**线定义：**

红色： 3.3V

黑色：SDA

黄色：SCL

绿色：GND



**算法：**

* NA2302器件地址为：0x00
* 压力0x06 温度地址0x09



量程计算

P（压力）=【Value（采集值）/8388608-0.2】/0.7 \*Pr（压力范围）

举例：

1MPA压力传感器，IIC得到Value=4194304，于是

P（压力）=【4194304/8388608-0.2】/0.7 \*（1MPA-0） =0.428MPA

**程序：**

Iic.h

#ifndef \_\_IIC\_H

#define \_\_IIC\_H

#include "sys.h"

#include "delay.h"

//IO操作函数

#define IIC\_SCL PAout(5) //SCL

#define IIC\_SDA PAout(4) //SDA

#define READ\_SDA PAin(4) //输入SDA

//IO方向设置 CRL(Pin0~7)

#define SDA\_IN() {SDA\_IN\_Status();}

#define SDA\_OUT() {SDA\_OUT\_Status();}

void IIC\_Initializes(void);

//IIC所有操作函数

void IIC\_Init(void); //初始化IIC的IO口

void IIC\_Start(unsigned short num); //发送IIC开始信号

void IIC\_Stop(unsigned short num); //发送IIC停止信号

unsigned char IIC\_Wait\_Ack(unsigned short num); //IIC等待ACK信号

void IIC\_Ack(unsigned short num); //IIC发送ACK信号

void IIC\_NAck(unsigned short num); //IIC不发送ACK信号

void IIC\_Send\_Byte(unsigned short num, unsigned char T\_Data); //IIC发送一个字节

unsigned char IIC\_Read\_Byte(unsigned short num); //IIC读取一个字节

unsigned char IIC\_Write\_Len\_Byte(unsigned short num, unsigned char dev\_addr,unsigned char reg\_addr, unsigned char data\_len, unsigned char\* wData);

unsigned char IIC\_Read\_Len\_Byte(unsigned short num, unsigned char dev\_addr,unsigned char reg\_addr, unsigned char data\_len, unsigned char\* rData);

void Get\_CalData\_IIC (uint8\_t num, uint8\_t Mode, uint8\_t OutFormat);

void NSA2302\_CalData(void);

void SDA\_OUT\_Status(void);

void SDA\_IN\_Status(void);

#endif

//////////////////////////////////////////////////////////////////////////////////////////////////////

Iic.c

#include "usart.h"

#include "iic.h"

unsigned short Num = 10;

extern uint8\_t Cal\_Pdata[3];//读取校准后压力数组

extern uint8\_t Cal\_Tdata[3];//读取校准后温度数组

extern uint8\_t Cal\_data[6];//读取校准后数组

extern float Cal\_PData;//读取校准后压力数组

extern float Cal\_TData;//读取校准后温度数组

extern uint8\_t ID\_Num\_W[6];

extern uint8\_t ID\_Num\_R[6];

extern uint8\_t Drdy;

////////////////实际测试，最快num=4，否则IO反转速率不够//////////////

void IIC\_Initializes(void) //IIC初始化

{

IIC\_Init();

}

//初始化IIC

void IIC\_Init(void)

{

GPIO\_InitTypeDef GPIO\_InitStructure;

RCC\_APB2PeriphClockCmd( RCC\_APB2Periph\_GPIOA, ENABLE );

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_4|GPIO\_Pin\_5;

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_Out\_OD ; //开漏输出

GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz;

GPIO\_Init(GPIOA, &GPIO\_InitStructure);

IIC\_SCL=1;

IIC\_SDA=1;

}

void SDA\_OUT\_Status(void)

{

GPIO\_InitTypeDef GPIO\_InitStructure;

RCC\_APB2PeriphClockCmd( RCC\_APB2Periph\_GPIOA, ENABLE );

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_4;

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_Out\_OD ; //开漏输出

GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz;

GPIO\_Init(GPIOA, &GPIO\_InitStructure);

