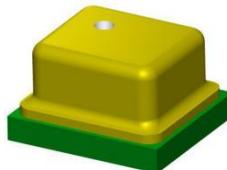


产品规格书

SVP-3729D-152A 压力传感器模块



1、产品描述

SVP-3729D-152A压力传感器模块是由本公司自主研发生产的扩散硅压阻式压力传感器芯片和专用信号处理集成电路芯片构成,每个模块都经过线性校验和温度补偿,使其电压输出信号线性正比于输入压力。

SVP-3729D-152A压力传感器模块设计灵活,高精度和低电流消耗的小型化的数字式气压传感器,兼具压力和温度测量两种特点。小型封装结构适合移动应用和可穿戴设备等。内部信号处理器将压力和温度传感器元件的输出转换为24位数据。每个压力传感器已被单独校准并包含校准系数。在应用中使用系数将测量结果转换成真实的压力和温度值。芯片表面采用中性软性透明硅胶覆盖保护,可有效防潮湿,被广泛应用于穿戴式设备、充气泵、多功能气压表、汽车电子等领域。

2、产品特点

- 压力范围: 0~1750kPa
- 工作电压: 1.8~5.5V
- 工作电流: 5 μ A@1Hz ODR
- 待机电流: <0.2 μ A@25 $^{\circ}$ C
- 数字I²C接口, 24bit ADC
- 工作温度范围: -30 $^{\circ}$ C ~ 105 $^{\circ}$ C
- 压力类型: 绝压型
- 适用于无腐蚀的气体
- 可靠性高、稳定性好、长期漂移低

3、产品应用

- 穿戴式设备
- 充气泵、胎压计
- 气动开关
- 高度计、多功能气压计
- 绝压控制系统

4、规格参数

(VDD=3.3±0.03VDC, T=25°C)

参数	最小值	典型值	最大值	单位	备注
供电电压	1.8	3.3	5.5	V	
工作量程	0		1750	kPa	
待机电流		0.1		uA	
综合精度		±0.5		%F.S	@25°C
		±1.0		%F.S	@0~60°C
内置温度传感器准确度		±1		°C	@-40~85°C
响应时间		2.5		ms	@OSR_P=1024X
ESD(HBM)		4		kV	
压力过载			2X	F.S	
补偿温度	0		60	°C	
工作温度	-30		105	°C	
存储温度	-40		125	°C	
时钟频率			400	kHz	

注：产品在 3.3V±0.3 电压条件下校准标定，超过范围将会影响产品输出精

5、I²C通讯协议

I²C 总线使用 SCL 和 SDA 作为信号线。这两根线都通过上拉电阻连接到 VDD，不通信时都保持为高电平。

I²C 设备地址为 0X6D。

表 5.1 I²C 地址

A7	A6	A5	A4	A3	A2	A1	W/R
1	1	0	1	1	0	1	0/1

表 5.2 I²C 通讯引脚的电气特性

标示	参数	条件	最小值	最大值	单位
F _{scl}	时钟频率			400	kHz
t _{BLOW}	时钟低脉冲维持时间		1.3		us
t _{BHIGH}	时钟高脉冲维持时间		0.6		us
t _{BSUDAT}	SDA 建立时间		0.1		us
t _{BHDDAT}	SDA 保持时间		0.0		us
t _{BSUSTA}	每次开始时的建立时间		0.6		us
t _{BHDSTA}	开始条件保持时间		0.6		us
t _{BSUSTO}	停止时间建立时间		0.6		us
t _{BBUF}	两次通讯之间间隔时间		1.3		us

