

XGZP6847D 压力传感器

● 产品特点

- 测量范围 -100kPa ~ 0 ~ 1kPa ~ 1500kPa
- 表压型 (正压或负压)
- 电源电压: 2.5V ~ 5.5V
- 电流消耗: 5uA(一次测量)
- 休眠状态电流: <100nA (25°C)
- 24 位 ADC, 高分辨率
- 温度精度: ±0.5°C@25°C
- 适用于无腐蚀性的气体
- I²C 信号输出, 宽温度补偿



● 应用领域

- 电子血压计、呼吸机、制氧机、监护仪等医疗领域
- 雾化器、喷雾器、负压传感器等消费电子领域
- 按摩器、按摩椅、气垫床等运动健身器材领域
- 热水器、活氧水机、啤酒机、咖啡机、气泵、真空泵、压力仪表等领域

● 概述

XGZP6847D 型压力传感器采用类 DIP 封装形式, PCB 板的两面分别安装有 SOP 封装的压力传感器与信号处理电路芯片, 对传感器的偏移、灵敏度、温漂和非线性进行数字补偿, 以供电电压为参考, 产生一个经过校准、温度补偿后的标准数字信号。

XGZP6847D 型压力传感器尺寸小、易安装, 广泛用于医疗电子、汽车电子、消费电子、运动健身器材、物联网等领域。

● 性能指标

供电电源: (5±0.25) V DC

参考温度: 25°C

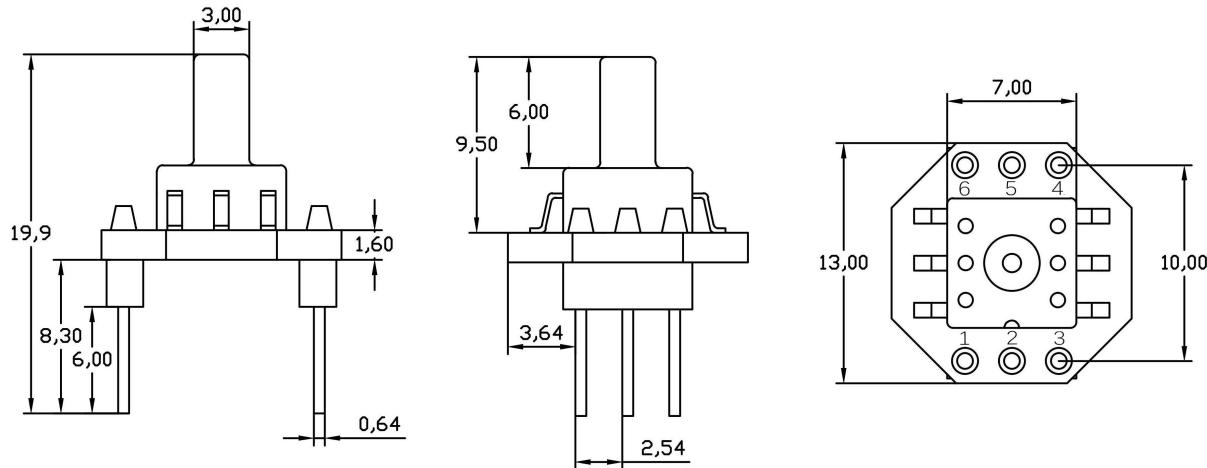
项目	数值	单位
精度 *	±2	%Span
响应时间	2.5ms@OSR_P=1024X; 001B(单次测量命令模式) 5ms@OSR_P=1024X; 010B(组合测量命令模式)	Ms
SDA/SCL 上拉电阻	4.7	Kohm
ESD HBM	4000	V
过载压力	5x (FS≤40kPa) 2x (FS≥100kPa) 1.5x (FS > 200kPa)	Rated
补偿温度	0 ~ 60	°C
工作温度	-30 ~ 100	°C
贮存温度	-30 ~ 125	°C

* 精度即综合误差，由压力的线性、重复性、迟滞的误差组成，其压力量程不同，精度不同。
请咨询客服获取更多细节。

● 电气特性

参数	最小值	典型值	最大值	单位	备注
供电电压	2.5		5.5	V	
待机电流		100		nA	
LDO 输出	1.62	1.8	1.98	V	'regulator_sel' = 0
	3.24	3.6	3.96	V	'regulator_sel' = 1
PSRR		60		dB	
分辨率		24		Bits	
输出数据分辨率		24		Bits	LSB=(1/2^23)*VEXT
输入共模信号抑制比	80	110			
内置温度传感器准确度			±0.5	°C	@25°C
			±1	°C	-40 to 85 °C
输出数据解析度		16		Bit	LSB = (1/256) °C
时钟脉冲频率			400	khz	I2C 通讯

- 外形结构(单位: mm)



