

}

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

\*\* 函数名称: void Ds1302\_Set\_Time(uchar addr,uchar value)

\*\* 功能描述: 时间设置

\*\* 输　入: uchar addr（设置的位置）,uchar value（设置的值）

\*\*

\*\*

\*\* 输　出:

\*\*

\*\* 全局变量:

\*\* 调用模块:

\*\*

\*\* 作　者: 吴鉴鹰

\*\* 日　期: 14.07.02

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

void Ds1302\_Set\_Time(uchar addr,uchar value)

{

Ds1302\_Protect\_Fu(0);

Ds1302\_Dat\_Wr(addr,value);

Ds1302\_Protect\_Fu(1);

}

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

\*\* 函数名称: void Ds1302\_Set\_Plus(uchar adr)

\*\* 功能描述: 时间增加设置

\*\* 输　入: uchar adr

\*\*

\*\*

\*\* 输　出:

\*\*

\*\* 全局变量:

\*\* 调用模块:

\*\*

\*\* 作　者: 吴鉴鹰

\*\* 日　期: 14.07.02

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

void Ds1302\_Set\_Plus(uchar adr)

{

uchar dat;

dat=Ds1302\_Dat\_Rd(adr);

if((dat&0x0f)==9)

{

Ds1302\_Dat\_Wr(adr,((dat&0xf0)+0x10));

}

else

{

Ds1302\_Dat\_Wr(adr,++dat);

}

}

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

\*\* 函数名称: void Ds1302\_Set\_Reduc(uchar adr)

\*\* 功能描述: 时间减少设置

\*\* 输　入: uchar adr

\*\*

\*\*

\*\* 输　出:

\*\*

\*\* 全局变量:

\*\* 调用模块:

\*\*

\*\* 作　者: 吴鉴鹰

\*\* 日　期: 14.07.02

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

void Ds1302\_Set\_Reduc(uchar adr)

{

uchar dat;

dat=Ds1302\_Dat\_Rd(adr);

if((dat&0x0f)==0)

{

Ds1302\_Dat\_Wr(adr,(((dat&0xf0)-0x10)|0x09));

}

else

{

Ds1302\_Dat\_Wr(adr,--dat);

}

}

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

\*\* 函数名称: void day\_set(bit flagg)

\*\* 功能描述: 12/24小时时间设置

\*\* 输　入: bit flagg:为1为12时制,为0则为24小时制

\*\*

\*\*

\*\* 输　出:

\*\*

\*\* 全局变量:

\*\* 调用模块:

\*\*

\*\* 作　者: 吴鉴鹰

\*\* 日　期: 14.07.02

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

void day\_set(bit flagg) //flag为1为12时制,为0则为24小时制

{

uchar hour;

hour=(Ds1302\_Dat\_Rd(0x85)&0x7f); //保留小时寄存器中原有的时间值

Ds1302\_Protect\_Fu(0) ;

if(flagg)

{

Ds1302\_Dat\_Wr(0x84,0x80|hour);

}

else

{

Ds1302\_Dat\_Wr(0x84,0x00|hour);

}

Ds1302\_Protect\_Fu(1);

}

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

\*\* 函数名称: void Ds1302\_Get\_Time(systemtime \*time)

\*\* 功能描述: 从ds1302中提取时间

\*\* 输　入: systemtime \*time（形参 为存储时间结构体的首地址）

\*\*

\*\*

\*\* 输　出:

\*\*

\*\* 全局变量:

\*\* 调用模块:

\*\*

\*\* 作　者: 吴鉴鹰

\*\* 日　期: 14.07.02

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

void Ds1302\_Get\_Time(systemtime \*time)

{

uchar readvalue;

readvalue=Ds1302\_Dat\_Rd(Ds1302\_Second\_Addr);

time->second=((readvalue&0x70)>>4)\*10+(readvalue&0x0f);

readvalue=Ds1302\_Dat\_Rd(Ds1302\_Minute\_Addr);

time->minute=((readvalue&0x70)>>4)\*10+(readvalue&0x0f);

readvalue=Ds1302\_Dat\_Rd(Ds1302\_Hour\_Addr);

time->hour=((readvalue&0x30)>>4)\*10+(readvalue&0x0f);

readvalue=Ds1302\_Dat\_Rd(Ds1302\_Day\_Addr);

time->day=((readvalue&0x30)>>4)\*10+(readvalue&0x0f);

readvalue=Ds1302\_Dat\_Rd(Ds1302\_Week\_Addr);

time->week=(readvalue&0x07);

readvalue=Ds1302\_Dat\_Rd(Ds1302\_Month\_Addr);

time->month=((readvalue&0x10)>>4)\*10+(readvalue&0x0f);

readvalue=Ds1302\_Dat\_Rd(Ds1302\_Year\_Addr);

time->year=((readvalue&0xf0)>>4)\*10+(readvalue&0x0f);

datastring[0]=time->year/10;

datastring[1]=time->year%10;

datastring[2]=time->month/10;

datastring[3]=time->month%10;

datastring[4]=time->day/10;

datastring[5]=time->day%10;

datastring[6]=time->hour/10;

datastring[7]=time->hour%10;

datastring[8]=time->minute/10;

datastring[9]=time->minute%10;

datastring[10]=time->second/10;

datastring[11]=time->second%10;

datastring[12]=time->week;