}

void SDA\_IN\_Status(void)

{

GPIO\_InitTypeDef GPIO\_InitStructure;

RCC\_APB2PeriphClockCmd( RCC\_APB2Periph\_GPIOA, ENABLE );

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_4;

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_IN\_FLOATING ; //浮空输入

GPIO\_Init(GPIOA, &GPIO\_InitStructure);

}

//产生IIC起始信号

void IIC\_Start(unsigned short num)

{

SDA\_OUT(); //SDA线输出

IIC\_SDA=1;

IIC\_SCL=1;

delay\_us(num);

IIC\_SDA=0; //Start标志

delay\_us(num);

IIC\_SCL=0; //钳住I2C总线，准备发送或接收数据

}

//产生IIC停止信号

void IIC\_Stop(unsigned short num)

{

IIC\_SCL=0;

SDA\_OUT(); //SDA线输出

IIC\_SDA=0; //STOP标志

delay\_us(num);

IIC\_SCL=1;

delay\_us(num);

IIC\_SDA=1; //发送结束信号

delay\_us(num);

}

//等待应答信号到来

//返回值：1，接收应答失败

// 0，接收应答成功

unsigned char IIC\_Wait\_Ack(unsigned short num)

{

unsigned char ucErrTime=0;

delay\_us(num/4);IIC\_SCL=1;

SDA\_IN(); //SDA设置为输入

while(READ\_SDA)

{

ucErrTime++;

if(ucErrTime>250)

{

IIC\_Stop(num);

return 0xff;

}

}

IIC\_SCL=0;

delay\_us(num/2);

IIC\_SDA=1; //时钟输出0

return 0x00;

}

//产生ACK应答

void IIC\_Ack(unsigned short num)

{

IIC\_SCL=0;

SDA\_OUT();

IIC\_SDA=0;

delay\_us(num/2);

IIC\_SCL=1;

delay\_us(num/2);

IIC\_SCL=0;

}

//不产生ACK应答

void IIC\_NAck(unsigned short num)

{

IIC\_SCL=0;

SDA\_OUT();

IIC\_SDA=1;

delay\_us(num/2);

IIC\_SCL=1;

delay\_us(num/2);

IIC\_SCL=0;

}

//IIC发送一个字节

//返回从机有无应答

//1，有应答

//0，无应答

void IIC\_Send\_Byte(unsigned short num, unsigned char T\_Data)

{

unsigned char t;

SDA\_OUT();

IIC\_SCL=0; //拉低时钟开始数据传输

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

{

IIC\_SDA=(T\_Data&0x80)>>7;

T\_Data<<=1;

delay\_us(num/4);

IIC\_SCL=1;

delay\_us(num/2);

IIC\_SCL=0;

delay\_us(num/4);

}

}

//读1个字节，ack=1时，发送ACK，ack=0，发送nACK

unsigned char IIC\_Read\_Byte(unsigned short num)

{

unsigned char i,receive=0;

SDA\_IN(); //SDA设置为输入

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

{

IIC\_SCL=0;

delay\_us(num/2);

IIC\_SCL=1;

receive<<=1;

if(READ\_SDA)receive++;

delay\_us(num/2);

}

return receive;

}

unsigned char IIC\_Read\_Len\_Byte(unsigned short num, unsigned char dev\_addr,unsigned char reg\_addr, unsigned char data\_len, unsigned char\* rData)

{

unsigned char i = 0;

unsigned char dev\_addr\_r = dev\_addr | 0x01;

unsigned char dev\_addr\_w = dev\_addr & 0xFE; //防止读写IIC地址出错，写统一地址即可

IIC\_Start(num); //IIC开始

IIC\_Send\_Byte(num, dev\_addr\_w); //先发设备地址

if(IIC\_Wait\_Ack(num) == 0xff) //没有等到Ack返回则返回0xFF

{ return 0xff; }

IIC\_Send\_Byte(num, reg\_addr); //开始发送寄存器地址

if(IIC\_Wait\_Ack(num) == 0xff)

