



GZP6826D

型压力传感器

数字输出
无铅产品

产品规格书

版本号： V1.4

文件发行日期： 2022.03.16



目录

1.产品特点	4
2.应用领域	4
3.概述	4
4.性能指标	4
5.电气特性	5
6.外形结构（单位为毫米）	6
7.电气连接	6
8.I ² C 通讯协议	7
9.寄存器描述	8
10.工作模式说明：	10
11.选型指南	11
12.常用量程	11
13.选型提示	11
14.使用注意事项	11
14.1.焊接	11
14.2.清洗要求	12
14.3.存储和运输	12
14.4.其他使用注意事项	13
15.包装信息	14
安全注意事项	15
IIC Example Code（附件：IIC 代码案例）	错误！未定义书签。
免责声明	22



● 文件修订历史

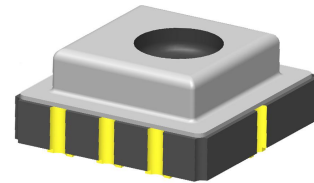
修订	描述	日期
V1.0	初始版本	2021.02.02
V1.1	变更金属盖板	2021.03.29
V1.2	修改金属盖板	2021.04.30
V1.3	增加封面目录	2021.10.18
V1.4	调整产品归类	2022.03.16

公司保留在不另行通知的情况下对其所包含的规格进行更改的权利。
产品规格书版权及产品最终解释权归芯感智所有。



1.产品特点

- 测量范围 0kPa ~ 100kPa...2500kPa
- 绝压型
- 电源电压: 3.3V ~ 5.5V
- IIC 通讯
- 陶瓷封装结构, 防潮防油
- 压力量程可定制



2.应用领域

- 汽车、摩托车进气系统以及胎压计、车载气泵等汽车电子领域
- 打气泵、真空泵、压力开关、气动控制系统和工业设备等
- 消防灭火器、气象站、导航等领域
- 其它绝压测量系统

3.概述

GZP6826D 型压力传感器采用陶瓷板作为基底材料, 内有封装的压力传感器与信号调理芯片, 对传感器的偏移、灵敏度、温漂和非线性进行数字补偿。采用 21 位 ADC, 并且调理芯片内置温度传感器, 可以输出高精度的压力值和温度值。同时提供 IIC 通讯协议接口, 抗干扰能力强。所有测量数据都经过充分校准和温度补偿。

GZP6826D 型压力传感器内部芯片和金线上覆有含氟硅胶, 可防水防潮、耐油气。

GZP6826D 型压力传感器响应快、精度高、功耗低, 具有良好的线性以及稳定性。尺寸小、易安装, 便于系统集成, 广泛应用于消费电子、汽车电子、消防、家电和其他民用气象等领域。

4.性能指标

供电电源: $(5 \pm 0.25)V$ DC

参考温度: 25℃



表 1.性能指标

项目	数值	单位
精度*	±1	%Span
响应时间	2.5ms@OSR_P=1024X	ms
SDA/SCL 上拉电阻	2.2	K ohm
ESD HBM	4000	V
零点温度漂移	±0.03	%FS/°C
满程温度漂移	±0.03	%FS/°C
过载压力	2× (量程 ≤350kPa) 1.5× (量程 >350kPa)	Rated
破坏压力	3× (量程 ≤350kPa) 2× (量程 >350kPa)	
补偿温度	0 ~ 60 (可定制)	°C
工作温度	-30 ~ 125	°C
贮存温度	-40 ~ 150	°C

* 精度为 0 ~ 70°C 范围内的输出误差，由压力的线性、重复性、迟滞组成，其压力量程不同，精度不同，请咨询客服获取更多细节。

5.电气特性

表 2.电气特性

参数	最小值	典型值	最大值	单位	备注
供电电压	3.3		5.5	V	
待机电流		100		nA	
LDO 输出		3.0		V	
PSRR		60		dB	
输出数据分辨率		21		Bits	
输入共模信号抑制比	80	110			
内置温度传感器 准确度			±0.5	°C	@0~60°C
			±1	°C	非校准温区
输出数据解析度	16			Bit	LSB = (1/256) °C
时钟脉冲频率			400	KHz	I2C 通讯

