

GM1302 C 语言参考程序

功 能：该程序（函数）是 GM1302 的驱动程序，包括对 1302 操作的各种子函数
和一个简单的 main（）主函数

说 明：该程序仅做参考用

```
#include <AT89X51.H>
```

```
#define uchar unsigned char
```

```
#define uint unsigned int
```

```
#define byte unsigned char
```

```
#define word unsigned int
```

```
sbit T_CLK = P2^0; /*实时时钟时钟线引脚定义 */
```

```
sbit T_IO = P2^1; /*实时时钟数据线引脚定义 */
```

```
sbit T_RST = P2^2; /*实时时钟复位线引脚定义 */
```

```
sbit ACC0 = ACC^0;
```

```
sbit ACC7 = ACC^7;
```

```
static uchar twdata[8]={0x58,0x59,0x23,0x31,0x12,0x06,0x04,0x80};/*定义初始化时钟数据数  
组，前 7 个为时钟数据，第 8 个为控制寄存器数据*/
```

```
static uchar trdata[7];/*定义 1302 时钟数据数组，用来存放读出的时钟数据（BCD 码） */
```

```
static uchar asc[14]; /*定义 ASC 码转换数组，用来存放时钟数据（BCD）转换过来的 ASC  
码 */
```

```
void RTInputByte(uchar); /*1302 输入子函数： 输入 1Byte */
```

```
uchar RTOutputByte(void); /* 1302 输出子函数： 1Byte */
```

```
void W1302(uchar, uchar);
```

```
uchar R1302(uchar);
```

```
void Set1302(uchar *); /* 单字节设置时间函数 */
```

```
void Bcd2asc(void);
```

```
void Get1302(uchar curtime[]);/* 单字节读取 1302 当前时间的函数 */
```

```
void BurstW1302T(uchar *); /* 多字节设置时间函数 */
```

```
void BurstR1302T(uchar *); /* 多字节读取 1302 当前时间的函数 */
```

```
void BurstW1302R(uchar *); /* 单字节设置 RAM 函数 */
```

```
void BurstR1302R(uchar *); /* 多字节读取 RAM 函数 */
```

函 数 名：RTInputByte()

功 能：实时时钟写入一字节

说 明：往 GM1302 写入 1Byte 数据（内部函数）

入口参数：d 写入的数据

返 回 值：无

```
void RTInputByte(uchar d)
```

```

{
    uchar i;
    ACC = d;
    for(i=8; i>0; i--)
    {
        T_IO = ACC0;          /*相当于汇编中的 RRC */
        T_CLK = 1;
        T_CLK = 0;
        ACC = ACC >> 1;
    }
}
}
/*****
函数名: RTOutputByte()
功能: 实时时钟读取一字节
说明: 从 GM1302 读取 1Byte 数据 (内部函数)
入口参数: 无
返回值: ACC
*****/
uchar RTOutputByte(void)
{
    uchar i;
    for(i=8; i>0; i--)
    {
        ACC = ACC >>1;      /*相当于汇编中的 RRC */
        ACC7 = T_IO;
        T_CLK = 1;
        T_CLK = 0;
    }
    return(ACC);
}
/*****
函数名: W1302()
功能: 往 GM1302 写入数据
说明: 先写地址, 后写命令/数据 (内部函数)
调用: RTInputByte(), RTOutputByte()
入口参数: ucAddr: GM1302 地址, ucData: 要写的数据
返回值: 无
*****/
void W1302(uchar ucAddr, uchar ucDa)
{
    T_RST = 0;
    T_CLK = 0;
    T_RST = 1;
    RTInputByte(ucAddr);    /* 地址, 命令 */
}

```

```

    RTInputByte(ucDa);          /* 写 1Byte 数据*/
    T_CLK = 1;
    T_RST = 0;
}
/*****

函数名: R1302()
功能: 读取 GM1302 某地址的数据
说明: 先写地址, 后读命令/数据 (内部函数)
调用: RTInputByte(), RTOutputByte()
入口参数: ucAddr: GM1302 地址
返回值: ucData :读取的数据
*****/

uchar R1302(uchar ucAddr)
{
    uchar ucData;
    T_RST = 0;
    T_CLK = 0;
    T_RST = 1;
    RTInputByte(ucAddr);          /* 地址, 命令 */
    ucData = RTOutputByte();      /* 读 1Byte 数据 */
    T_CLK = 1;
    T_RST = 0;
    return(ucData);
}
/*****

函数名: BurstW1302T()
功能: 往 GM1302 写入时钟数据(多字节方式)
说明: 先写地址, 后写命令/数据
调用: RTInputByte()
入口参数: pWClock: 时钟数据地址 格式为: 秒 分 时 日 月 星期 年 控制
          8Byte (BCD 码)1B 1B 1B 1B 1B 1B 1B 1B
返回值: 无
*****/

void BurstW1302T(uchar *pWClock)
{
    uchar i;
    W1302(0x8e,0x00);          /* 控制命令,WP=0,写操作?*/
    T_RST = 0;
    T_CLK = 0;
    T_RST = 1;
    RTInputByte(0xbe);          /* 0xbe:时钟多字节写命令 */
    for (i = 8; i>0; i--)      /*8Byte = 7Byte 时钟数据 + 1Byte 控制*/
    {
        RTInputByte(*pWClock); /* 写 1Byte 数据*/
    }
}

```

```

        pWClock++;
    }
    T_CLK = 1;
    T_RST = 0;
}

```

函数名: BurstR1302T()