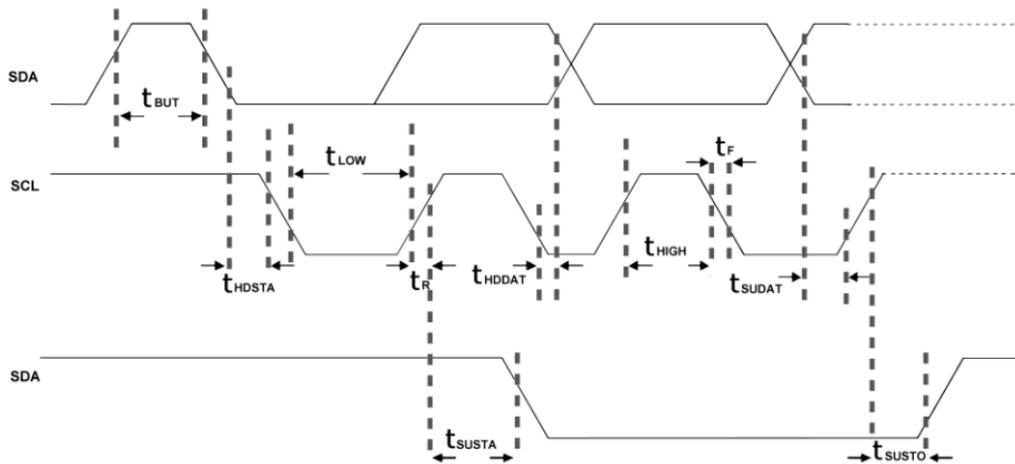


图5.1 I²C时序图

I²C 通讯协议有着特殊的开始(S)和终止(P)条件。当 SCL 处于高电平同时，SDA 的下降沿标志数据传输开始。I²C 主设备依次发送从设备的

地址（7 位）和读/写控制位。当从设备识别到这个地址后，产生一个应答信号并在第九个周期将 SDA 拉低。得到从设备应答后，主设备继续发送 8 位寄存器地址，得到应答后继续发送或读取数据。SCL 处于高电平，SDA 发生一个上升沿动作标志 I²C 通信结束。除了开始和结束标志之外，当 SCL 为高时 SDA 传输的数据必须保持稳定。当 SCL 为低时 SDA 传输的值可以改变。I²C 通信中的所有数据传输以 8 位为基本单位，每 8 位数据传输之后需要一位应答信号以保持继续传输。

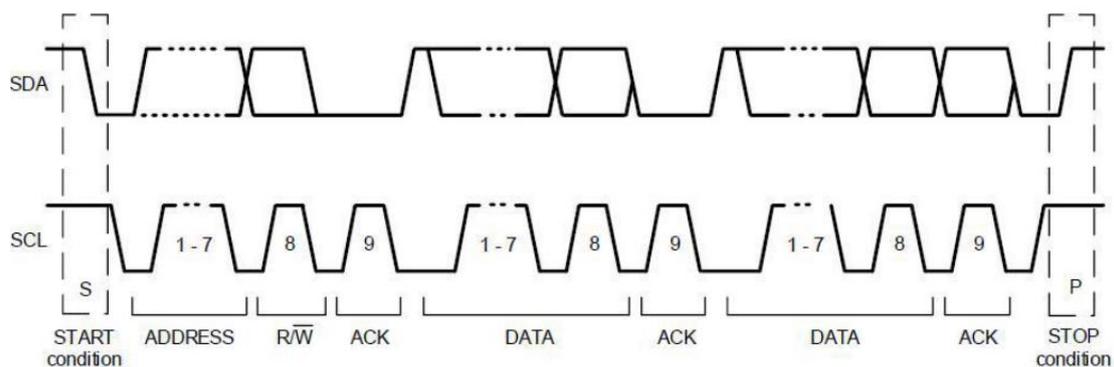


图 5.2 I²C 协议

I²C 数据输出寄存器定义:

地址	描述	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
0x06	DATA_MSB	R	Pdata <23:16>								0x00
0x07	DATA_CSB	R	Pdata <15:8>								0x00
0x08	DATA_LSB	R	Pdata <7:0>								0x00
0x09	TEMP_MSB	R	Tdata <15:8>								0x00
0x0A	TEMP_LSB	R	Tdata <7:0>								0x00
0x30	CMD	RW	Sleep_time<3:0>				Sc0	Measurement_ctrl<2:0>			0x00

Reg0x30 (测量命令)

Sleep_time<3:0>: 0000:0ms, 0001:62.5ms, 0010:125ms ... 1111: 1s,

仅在休眠工作模式下有效。

Measurement_control<1:0>:

000b, 单次温度采集模式。

001b, 单次传感器信号采集模式。

010b: 组合采集模式 (一次温度采集后立即进行一次传感器信号采集)。

011b: 休眠工作模式 (定期的执行一次组合采集模式, 间隔时间由 'sleep_time' 决定)。

Sc0: 1, 开始数据采集, 采集结束时自动回 0 (休眠工作模式除外)。

读取数据按照如下指令顺序进行操作:

- 1、 读取 0xA5 寄存器值, 将读取到的二进制值 “与” 上 “11111101” 后再写入 0xA5。
- 2、 发送指令 0x0A 到 0x30 寄存器进行一次温度采集, 一次压力数据采集。
- 3、 读取 0x30 寄存器地址, 若 Sco 位为 0 代表采集结束, 可以读取数据。或等待延迟 10ms。
- 4、 读取 0x06、0x07、0x08 三个寄存器地址数据构成 24 位 AD 值 (压力数据 AD 值)。

其中 REG0x06、REG0x07、REG0x08 寄存器内为压力输出值共计 24 位, 其中最高位为符号位, 符号位数值为 “1” 时代表 “负”, 符号位数值为 “0” 时代表 “正”。

REG0x09、REG0x0A 寄存器内为温度输出数值共计 16 位, 其中最高位为符号位, 符号位数值为 “1” 时代表 “负”, 符号位数值为 “0” 时代表 “正”。

例如假设 REG0x06、REG0x07、REG0x08、REG0x09、REG0x0A 读出的十进制值为 data1、data2、data3、data4、data5。

传感器输出与 ADC 值换算公式如下:

- 1、 压力值 $P_ADC = data1 * 65536 + data2 * 256 + data3$;

若 $P_ADC > 8388607$, 则 $P_out = P_ADC - 16777215 / 2^{23} * Fullscale$ (Pa);

若 $P_ADC \leq 8388607$, 则 $P_out = P_ADC / 2^{23} * Fullscale$ (Pa);

注意 Fullscale 的取值:

eg: 当满量程压力为 600000Pa 时, $2^{19} < 600000 < 2^{20}$, 在 24 位 AD 中最少需要 20 位来表示, 即此时 Fullscal 取 2^{20} 。

- 2、 温度值 $T_ADC = data4 * 256 + data5$;

若 $T_ADC > 32768$, 则 $T_out = T_ADC - 65536 / 256$ (°C);

若 $T_ADC \leq 32768$, 则 $T_out = T_ADC / 256$ (°C);

6、典型应用

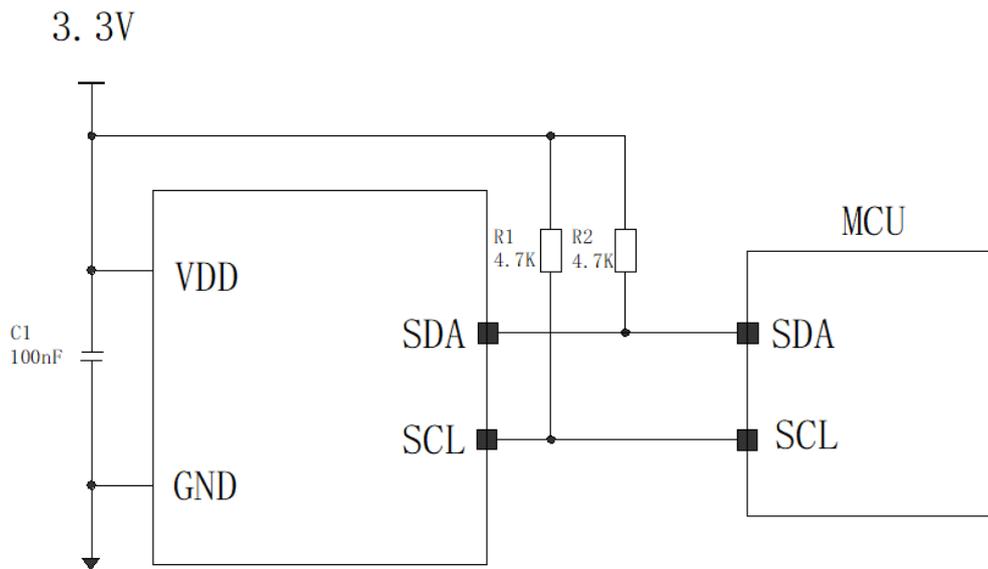
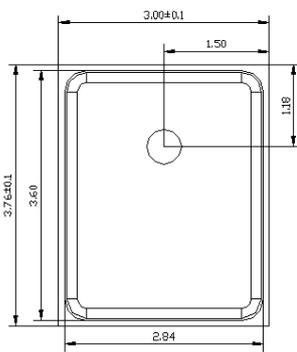