{ return 0xff; }

IIC\_Start(num); //IIC开始

IIC\_Send\_Byte(num, dev\_addr\_r); //开始发读命令

if(IIC\_Wait\_Ack(num) == 0xff)

{ return 0xff; }

for(i=0; i<data\_len-1; i++)

{

rData[i] = IIC\_Read\_Byte(num);

IIC\_Ack(num); //发送ACK

}

rData[data\_len -1] = IIC\_Read\_Byte(num);

IIC\_NAck(num);//发送nACK

IIC\_Stop(num); //产生一个停止条件

return 0x00;

}

unsigned char IIC\_Write\_Len\_Byte(unsigned short num, unsigned char dev\_addr,unsigned char reg\_addr, unsigned char data\_len, unsigned char\* wData)

{

unsigned char i = 0;

unsigned char dev\_addr\_w = dev\_addr & 0xFE; //防止读写IIC地址出错，写统一地址即可

IIC\_Start(num); //IIC开始

IIC\_Send\_Byte(num, dev\_addr\_w); //先发设备地址

if(IIC\_Wait\_Ack(num) == 0xff) //没有等到Ack返回则返回0xFF

{ return 0xff; }

IIC\_Send\_Byte(num, reg\_addr); //开始发送寄存器地址

if(IIC\_Wait\_Ack(num) == 0xff)

{ return 0xff; }

if(data\_len == 1) //单次写入

{

IIC\_Send\_Byte(num, wData[data\_len-1]);

if(IIC\_Wait\_Ack(num) == 0xff)

{ return 0xff; }

}

else

{

for(i=0; i<data\_len; i++) //连续写入

{

IIC\_Send\_Byte(num, wData[i]);

if(IIC\_Wait\_Ack(num) == 0xff)

{ return 0xff; }

}

}

IIC\_Stop(num);

return 0x00;

}

void NSA2302\_CalData(void)

{

while((Drdy & 0x01) == 0x01 )

{

IIC\_Read\_Len\_Byte(4 ,0xFE, 0x02, 1, &Drdy);

delay\_us(4);

}

IIC\_Read\_Len\_Byte(4, 0xFE, 0x06, 5, ID\_Num\_R);

Cal\_PData = ID\_Num\_R[0] \*65535 + ID\_Num\_R[1] \*256 + ID\_Num\_R[2];

Cal\_TData = ID\_Num\_R[3] \*65535 + ID\_Num\_R[4] \*256;

if(Cal\_PData > 8388608) { Cal\_PData = (Cal\_PData - 16777216)/ 8388608; }

else { Cal\_PData = Cal\_PData / 8388608; }

if(Cal\_TData > 8388608) { Cal\_TData = (Cal\_TData - 16777216)/ 65535; }

else { Cal\_TData = Cal\_TData / 65535; }

printf("0x06：%d 0x07：%d 0x08：%d \r\n ", ID\_Num\_R[0], ID\_Num\_R[1], ID\_Num\_R[2]);

printf("0x09：%d 0x0A：%d \r\n ", ID\_Num\_R[3], ID\_Num\_R[4]);

printf("压力数字量：%f 温度值：%f \r\n ", Cal\_PData, Cal\_TData);

printf("第一个数据：0x%X 第二个数据：0x%X 第三个数据：0x%X \r\n ", ID\_Num\_R[0], ID\_Num\_R[1], ID\_Num\_R[2]);

printf("第四个数据：0x%X 第五个数据：0x%X 第六个数据：0x%X \r\n ", ID\_Num\_R[3], ID\_Num\_R[4], ID\_Num\_R[5]);

}

void Get\_REG\_Value\_IIC (uint8\_t research\_register)

{

uint8\_t receive\_data =0;

IIC\_Read\_Len\_Byte(10, 0xDA, research\_register, 1, &receive\_data);

printf("寄存器 %#x 值为：%#x\n \r\n ", research\_register , receive\_data);

delay\_ms(1000);

}