6.外形结构 (单位为毫米)

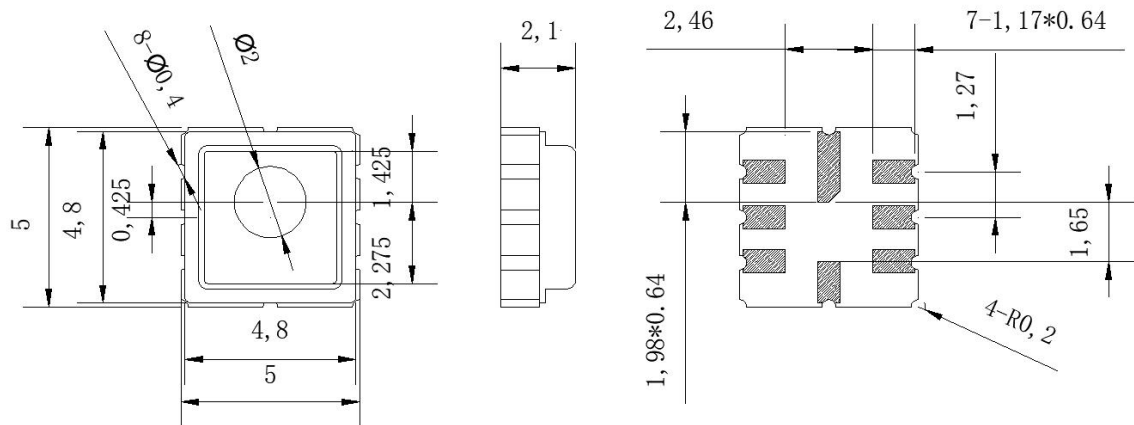


图 1. 外形结构

7.电气连接

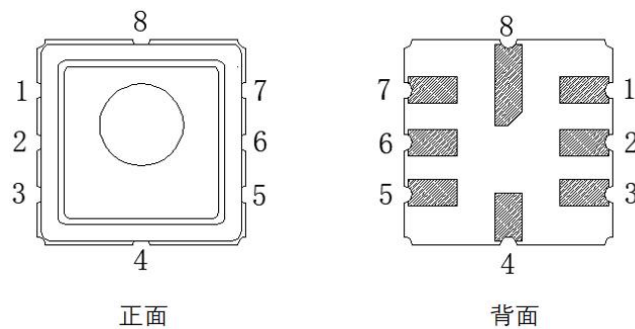


表 3. 引脚对应关系

1	2	3	6	4、5、7、8
GND	SCL	SDA	VDD	NC

注意:

1. 装配前请确认好电气定义
2. NC 脚不要有任何的电气连接，否则可能会造成产品功能失效
3. 焊装过程中做好防静电保护
4. 过载电压(6.5Vdc)可能烧毁电路芯片
5. 请在 VDD 和 GND 之间加上 0.1uf 电容
6. 本产品无反接保护，装配时请注意电源极性

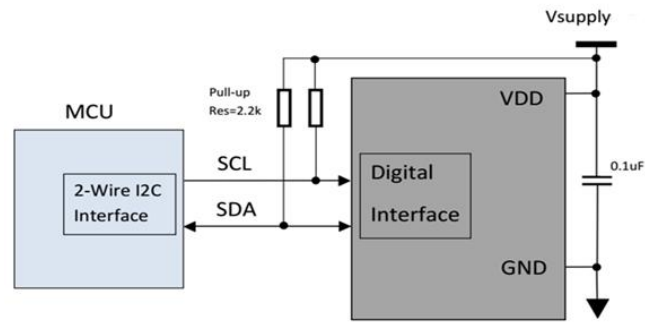


图 2.典型应用

8.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（典型值 2.2K）连接到 VDD，不通信时都保持为高电平。I2C 设备地址为 0x58。

