```
/*******************************************************
          modbus-rtu 通讯规约

通讯方式: rs-485 半双功
校验方式: crc16
停止位: 2位
编写: 孙可
编写日期: 2008年6月18日
版本: v0.2
编程思路:
    1.串口中断允许自动接收总线上的信息,当接收的
    字节后超过3.5个字节时间没有新的字节认为本次
    接收完成,接收完成标志置1;如果接收完成标志已
    经置1又有数据进来则丢弃新来的数据。
    2.串口接收数据的处理, 当接收完成标志置1进入
    接收数据处理, (1)首先判断接收的第一位数据与
    本机地址是否相同,如果不相同清空接收缓存不发
    送任何信息; (2)接收的第一位数据与本机地址相
    同,则对接收缓存中的数据进行crc16校验,如果接
    收的校验位与本校验结果不相同清空接收缓存不发
    送任何信息;
    (3)如果crc16校验正确则根据数据串中的命令码进
    行相应的处理。
*******************************************************/
#include "modbus.h"


u8 Com0_id = 0x05;//本机串口0的通讯地址
u8 Uart0_rev_buff[100];//com0串口接收缓冲区
u8 Uart0_send_buff[100];//com0串口发送缓冲区
vu8 Uart0_rev_count;
vs8 Uart0_send_counter = 0;
vu8 Uart0_rev_comflag;
vu8 Crc_counter = 0;//com0校验计数器
vu8 *Uart0_send_pointer = Uart0_send_buff;//com0串口发送指针

vu16 Mkgz_bz = 0;//模块故障标志1:输入异常, 2:过压, 3:欠压, 4:过温
vu16 Out_current = 50;//输出电流
vu16 Out_voltage = 240;//输出电压
vu16 Mkzt_bz = 0;//模块状态标志
vu16 OutX_current = 1000;//输出限流
vu16 Jc_voltage = 2530;//均充电压
vu16 Fc_voltage = 2400;//浮充电压
vu16 user_day = 1825;//使用天数


void Delay(vu32 nCount);
unsigned short getCRC16(volatile unsigned char *ptr,unsigned char len) ;
void mov_data(u8 a[100],u8 b[100],u8 c);
void Modbus_Function_3(void);
void Modbus_Function_6(void);
/****************************************
函数名称: crc16校验
函数功能: crc16校验
函数输入: 字节指针*ptr, 数据长度len
函数返回: 双字节crc
函数编写: 孙可
编写日期: 2008年6月9日
函数版本: v0.2
****************************************/
unsigned short getCRC16(volatile unsigned char *ptr,unsigned char len)
{
    unsigned char i;
    unsigned short crc = 0xFFFF;
    if(len==0)
    {
        len = 1;
    }
    while(len--)
    {
        crc ^= *ptr;
        for(i=0; i<8; i++)
        {
            if(crc&1)
                {
                crc >>= 1;
                crc ^= 0xA001;
                }
                else
                {
                crc >>= 1;
                }
        }
        ptr++;
    }
    return(crc);
}

/****************************************
    块数据复制数据函数
功能：把数组a的c个数据复制到数组b中
输入：指针a,指针b,数据个数c
返回：无
编写：孙可
编写日期：2008年3月28日
版本：v0.1
****************************************/
void mov_data(u8 a[100],u8 b[100],u8 c)
{
    u8 i;
    for(i=c; i>0; i--)
    {
            a[i] = b[i];
    }
}

///////////////////////////////////////////////////////////////////////
 void Modbus_Function_3(void)
{
    u16 tempdress = 0;
    u8 i = 3;
    u16 crcresult;
    tempdress = (Uart0_rev_buff[2] << 8) + Uart0_rev_buff[3];
    if((tempdress >= 0x0120) & (tempdress + Uart0_rev_buff[5] < 0x0132))
    {
        Uart0_send_buff[0] = Com0_id;
        Uart0_send_buff[1] = 0x03;
        Uart0_send_buff[2] = 2 * Uart0_rev_buff[5];
        Uart0_send_counter = 2 * Uart0_rev_buff[5] + 3;

        switch(tempdress)
        {
            case 0x0120:
                {
                Uart0_send_buff[i] = Mkgz_bz & 0xff;
                        i++;
                Uart0_send_buff[i] = (Mkgz_bz >> 8) & 0xff;
                i++;
            }//后面不放break的目的是继续往下执行
            case 0x0122:
                {
                Uart0_send_buff[i] = Out_voltage & 0xff;
                        i++;
                Uart0_send_buff[i] = (Out_voltage >> 8) & 0xff;
                i++;
                }
            case 0x0124:
                {
```

```
                Uart0_send_buff[i] = Out_current & 0xff;
                        i++;
                Uart0_send_buff[i] = (Out_current >> 8) & 0xff;
                i++;
                }
            case 0x0126:
                {
                Uart0_send_buff[i] = Mkzt_bz & 0xff;
                        i++;
                Uart0_send_buff[i] = (Mkzt_bz >> 8) & 0xff;
                i++;
                }
            case 0x0128://这个地址是备用的里面的数据没有意义
                {
                Uart0_send_buff[i] = 0x00;
                        i++;
                Uart0_send_buff[i] = 0x00;
                i++;
                }
            case 0x012A:
                {
                Uart0_send_buff[i] = OutX_current & 0xff;
                        i++;
                Uart0_send_buff[i] = (OutX_current >> 8) & 0xff;
                        i++;
                }
            case 0x012C:
                {
                Uart0_send_buff[i] = Jc_voltage & 0xff;
                        i++;
                Uart0_send_buff[i] = (Jc_voltage >> 8) & 0xff;
                        i++;
                }
            case 0x012E:
                {
                Uart0_send_buff[i] = Fc_voltage & 0xff;
                        i++;
                Uart0_send_buff[i] = (Fc_voltage >> 8) & 0xff;
                        i++;
                }
            case 0x0130:
                {