功能: 读取 GM1302 时钟数据

说明: 先写地址/命令, 后读数据(时钟多字节方式)

调用: RTInputByte(), RTOutputByte()

入口参数: pRClock: 读取时钟数据地址 格式为: 秒 分 时 日 月 星期 年
7Byte (BCD 码)1B 1B 1B 1B 1B 1B 1B

返回值: 无

```

void BurstR1302T(uchar *pRClock)

```

```

{
    uchar i;
    T_RST = 0;
    T_CLK = 0;
    T_RST = 1;
    RTInputByte(0xbf);          /* 0xbf:时钟多字节读命令 */
    for (i=8; i>0; i--)
    {
        *pRClock = RTOutputByte(); /* 读 1Byte 数据 */
        pRClock++;
    }
    T_CLK = 1;
    T_RST = 0;
}

```

函数名: BurstW1302R()

功能: 往 GM1302 寄存器数写入数据(多字节方式)

说明: 先写地址, 后写数据(寄存器多字节方式)

调用: RTInputByte()

入口参数: pWReg: 寄存器数据地址

返回值: 无

```

void BurstW1302R(uchar *pWReg)

```

```

{
    uchar i;
    W1302(0x8e,0x00);          /* 控制命令,WP=0,写操作*/
    T_RST = 0;
    T_CLK = 0;
    T_RST = 1;
}

```

```

RTInputByte(0xfe);          /* 0xfe:时钟多字节写命令 */
for (i=31; i>0; i--)        /* 31Byte 寄存器数据 */
{
    RTInputByte(*pWReg);    /* 写 1Byte 数据*/
    pWReg++;
}
T_CLK = 1;
T_RST = 0;
}
/*****

```

函数名: BurstR1302R()

功能: 读取 GM1302 寄存器数据

说明: 先写地址, 后读命令/数据(寄存器多字节方式)

调用: RTInputByte(), RTOutputByte()

入口参数: pRReg: 寄存器数据地址

返回值: 无

```

/*****
void BurstR1302R(uchar *pRReg)
{
    uchar i;
    T_RST = 0;
    T_CLK = 0;
    T_RST = 1;
    RTInputByte(0xff);      /* 0xff:时钟多字节读命令 */
    for (i=31; i>0; i--)    /* 31Byte 寄存器数据 */
    {
        *pRReg = RTOutputByte(); /* 读 1Byte 数据 */
        pRReg++;
    }
    T_CLK = 1;
    T_RST = 0;
}
/*****

```

函数名: Set1302()

功能: 设置初始时间, 以单字节方式连续设置 GM1302 的 7 个时钟寄存器

说明: 先写地址, 后写数据(单字节方式)

调用: W1302()

入口参数: pClock: 设置时钟数据地址 格式为: 秒 分 时 日 月 星期 年

7Byte (BCD 码)1B 1B 1B 1B 1B 1B 1B

返回值: 无

```

/*****
void Set1302(uchar *pClock)
{

```

```

    uchar i;
    uchar ucAddr = 0x80;
    //W1302(0x8e,0x00);          /* 控制命令,WP=0,写操作?*/
    for(i=7; i>0; i--)
    {
        W1302(ucAddr,*pClock); /* 秒 分 时 日 月 星期 年 */
        pClock++;
        ucAddr +=2;
    }
    W1302(0x8e,0x80);          /* 控制命令,WP=1,写保护?*/
}
/*****

```

函数名: Get1302()

功能: 单字节方式读取 GM1302 当前时间 (7 个时钟数据)

说明:

调用: R1302()

入口参数: ucCurtime: 保存当前时间地址。当前时间格式为: 秒 分 时 日 月 星期 年

返回值: 无

```

    *****/
void Get1302(uchar ucCurtime[])
{
    uchar i;
    uchar ucAddr = 0x81;
    for (i=0; i<7; i++)
    {
        ucCurtime[i] = R1302(ucAddr);/*格式为: 秒 分 时 日 月 星期 年 */
        ucAddr += 2;
    }
}
/*****

```

函数名: Bcd2asc()

功能: 将读取的 GM1302 时钟数据 (压缩型 bcd 码) 转换成 asc 码存储在 asc 数组中

说明:

调用:

入口参数:

返回值: 无

```

    *****/
void Bcd2asc(void)
{
    uchar i,j;

```

```

    for (j=0,i=0; i<7; i++)
    {
        asc[j++]=(trdata[i]&0xf0)>>4|0x30 ;/*格式为: 秒 分 时 日 月 星期 年 */
        asc[j++] =trdata[i]&0x0f|0x30;
    }
}
/*****

```

函数名: main()

功能: 先设置 GM1302 的时间, 然后循环读取 GM1302 的时钟数据并保存,
将读取的时钟数据转换成 asc 码保存

说明: 该主函数很简单, 用户可以根据需要来选择多字节方式或者单字节方式

调用:

入口参数:

返回值: 无

```

    *****/
void main(void)
{
    BurstW1302T(twdata);    /* 多字节方式设置 GM1302 时间 */
    //Set1302(twdata);     /* 单字节方式设置 GM1302 时间 */
    while(1)
    {
        BurstR1302T(trdata); /* 多字节方式读取 GM1302 时间*/
        //Get1302(trdata);   /* 单字节方式读取 GM1302 时间*/
        Bcd2asc();          /* 时钟数据转换成 asc 码*/
    }
}
}

```