}

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

\*\* 函数名称: void Ds1302\_Init\_Time(void)

\*\* 功能描述: 初始时间设置为2014-07-2 08时08分08秒

\*\* 输　入:

\*\*

\*\*

\*\* 输　出:

\*\*

\*\* 全局变量:

\*\* 调用模块:

\*\*

\*\* 作　者: 吴鉴鹰

\*\* 日　期: 14.07.02

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

void Ds1302\_Init\_Time(void)

{

day\_set(0);

Ds1302\_Set\_Time(Ds1302\_Second\_Addr,0x08);

Ds1302\_Set\_Time(Ds1302\_Minute\_Addr,0x08);

Ds1302\_Set\_Time(Ds1302\_Hour\_Addr,0x08);

Ds1302\_Set\_Time(Ds1302\_Day\_Addr,0x02);

Ds1302\_Set\_Time(Ds1302\_Month\_Addr,0x07);

Ds1302\_Set\_Time(Ds1302\_Year\_Addr,0x14);

Ds1302\_Set\_Time(Ds1302\_Week\_Addr,0x05);

}

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

\*\* 函数名称: void Ds1302\_Init\_Fuc(void)

\*\* 功能描述: 对DS1302的初始化函数

\*\* 输　入:

\*\*

\*\*

\*\* 输　出:

\*\*

\*\* 全局变量:

\*\* 调用模块:

\*\*

\*\* 作　者: 吴鉴鹰

\*\* 日　期: 14.07.02

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

void Ds1302\_Init\_Fuc(void)

{

uchar second=Ds1302\_Dat\_Rd(Ds1302\_Second\_Addr);

if(second&0x80)

Ds1302\_Set\_Time(Ds1302\_Second\_Addr,0);

}

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

\*\*作品：吴鉴鹰单片机驱动12864大字万年历设计和温度显示

\*\*申明：转载请标明作品来源 知识产权归作者本人所有！

\*\* 这一讲的主要内容: LCD12664液晶显示原理

\*\* 功能描述:单片机驱动12864大字万年历设计和温度显示

\*\* 输　入:

\*\* 液晶：Proteus元件库—AMPIRE128X64(无字库)

\*\* 输　出:

\*\*

\*\* 全局变量:

\*\* 调用模块:

\*\*

\*\* 作　者: 吴鉴鹰

\*\* 日　期: 14.07.2

\*\*作者所属单位：鉴鹰电子工作室

\*\*交流QQ群: 第一群： 332164870 第二群：194314772 联系QQ:1123942529

\*\*申明：转载请标明作品来源 知识产权归作者吴鉴鹰所有！

备注：有什么错误的地方，欢迎各大读者指正

O(∩\_∩)O~

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

#ifndef \_1302\_H\_\_

#define \_1302\_H\_\_

#define uchar unsigned char

#define uint unsigned int //宏定义

#define Ds1302\_Second\_Addr 0x80

#define Ds1302\_Minute\_Addr 0x82

#define Ds1302\_Hour\_Addr 0x84

#define Ds1302\_Day\_Addr 0x86

#define Ds1302\_Week\_Addr 0x8a

#define Ds1302\_Month\_Addr 0x88

#define Ds1302\_Year\_Addr 0x8c

/\*\*端口定义\*\*/

/\*sbit IO=P3^7;

sbit CLK=P3^6;

sbit RST=P3^5;

sbit ACC0=ACC^0;

sbit ACC7=ACC^7;

char datastring[13]; \*/ ////存放时间的数组

typedef struct

{

uchar second;

uchar minute;

uchar hour;

uchar day;

uchar week;

uchar month;

uchar year;

}systemtime;//定义的时间的类型

void Ds1302\_Write\_Fu(uchar dat);

/\*\*实时时钟读出一个字节\*\*/

uchar Ds1302\_Read\_Fu();

/\*\*写入数据\*\*/

void Ds1302\_Dat\_Wr(uchar addr,uchar dat); //addr为要写数据的地址,dat为要写的数据

/\*\*读出数据\*\*/

uchar Ds1302\_Dat\_Rd(uchar addr);

/\*\*是否写保护\*\*/

void Ds1302\_Protect\_Fu(bit Time\_Jinzhi\_Flag);

/\*\*时间设置\*\*/

void Ds1302\_Set\_Time(uchar addr,uchar value);

/\*\*时间增减设置\*\*/

void Ds1302\_Set\_Plus(uchar adr);

void Ds1302\_Set\_Reduc(uchar adr);

/\*\*12/24小时时间设置\*\*/

void day\_set(bit Time\_Jinzhi\_Flag); //flag为1为12时制,为0则为24小时制

/\*\*取时间\*\*/

void Ds1302\_Get\_Time(systemtime \*time);

/\*\*初始时间设置为2014-07-2 08时08分08秒\*\*/

void Ds1302\_Init\_Time(void);

/\*\*DS1302初始化\*\*/

void Ds1302\_Init\_Fuc(void) ;

#endif