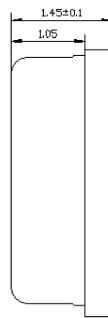
图 6.1 典型应用

7.外形结构

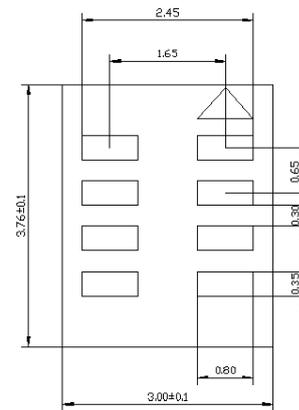
(单位: mm)



Top view

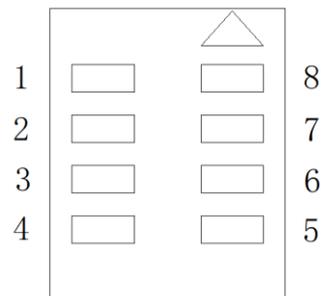


Side view



Bottom view

引脚配置

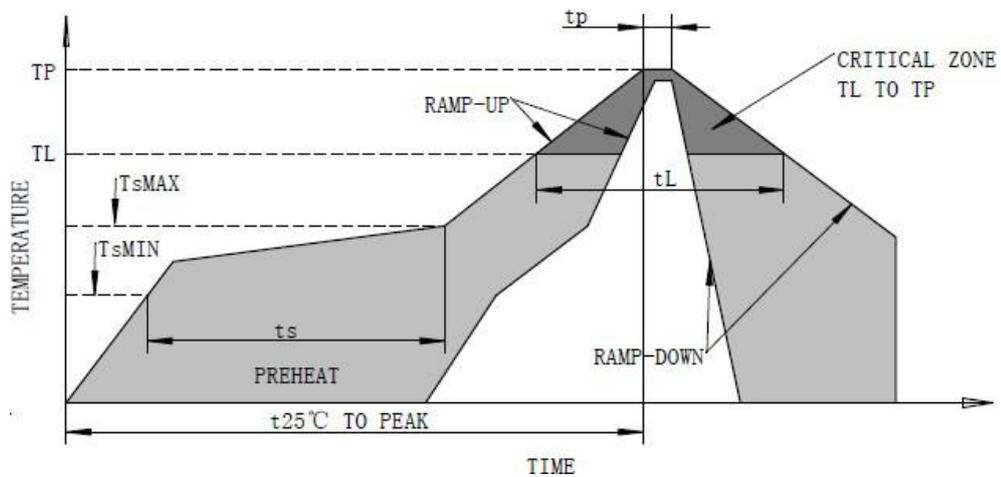


(Bottom view)

引脚	1	2	3	4	5	6	7	8
定义	GND	NC	SDA	SCL	NC	NC	GND	VDD

8、焊接推荐

推荐回流焊接参数



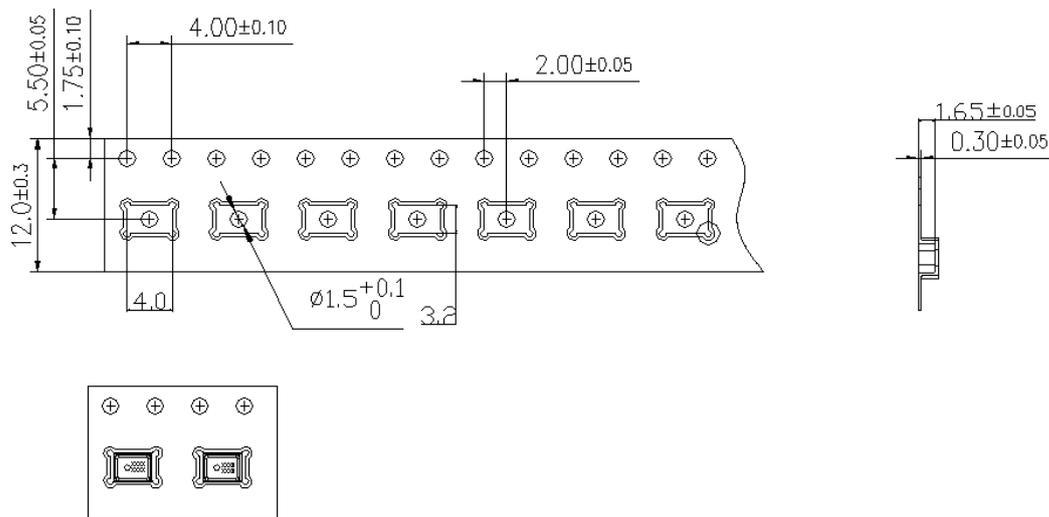
参数	数值
平均升温率	最大 3°C/秒
预热	
-温度最小 (TsMIN)	150°C
-温度最大值 (Tsmax)	200°C
-时间 (TsMIN 到 TsMAX) (Ts)	60~80 秒
以上时间保持:	
-温度 (TL)	217°C
-时间 (tL)	60~150 秒
峰值温度 (TP)	260°C
在实际峰值温度 (TP) 5°C 内的时间 (tP)	20~40 秒
降温率	最大 4°C/秒
25°C 至峰值温度时间	最大 8 分钟

