

CHTA31IW 数字温湿度传感器 I2C 范例:

```
#define Sensor_SCLC _pfc3 // SCL 脚状态设定
#define Sensor_SDAC _pfc2 // SDA 脚状态设定
#define Sensor_SCL _pf3 //主控端使用 PF3 作为 SCL 输出脚
#define Sensor_SDA _pf2 //主控端使用 PF2 作为 SDA 输出脚
#define Sensor_SDAPU _pfpu2 //SDA Pull High 状态
#define LED_port _ph0 //PH0 控制 LED 灯
#define LED_portC _phc0 //PH0 脚状态设定

void I2C_START(void); //起始信号
void I2C_STOP(void); //停止信号
void send_ACK(void); //返回 ACK 信号
void send_NON_ACK(void); //返回无 ACK 信号
_Bool CheckACK(void);
void WriteByte(u8); //写操作
u8 ReadByte(void); //读操作
void Sensor_write_command(u8 slave_ID); //发送命令
void Sensor_Read_Operation(u8 slave_ID); //读回数据
void Sensor_WakeUp (u8 slave_ID); //唤醒
void RH_Module_CheckSum(u8 *Buff,u8 Len); //数据校验
u8 Sensor_Read_Byte(void);
void Delayms(u16);

u8 CheckSum_MSB, CheckSum_LSB; //校验变量高低位
u8 RH_MSB, RH_LSB, TS_MSB, TS_LSB; //温湿度高低位
u16 Temp, Hum; //温湿度变量
extern u8 Sensor_buffer[10]; //定义数组

/*****
**/
void main()
{
/*****
**/
    _wdtc=0b10101111; LED_portC=0; LED_port=0; //Sensor_SDAPU=1;
/*****
**/
```

```

while(1)
{
    Sensor_WakeUp(0xB8);          //唤醒
    Sensor_write_command(0xB8);  //写命令
    Delayms(2);
    Sensor_Read_Operation(0xB8); //读数据
    Delayms(2000);                //延迟 2s
}
}

/*****
**/
void I2C_START(void)
{
    Sensor_SCL=0; Sensor_SDA=1;
    Sensor_SCLC=0; Sensor_SDAC=0;
    Sensor_SCL=1; GCC_DELAY(36);
    Sensor_SDA=0; GCC_DELAY(36);
    Sensor_SCL=0;
}

void I2C_STOP(void)
{
    Sensor_SDA=0; Sensor_SDAC=0;
    GCC_DELAY(36);
    Sensor_SCL=1; GCC_DELAY(36);
    Sensor_SDA=1; GCC_DELAY(36);
    //Sensor_SCL=0;
}

void send_ACK(void)
{
    Sensor_SDA=0; Sensor_SDAC=0;
    GCC_DELAY(36);
    Sensor_SCL=1; GCC_DELAY(36);
    Sensor_SCL=0; GCC_DELAY(36);
    Sensor_SDAC=1;
}

void send_NON_ACK(void)
{

```

```

    Sensor_SDA=1; Sensor_SDAC=0;
    GCC_DELAY(36);
    Sensor_SCL=1; GCC_DELAY(36);
    Sensor_SCL=0; GCC_DELAY(36);
}

_Bool CheckACK(void)
{
    u8 i=200;
    Sensor_SDAC=1;
    GCC_DELAY(36);
    Sensor_SCL=1;
    GCC_DELAY(36);
    while(i--)
    {
        if(!Sensor_SDA) { Sensor_SCL=0; return 1;}
        GCC_DELAY(200);
    }
    Sensor_SCL=0;
    return 0;
}

void WriteByte(u8 I2CData)
{
    u8 temp=0b10000000;
    Sensor_SDAC=0;
    while (temp!=0)
    {
        Sensor_SDA=(I2CData & temp)?1:0;
        Sensor_SCL=1; GCC_DELAY(36);
        Sensor_SCL=0; GCC_DELAY(36);
        temp>>=1;
    }
}

u8 ReadByte(void)
{
    u8 retval=0, temp=0b10000000;
    Sensor_SDAC=1;
    while (temp!=0)