■ I²C 通讯引脚的电性特性

表 4. I²C 通讯引脚的电性特性

标示	参数	条件	最小值	最大值	单位
f_{scl}	时钟频率			400	KHz
t_{LOW}	时钟低脉冲维持时间		1.3		US
t_{HIGH}	时钟高脉冲维持时间		0.6		US
t_{SUDAT}	SDA 建立时间		0.1		US
t_{HDDAT}	SDA 保持时间		0.0		US
t_{SUSTA}	每次开始时的建立时间		0.6		US
t_{HDSTA}	开始条件保持时间		0.6		US
t_{SUSTO}	停止条件建立时间		0.6		US
t_{BUF}	两次通讯间隔时间		1.3		US

■ I²C 时序图

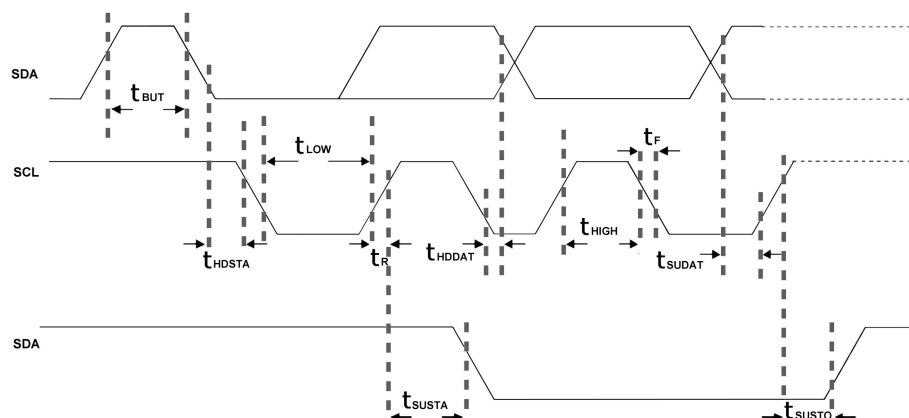


图 3. I²C 时序图



I²C 通讯协议有特殊的开始(S)和终止(P)条件。当 SCL 处于高电平时，SDA 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。

■ I²C 协议

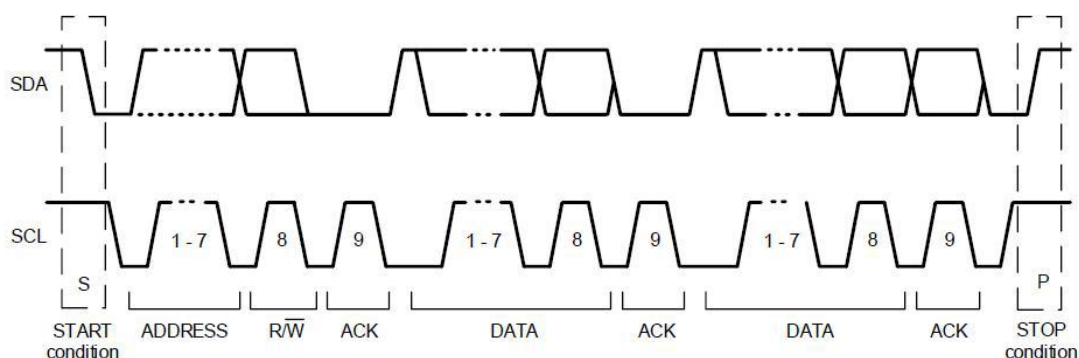


图 4. I²C 协议

9. 寄存器描述

表 5. 寄存器描述

地址	描述	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值	
0x00	ID	R	ID<7:0>								0x58	
0x01	Chip_Contr ol	R/W	reserved<7 :6>	data_ready		reser ved	data_o ut	measur ent_ctrl	Active<1:0>		OTP	
0x02	CFG_OSR	R/W	OSR_T<7:5>			OSR_P<4:2>			MODE[1:0]		OTP	
0x03	CFG_MEAS	R/W	reserved<7 :6>	T_SB[5:3]				PT_R[2:0]			OTP	
0x04	P_data	R	Data out<23:16>								0x00	
0x05	P_data	R	Data out<15:8>								0x00	
0x06	P_data	R	Data out<7:0>								0x00	
0x07	T_data	R	Temp out<15:8>								0x00	
0x08	T_data	R	Temp out<7:0>								0x00	
0x24	CFG_OPER	R/W	reserved<7:1>							DAC_EN		OTP