9、包装规格

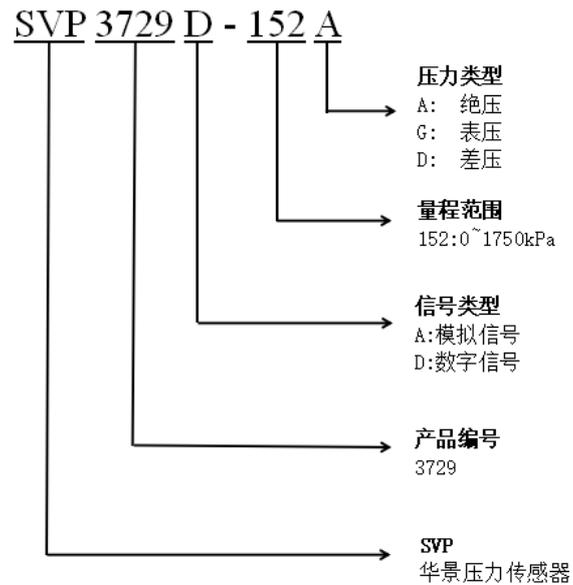
载带信息

单位: mm

每卷数量: 5K PCS



10、订购型号



11、更改版本

版本号	变更内容	变更日期
1.0	新建	2021-9-03

12、联系方式

华景传感科技（无锡）有限公司

地址：江苏省无锡市新吴区菱湖大道200号F2栋，213135

电话：(86) 0510-85622282

传真：(86) 0510-85622278

邮箱：sales@svsens.com

网站：www.svsens.com

附件：IIC 参考例程

```
#include"SH79.h"

#include"8011.h"

/*定义通讯接口*/

sbit 8011_SDA=P1^6;          //通信数据线

sbit 8011_SCL=P1^7;        //通信时钟线

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

// 程序名称：1us 延时程序

// 程序功能：用于调节 IIC 的通讯速度

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

void delay_1us(unsigned char n)

{

    unsigned char i,j;

    for(i=n;i>0;i--)

    {

        for(j=0;j<5;j++);

    }

}

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

// 程序名称：检测 ACK 应答信号函数
```

// 程序功能：用于检测压力传感器返回的应答信号

//

void 8011_Wait_Ack()

{

8011_SCL = 1;

delay_1us(1);

8011_SCL = 0;

delay_1us(1);

}

//

// 程序名称：写 ACK 应答信号

// 程序功能：用于读取压力传感器的一个字节数据

//

void 8011_Write_Ack()

{

8011_SDA = 0;

delay_1us(1);

8011_SCL = 1;

delay_1us(1);

8011_SCL = 0;

delay_1us(1);

8011_SDA = 1;