```

```

    {
        Sensor_SCL=1; GCC_DELAY(100);
        if(Sensor_SDA) retval|=temp;
        Sensor_SCL=0; GCC_DELAY(100);
        temp>>=1;
    }
    return retval;
}

u8 Sensor_Read_Byte(void)
{
    u8 cnt; u8 data;
    for(cnt=0; cnt<8; cnt++)
    {
        data=data<<1;
        Sensor_SCL=1;
        GCC_DELAY(100);
        if(Sensor_SDA)
        {
            data |=0x01;
        }
        Sensor_SCL=0;
        GCC_DELAY(100);
    }
    return data;
}

void Sensor_WakeUp (u8 slave_ID)
{
    u8 DeviceID=((slave_ID/2)<<1);
    I2C_START();           //IIC 起始信号
    WriteByte(DeviceID|0); //发送 CHTA30 IIC 地址+写
    send_NON_ACK();
    Delayms(1);
    I2C_STOP();           //IIC 停止信号
}

void Sensor_write_command (u8 slave_ID)
{
    u8 DeviceID=((slave_ID/2)<<1);

```

```

I2C_START();          //IIC 起始信号
    WriteByte(DeviceID|0);          //发送 CHTA30 IIC 地址+写
    if(CheckACK())          //等待 CHTA30 应答
    {
        WriteByte(0x03);          //发送 0x03
        CheckACK();          //等待 CHTA30 应答
        WriteByte(0x00);          //发送 0x00
        CheckACK();          //等待 CHTA30 应答
        WriteByte(0x04);          //发送 0x04
        CheckACK();          //等待 CHTA30 应答
        I2C_STOP();          //IIC 停止信号

    }
    else
    {
        LED_port=1;
        I2C_STOP();          //IIC 停止信号
    }
}

void Sensor_Read_Operation(u8 slave_ID)
{
    u8 Sensor_buffer[10];
    u8 DeviceID=((slave_ID/2)<<1);
    I2C_START();          //IIC 起始信号
    WriteByte(DeviceID|1);          //发送 CHTA30 IIC 地址+读
    if(CheckACK())          //等待 CHTA30 应答
    {
        Sensor_buffer[0]=Sensor_Read_Byte();//写入数据
        send_ACK();          //发送应答信号
        Sensor_buffer[1]=Sensor_Read_Byte();//写入数据
        send_ACK();          //发送应答信号
        Sensor_buffer[2]=Sensor_Read_Byte();//写入数据
        send_ACK();          //发送应答信号
        Sensor_buffer[3]=Sensor_Read_Byte();//写入数据
        send_ACK();          //发送应答信号
        Sensor_buffer[4]=Sensor_Read_Byte();//写入数据
        send_ACK();          //发送应答信号
    }
}

```

```

        Sensor_buffer[5]=Sensor_Read_Byte();//写入数据
        send_ACK();                      //发送应答信号
        Sensor_buffer[6]=Sensor_Read_Byte();//写入数据
        send_ACK();                      //发送应答信号
        Sensor_buffer[7]=Sensor_Read_Byte();//写入数据
        send_NON_ACK();
        I2C_STOP();                      //IIC 停止信号
        RH_Module_CheckSum(&Sensor_buffer[0],6);
        if((Checksum_LSB == Sensor_buffer[6]) && (Checksum_MSB ==
Sensor_buffer[7]))
        {
            RH_MSB = Sensor_buffer[2];
            RH_LSB = Sensor_buffer[3];
            TS_MSB = Sensor_buffer[4];
            TS_LSB = Sensor_buffer[5];
            Temp=(((TS_MSB)<<8)+TS_LSB);
            Hum=(((RH_MSB)<<8)+RH_LSB);
            GCC_NOP();
        }

    }
    else
    {
        LED_port=1;
        I2C_STOP();                      //IIC 停止信号
    }

}

```

```

void RH_Module_CheckSum(u8 *Buff,u8 Len)
{
    unsigned short a_temp_crc = 0xFFFF;
    unsigned char i;
    while(Len--)
    {
        a_temp_crc ^= *Buff++;
        for(i = 0 ; i < 8 ; i++)
        {

```

```

        if(a_temp_crc & 0x01)
        {
            a_temp_crc >>= 1;
            a_temp_crc ^= 0xA001;
        }
        else
        {
            a_temp_crc >>= 1;
        }
    }
}
Checksum_LSB = a_temp_crc & 0x00FF;
Checksum_MSB = (a_temp_crc & 0xFF00) >> 8;
}

```

```

void Delayms(u16 del)
{
    u16 i;
    for(i=0;i<del;i++) GCC_DELAY(2000);
}

```