Reg0x00

I²C 设备地址，默认地址为 0x58。

Reg0x01

芯片控制寄存器

Active<1:0>:00,芯片掉电; 01, 芯片启动;

measurement_ctrl: 0, 压力测量; 1, 温度测量;

data_out: 0, 输出校准数据; 1, 输出原始数据;

data_ready: 0, 数据转换未完成; 1, 数据转换完成。

Reg0x02

MODE[1:0]: 00: Sleep mode 睡眠模式, 01: Normal mode 正常模式, 10: One shot mode 单次采集模式, 11: Normal mode, cyclic measurement 正常模式循环测量

OSR_P[4:2] (压力过采样) : 000: over sampling x 256

001: over sampling x 512

010: over sampling x 1024

011: over sampling x 2048

100: over sampling x 4096

101: over sampling x 8192

110: over sampling x 16384

111: over sampling x 32768

OSR_T[7:5] (温度过采样) : 000: over sampling x 256

001: over sampling x 512

010: over sampling x 1024

011: over sampling x 2048

100: over sampling x 4096

101: over sampling x 8192

110: over sampling x 16384

111: over sampling x 32768

Reg0x03

PT_R[2:0]: 000: 64/1, 001: 32/1, 010: 16/1, 011: 8/1, 100: 4/1, 101: 1/1, Others: 128/1 (正常模式下, 压力/温度测量比)

T_SB[5:3]: 000: 0ms, 001: 62.5ms, 010: 125ms, 011: 250ms, 100: 500ms, 101: 750ms, 110: 1000ms, 111: 2000ms (正常模式下的待机时间设置)



Reg0x04-Reg0x06

压力数据寄存器

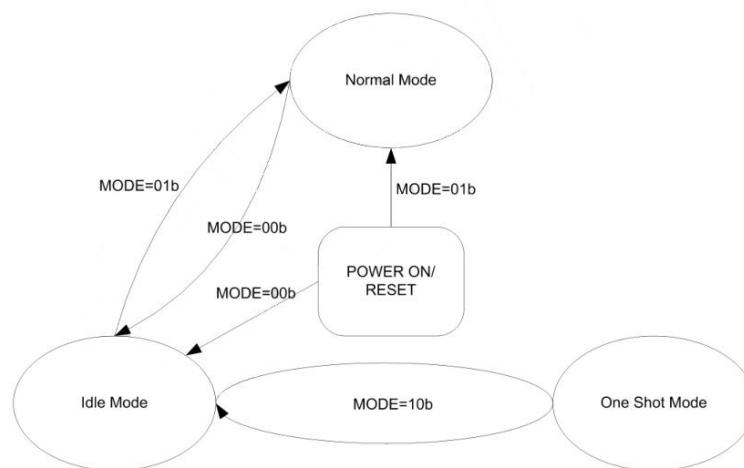
Reg0x07-Reg0x08

温度数据寄存器

Reg0x24

DAC_EN: 0:禁用 DAC, 1:启用 DAC

10.工作模式说明:



正常(Normal Mode)模式，读取数据按照如下指令顺序进行操作：

- 1) VDD 上电
- 2) 先往 0x01 地址里写 0x01，启动芯片
- 3) 延时 20ms 后续就可以从 0x04 连续读 5 个 bytes
- 4) 前 3 个 bytes 为气压数据，需要算补码，具体为：

$sum = (0x04 \text{ 值} * 2^{16} + 0x05 \text{ 值} * 2^8 + 0x06 \text{ 值})$ ，以 30~120kpa 量程为例：