}

```
////////////////////////////////////
```

```
// 程序名称: IIC 开始时序函数
```

```
// 程序功能: 用于启动压力传感器的 IIC 接口
```

```
////////////////////////////////////
```

```
void 8011_Start()
```

```
{
```

```
    8011_SCL = 1;
```

```
    8011_SDA = 1;
```

```
    delay_1us(1);
```

```
    8011_SDA = 0;
```

```
    delay_1us(1);
```

```
    8011_SCL = 0;
```

```
    delay_1us(1);
```

```
}
```

```
////////////////////////////////////
```

```
// 程序名称: IIC 停止时序函数
```

```
// 程序功能: 用于关闭压力传感器的 IIC 接口
```

```
////////////////////////////////////
```

```
void 8011_Stop()
```

```
{
```

```
    delay_1us(1);
```

```
8011_SCL = 1;

8011_SDA = 0;

delay_1us(1);

delay_1us(1);

8011_SDA = 1;

delay_1us(1);

}

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

// 程序名称: 写单字节函数

// 程序功能: 用于给压力传感器写参数

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

void 8011_WriteChar(unsigned char byte)

{

    unsigned char i,temp;

    8011_SCL = 0;

    temp = byte;

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

    {

        8011_SDA = (temp&0x80)>>7;

        8011_SCL = 1;

        delay_1us(1);

        8011_SCL = 0;

        delay_1us(1);

        temp = temp<<1;

    }

}
```

```
    }  
  
}  
  
////////////////////////////////////  
// 程序名称: 读字节函数  
// 程序功能: 用于读取压力传感器的一个字节数据  
////////////////////////////////////  
  
unsigned char 8011_ReadChar(void)  
{  
    unsigned char i,value;  
    delay_1us(1);  
    /* 读到第 1 个 bit 为数据的 bit7 */  
    for (i = 0; i < 8; i++)  
    {  
        value <<= 1;  
        8011_SCL = 1;  
        delay_1us(1);  
        if (8011_SDA == 1)  
        {  
            value++;  
        }  
        delay_1us(1);  
        8011_SCL = 0;  
        delay_1us(1);  
    }  
}
```

```
    }  
  
    return value;  
}  
  
////////////////////////////////////  
// 程序名称: 读某个地址的数据函数  
// 程序功能: 用于读取压力传感器某个地址的数据  
////////////////////////////////////  
unsigned char 8011_Read(unsigned char reg_addr)  
{  
    unsigned char temp;  
  
    8011_Start();                //开始传输  
  
    8011_WriteChar(0xDA);       //设备地址+写  
  
    8011_Wait_Ack();           //接收应答信号  
  
    8011_WriteChar(reg_addr); //发送寄存器地址  
  
    8011_Wait_Ack();           //接收应答信号  
  
    delay_1us(1);  
  
    8011_Start();                //开始传输  
  
    8011_WriteChar(0xDA+1);    //设备地址+读  
  
    temp=8011_ReadChar();      //读取寄存器数据值  
  
    8011_Write_Ack();          //发送应答信号  
  
    delay_1us(1);  
  
    8011_Stop();                //结束传输  
  
    return(temp);  
}
```

```
////////////////////////////////////  
// 程序名称: 写某个地址的参数函数  
// 程序功能: 用于写压力传感器的某个地址的参数  
////////////////////////////////////  
void 8011_Write(unsigned char reg_addr,unsigned char reg_data)  
{  
    8011_Start();                //开始传输  
    8011_WriteChar(0xDA);        //设备地址+写  
    8011_Wait_Ack();             //接收应答信号  
    8011_WriteChar(reg_addr);    //发送寄存器地址  
    8011_Wait_Ack();             //接收应答信号  
    delay_1us(1);  
    8011_WriteChar(reg_data);    //发送寄存器数据值  
    8011_Wait_Ack();             //接收应答信号  
    delay_1us(1);  
    8011_Stop();                 //结束传输  
}  
  
while(1)  
{  
  
////////////////////////////////////
```

```
                                // 程序名称: 启动测量命令

// 程序功能: 用于启动压力传感器的测量命令

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

    da = 8011_Read(0xA5);

    da &= 0xFD;

    8011_Write(0xA5,da);

    8011_Write(0x30,0x0A);                //启动测量命令

    Delayms(500);

    while((8011_Read(0x30))&0x08); //等待数据转换完成

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

// 程序名称: 读取温度和压力值

// 程序功能: 用于读取压力传感器的温度和压力值

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

    PressH = 8011_Read(0x06);            //读取压力高 8 位

    PressM = 8011_Read(0x07);            //读取压力中 8 位

    PressL = 8011_Read(0x08);            //读取压力低 8 位

    TempH = 8011_Read(0x09);             //读取温度高 8 位

    TempL = 8011_Read(0x0A);             //读取温度低 8 位

}
```