                Uart0_send_buff[i] = 0x00;
                        i++;
                Uart0_send_buff[i] = 0x00;
                        i++;
                }
        }
        //UCSRB |= (1<<TXCIE)|(1<<TXEN);//发送、发送中断允许
        crcresult = getCRC16(Uart0_send_buff,Uart0_send_counter);
        Uart0_send_buff[Uart0_send_counter] = crcresult & 0xff;
        Uart0_send_buff[Uart0_send_counter+1] = (crcresult >> 8) & 0xff;
        Uart0_send_counter = Uart0_send_counter+2;
        Uart0_send_pointer = Uart0_send_buff;
                USART_SendData(USART1, *Uart0_send_pointer++);
                USART_ITConfig(USART1, USART_IT_TXE, ENABLE);
    }
}
/////////////////////////////////////////////////////////////
void Modbus_Function_6(void)
{
    u16 tempdress = 0;
    u8 tx_flat = 0;
    u16 crcresult;
    tempdress = (Uart0_rev_buff[2]<<8) + Uart0_rev_buff[3];
    switch(tempdress)
    {
        case 0x0126:
        {
            Mkzt_bz = (Uart0_rev_buff[4]<<8) + Uart0_rev_buff[5];
            if(user_day > 0)
                {
                tx_flat = 1;
                }
        }break;
        case 0x012A:
        {
            OutX_current = (Uart0_rev_buff[4]<<8) + Uart0_rev_buff[5];
            if(user_day > 0)
                {
                tx_flat = 1;
                }
        }break;
        case 0x012C:
        {
            Jc_voltage = (Uart0_rev_buff[4]<<8) + Uart0_rev_buff[5];
            if(user_day > 0)
                {
                tx_flat = 1;
                }
        }break;
        case 0x012E:
        {
            Fc_voltage = (Uart0_rev_buff[4]<<8) + Uart0_rev_buff[5];
            if(user_day > 0)
                {
                tx_flat = 1;
                }
        }break;
        case 0x01EE:
        {
            user_day = (Uart0_rev_buff[4]<<8) + Uart0_rev_buff[5];
            tx_flat = 1;
            //eeprom_write_word (&user_day_eep,user_day);
        }break;
        default: //命令码无效不应答
                {
                    tx_flat = 0;
                }
    }
    if(tx_flat == 1)
    {
        Uart0_send_buff[0] = Com0_id;
        Uart0_send_buff[1] = 0x06;
        Uart0_send_buff[2] = Uart0_rev_buff[2];
        Uart0_send_buff[3] = Uart0_rev_buff[3];
        Uart0_send_buff[4] = Uart0_rev_buff[4];
        Uart0_send_buff[5] = Uart0_rev_buff[5];
        Uart0_send_counter = 6;

        //UCSRB |= (1<<TXCIE)|(1<<TXEN);//发送、发送中断允许
        crcresult = getCRC16(Uart0_send_buff,Uart0_send_counter);
        Uart0_send_buff[Uart0_send_counter] = crcresult & 0xff;
        Uart0_send_buff[Uart0_send_counter+1] = (crcresult >> 8) & 0xff;
        Uart0_send_counter = Uart0_send_counter+2;
        Uart0_send_pointer = Uart0_send_buff;
            USART_SendData(USART1, *Uart0_send_pointer++);
                USART_ITConfig(USART1, USART_IT_TXE, ENABLE);
    }
}
/////////////////////////////////////////////////////////////
void Com0_Communication(void)
{
    s8 i =0;
    if(Uart0_rev_comflag == 1)//接收完成标志=1处理，否则退出
    {
        if(Uart0_rev_buff[0] == Com0_id)//地址错误不应答
        {
            unsigned short crcresult;
            unsigned char temp[2];
            crcresult = getCRC16(Uart0_rev_buff,Crc_counter-2);
            temp[1] = crcresult & 0xff;
            temp[0] = (crcresult >> 8) & 0xff;
            if((Uart0_rev_buff[Crc_counter-1] == temp[0])&&(Uart0_rev_buff[Crc_count
er-2] == temp[1]))//crc校验错误不应答
                {
                //SETBIT(PORTC,PC6);
                Delay(1);
                    switch(Uart0_rev_buff[1])
                        {
                            case 0x03:
                                {
                        if(user_day > 0)
                                        {
                            Modbus_Function_3();
                                }
                                }
                            break;
                            case 0x06:
                                {
                        Modbus_Function_6();
                                }
                            break;
                    }
                }
        }
        Uart0_rev_comflag = 0;

        for(i = 100;i > -1;i--)
        {
                    Uart0_rev_buff[i] = 0;
        }
    }

}
/********************************************************************************
* Function Name  : Delay
* Description    : Inserts a delay time.
* Input          : nCount: specifies the delay time length.
* Output         : None
* Return         : None
*******************************************************************************/
void Delay(vu32 nCount)
{
  for(; nCount != 0; nCount--);
}
//////////////////////////////////////////////////////////////////////////
```