若 $sum < 8388608$ ，则 $P = sum / 2^{21} * 90 + 30$ （单位为 kpa）

若 $sum \geq 8388608$ ， $P = (sum - 16777216) / 2^{21} * 90 + 30$ （单位为 Kpa）

其中，30 为压力量程零点标定压力值。



11.选型指南

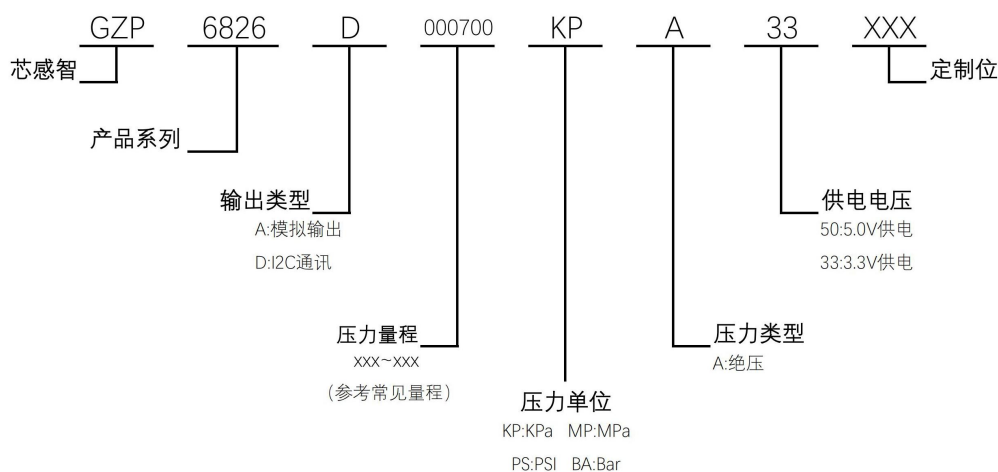


图 5.选型指南

12.常用量程

表 7. 常用量程表(以 3.3V 供电为例)

压力量程 (kPa)	型号
0 ~ 100	GZP6826D000100KPA33
0 ~ 350	GZP6826D000350KPA33
0 ~ 700	GZP6826D000700KPA33
0 ~ 1000	GZP6826D0001.0MPA33
0 ~ 2500	GZP6826D0002.5MPA33
15-115	GZP6826D015115KPA33
20-250	GZP6826D020250KPA33
50-400	GZP6826D050400KPA33
75-325	GZP6826D075325KPA33
更多定制量程及参数,请咨询我司客服或代理商	

13.选型提示

- 1.选型时请注意被测介质要与产品与介质相接触的部分相兼容。
- 2.若对产品的性能参数和功能上有特殊要求, 请与本公司商洽。

14.使用注意事项

14.1.焊接

由于本产品为热容量较小的小型构造，因此请尽量减少来自外部的热量的影响。否则可能会因热变形而造成破损，引起特性变动。请使用非腐蚀性的松香型助焊剂。另外，由于产品暴露在外，因此请注意不要使助焊剂侵入内部。

1) 手焊接

- 请使用头部温度在 260 ~ 300 °C (30 W) 的电烙铁 在 5 秒以内实施作业。
- 在端子上施加负载进行焊接的情况下，由于输出可能会 发生变化，因此请注意。
- 请充分清洗电烙铁头。

2) 回流焊接 (SMD 端子型)

推荐的回流炉温度设置条件如下所示

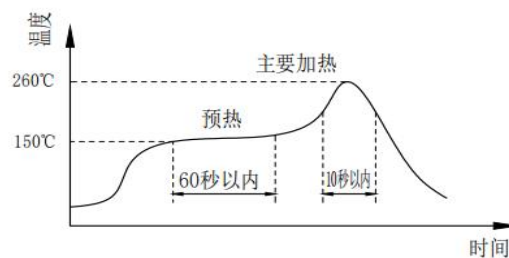


图 6. 回流温度设置条件

- 3) 印刷板的翘度相对于整个传感器应保持在 0.05mm 以下，请对此进行管理。
- 4) 安装传感器后，对基板进行切割弯折时，请注意不要使焊接部产生应力。
- 5) 由于传感器的端子为外露构造，因此金属片等触摸端子后，会引发输出异常。请注意不要用金属片或者手等触摸。
- 6) 焊接后，为了防止基板的绝缘恶化而实施涂层时，请注意不要使传感器上面附着药剂。

14.2.清洗要求

- 1) 由于产品为开放型，因此请注意不要整体清洗，并注意清洗所接触的材质。
- 2) 使用超声波进行清洗时，可能会使产品发生故障，因此请避免使用超声波进行清洗。

14.3.存储和运输

- 1) 本产品仅受压端防水防潮，故需防止液体沾到或溅到芯片电气连接端。
- 2) 请勿在产生凝露的环境中使用。另外，附着在传感器芯片上的水分冻结后，可能会造成传感器输出的变动或者破坏。



- 3) 压力传感器的芯片在构造上接触到光后, 输出会发生变动。尤其是通过透明套等施加压力时, 请避免使光接触到传感器的芯片。
- 4) 正常包装的压力传感器可通过普通输送工具运输。请注意: 产品在运输过程中防止潮湿、冲击、晒伤和压力。

14.4.其他使用注意事项

- 1) 安装方法错误时, 会造成事故, 因此请注意。
- 2) 请避免采用超声波等施加高频振动的使用方法。
- 3) 对于腐蚀性的测量介质, 请做腐蚀性验证或请咨询我司后再决定是否使用。
- 4) 压力导入口的内部配置有压力传感器芯片。从压力导入口插入针等异物后, 会造成芯片破损和导入口堵塞, 因此请绝对避免上述操作。
- 5) 关于使用压力, 请在额定压力的范围内使用。在范围外使用时, 会造成破损。
- 6) 由于可能因静电而造成破坏, 因此使用时请注意:
请将桌子上的带电物, 作业人员接地, 以使周围的静电安全放电。
- 7) 根据所使用的压力, 请充分注意产品的固定和套管, 导入管的固定及选择。另外, 如有疑问, 敬请垂询。

■ 请在实际使用状态下进行确认

由于本规格为产品单体规格, 为了提高实际使用时的可靠性, 请确认实际使用状态下的性能和品质。

15.包装信息

载带

每卷数量:3,000 pcs

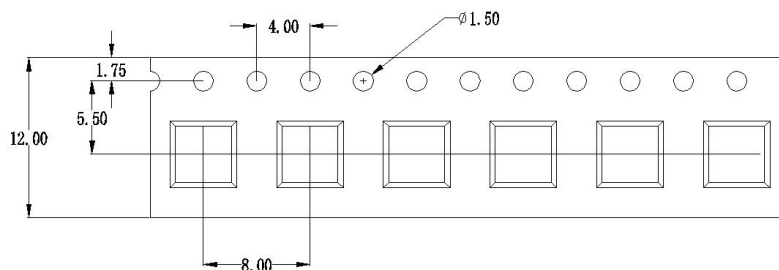


图 7. 载带

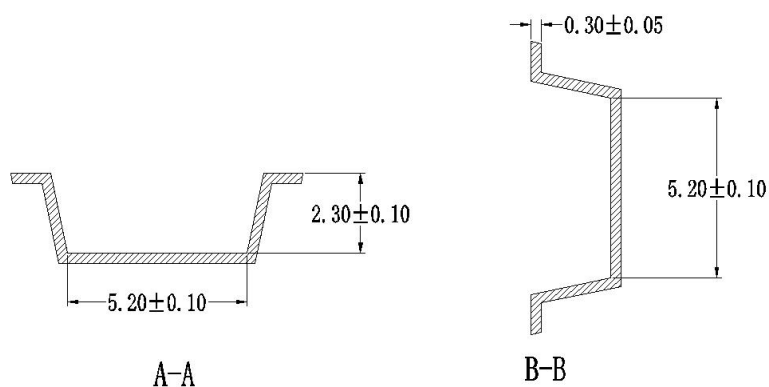


图 8. 载带口袋细节图

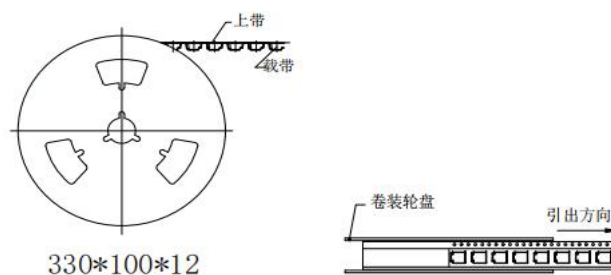


图 9. 载带转盘



安全注意事项

本产品是使用一般电子设备用（通信设备，测量设备，工作机械等）的半导体部品而制成的。使用这些半导体部品的产品，可能会因外来干扰和浪涌而发生误动作和故障，因此请在实际使用状态下确认性能及品质。为以防万一，请在装置上进行安全设计（保险丝，断路器等保护电路的设置，装置多重化等），一旦发生误动作也不会侵害生命，身体，财产等。为防止受伤及事故的发生，请务必遵守以下事项：

- 驱动电流和电压应在额定值以下使用。

- 请按照电气定义进行接线。特别是对电源进行逆连接后，会因发热，冒烟，着火等电路损伤引发事故，因此敬请注意。

- 对产品进行固定和对压力导入口进行连接时请慎重。



IIC Example Code (附件: IIC 代码案例)

```
#include <reg52.h>
#include <math.h>
#define DELAY_TIME 600
#define TRUE 1
#define FALSE 0
#define uchar unsigned char
#define uint unsigned int

//IIC SLAVER address 0x58
//IIC WRITE 0xB0
//IIC READ 0xB1

//-----define IIC SCL,SDA port-----

sbit SCL = P1 ^ 7;
sbit SDA = P1 ^ 6;

//-----delay time_us-----
void DELAY(uint t)
{
    while (t != 0)
        t--;
}

//-----IIC START CONDITION-----
void I2C_Start(void)
{
    SDA = 1;          //SDA output high
    DELAY(DELAY_TIME);
    SCL = 1;
    DELAY(DELAY_TIME); //SCL output high
    SDA = 0;
    DELAY(DELAY_TIME);
    SCL = 0;
    DELAY(DELAY_TIME);
}

//-----IIC STOP CONDITION-----
void I2C_Stop(void)
{

```




```
SDA = 0;          //SDA OUTPUT LOW
DELAY(DELAY_TIME);
SCL = 1;
DELAY(DELAY_TIME);
SDA = 1;
DELAY(DELAY_TIME);
SCL = 0;          //SCL OUTPUT LOW
DELAY(DELAY_TIME);
}

//-----IIC SEND DATA "0"-----
void SEND_0(void)
{
    SDA = 0;
    DELAY(DELAY_TIME);
    SCL = 1;
    DELAY(DELAY_TIME);
    SCL = 0;
    DELAY(DELAY_TIME);
}

//-----IIC SEND DATA "1"-----
void SEND_1(void)
{
    SDA = 1;
    DELAY(DELAY_TIME);
    SCL = 1;
    DELAY(DELAY_TIME);
    SCL = 0;
    DELAY(DELAY_TIME);
}

//-----Check SLAVE's Acknowledge -----
bit Check_Acknowledge(void)
{
    SDA = 1;
    DELAY(DELAY_TIME);
    SCL = 1;
    DELAY(DELAY_TIME / 2);
    FO = SDA;
    DELAY(DELAY_TIME / 2);
    SCL = 0;
```



```
    DELAY(DELAY_TIME);
    if (F0 == 1)
        return FALSE;
    return TRUE;
}

//-----Write One Byte of Data -----
void Writel2CByte(uchar b) reentrant
{

    char i;
    for (i = 0; i < 8; i++)
        if ((b << i) & 0x80)
            SEND_1();
        else
            SEND_0();
}

//-----Read One Byte of Data -----
uchar Readl2CByte(void) reentrant
{

    char b = 0, i;
    for (i = 0; i < 8; i++)
    {
        SDA = 1;
        DELAY(DELAY_TIME);
        SCL = 1;
        DELAY(DELAY_TIME);
        //DELAY(10);
        F0 = SDA;
        DELAY(DELAY_TIME);
        //DELAY(10);
        SCL = 0;
        if (F0 == 1)
        {
            b = b << 1;
            b = b | 0x01;
        }
        else
            b = b << 1;
    }
    return b;
}
```



```
//-----write One Byte of Data,Data from MASTER to the SLAVER
-----
//-----SLAVER address bit:01101101-----
void Write_One_Byte(uchar addr, uchar thedata) //Write "thedata" to the SLAVER's address of
"addr"
{
    bit acktemp = 1;
    I2C_Start(); //IIC START
    Writel2CByte(0xb0); //IIC WRITE operation,SLAVER address 0x58
    acktemp = Check_Acknowledge(); //check the SLAVER
    Writel2CByte(addr); /*address*/
    acktemp = Check_Acknowledge();
    Writel2CByte(thedata); /*thedata*/
    acktemp = Check_Acknowledge();
    I2C_Stop(); //IIC STOP
}

//-----Reaad One Byte of Data,Data from SLAVER to the MASTER
-----
uchar Read_One_Byte(uchar addr)
{
    bit acktemp = 1;
    uchar mydata;
    I2C_Start();
    Writel2CByte(0xb0);
    acktemp = Check_Acknowledge();
    Writel2CByte(addr);
    acktemp = Check_Acknowledge();
    I2C_Start();
    Writel2CByte(0xb1); //IIC READ operation
    acktemp = Check_Acknowledge();
    mydata = Readl2CByte();
    acktemp = Check_Acknowledge();
    I2C_Stop();
    return mydata;
}

//-----Delay_ms -----
void Delay_xms(uint x)
{
    uint i, j;
    for (i = 0; i < x; i++)
        for (j = 0; j < 112; j++)
```



```
        ;
    }

    void main(void)
    {
        uchar yali1, yali2, yali3, wendu1, wendu2, w3, w4;
        //uchar temp_a5;
        int ad, temp, EOFF;
        float k=90000/2097152; //标定区间是 30 到 120kpa, 跨度为 90000pa
        float pas, shift_N, T;
        uchar dis[8];
        Delay_xms(100);
        // temp_a5 = Read_One_Byte(0x00); //Read ID
        Write_One_Byte(0x01, 0x01); //启动采集数据命令
        Delay_xms(20);
        while (1)
        {
            // -----READ ADC output Data of Pressure -----
            yali1 = Read_One_Byte(0x04);
            yali2 = Read_One_Byte(0x05);
            yali3 = Read_One_Byte(0x06);

            ad = yali1 * 65536 + yali2 * 256 + yali3;

            // -----READ ADC output Data of Temperature -----
            wendu1 = Read_One_Byte(0x07);
            wendu2 = Read_One_Byte(0x08);
            temp = wendu1 * 256 + wendu2;
            if (temp > 32768)
                temp = temp - 65536;

            w3 = Read_One_Byte(0x20);
            w4 = Read_One_Byte(0x21);
            if (w3 == 0x0C)
                EOFF = 4096;
            else if (w3 == 0x8C)
                EOFF = -4096;
            else if (w3 == 0x0D)
                EOFF = 8192;
            else if (w3 == 0x8D)
                EOFF = -8192;
            else if (w3 == 0x0E)
                EOFF = 16384;
            else if (w3 == 0x8E)
```



```
EOFF=-16384;
shift_N=pow(2,w4/10);

T=(temp-EOFF)/(shift_N)+25;

/*Conversion, the following is the conversion formula of 120kpa*/

    if (ad > 8388608)
    {
        ad=ad-16777216;
        pas = (float)ad*k+30000; //单位为 pa
    }
    else
        pas = (float)ad*k+30000;    //单位为 pa
    }
}
```



免责声明

本表中的信息已经过仔细审查，并被认为是准确的；但是，不对不准确之处承担任何责任。此外，此信息不会向此类设备的购买者传达制造商专利权下的任何许可。芯感智保留对此处的任何产品进行更改的权利，恕不另行通知。芯感智对其产品对任何特定用途的适用性不作任何保证、陈述或保证，也不承担因应用或使用任何产品或电路而产生的任何责任，并明确否认任何和所有责任，包括但不限于后果性或附带损害。典型参数可以而且确实在不同的应用中有所不同。客户的技术专家必须针对每个客户应用验证所有操作参数。