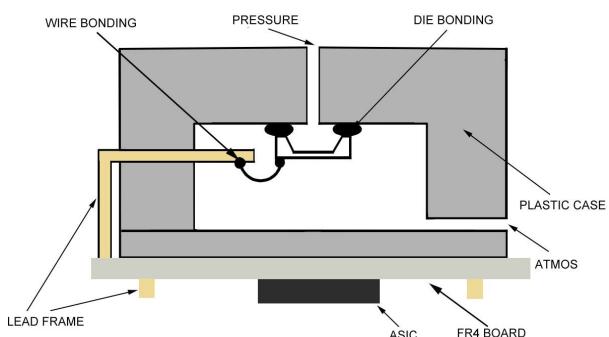
- 电气连接

1	2	3	4	5	6
SCL	NC	GND	VDD	SDA	NC

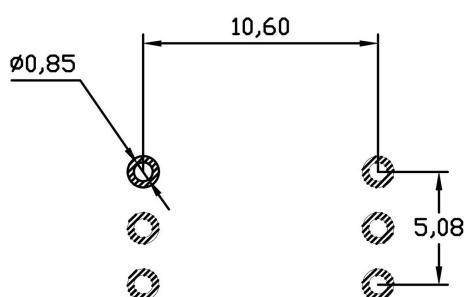
注意:

1. 焊装过程中做好防静电保护
2. 过载电压(6.5Vdc)或电流(5mA)极可能烧毁电路芯片
3. 请在 VDD 和 GND 之间加上 0.1uf 电容

结构示意图



焊盘尺寸图 (mm)



- 选型指南

XGZP6847

D 100

KP

G

压力类型(G:正表压 GN:负压 GPN:正表压+负压)

压力单位(MP: MPa KP: kPa HP:hPa)

压力量程(000-999)

输出信号(A: 模拟(0.5-4.5V) D:I2C)

产品系列

- 量程示例(备注：因少部分型号规格已停产，请在采购前咨询厂商所订购型号规格之量产情况)

压力量程 (kPa)	型号
0 ~ 2.5	XGZP6847D025HPG
0 ~ 5	XGZP6847D005KPG
0 ~ 10	XGZP6847D010KPG
0 ~ 20	XGZP6847D020KPG
0 ~ 40	XGZP6847D040KPG
0 ~ 100	XGZP6847D100KPG
0 ~ 200	XGZP6847D200KPG
0 ~ 700	XGZP6847D700KPG
0 ~ 1000	XGZP6847D001MPG
-100 ~ 0	XGZP6847D100KPGN
-20 ~ 0	XGZP6847D020KPGN
-5 ~ 5	XGZP6847D005KPGPN
-40 ~ 40	XGZP6847D040KPGPN
-100 ~ 100	XGZP6847D100KPGPN
-100 ~ 700	XGZP6847D700KPGPN
-100 ~ 1000	XGZP6847D001MPGPN
可根据需要定制各种量程及参数	

● 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 4.7K）连接到 VDD，不通信时都保持为高电平。I2C 设备地址为 **0x6D**。主设备可使用以下格式的命令与产品通信。



From master to slave



Start



Acknowledge



From slave to master



Stop



Not Acknowledge



Write One Byte To One Register (向寄存器中写 1 个字节)

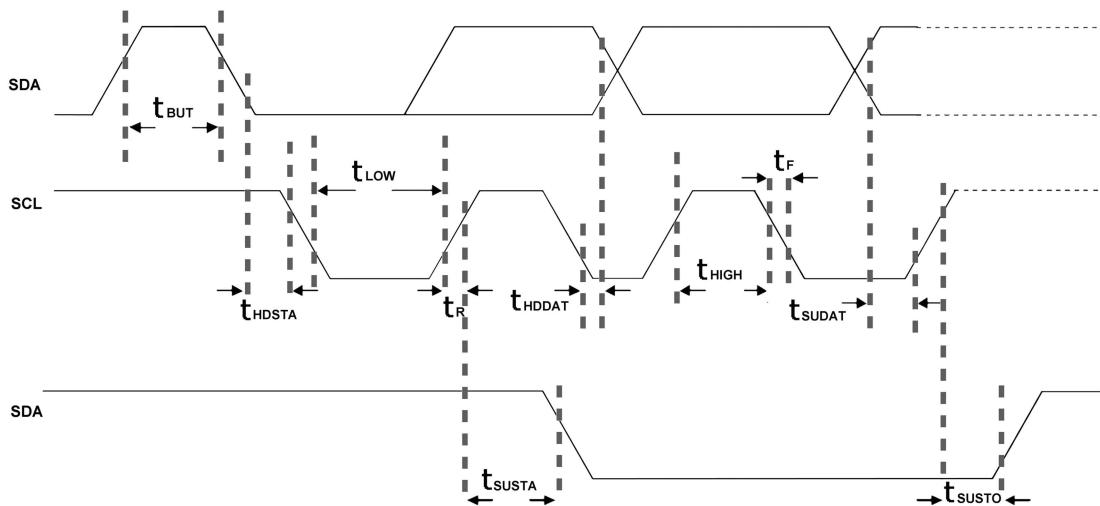


Read One Byte From One Register (从寄存器中读 1 个字节)

■ I²C 通讯引脚的电性特性

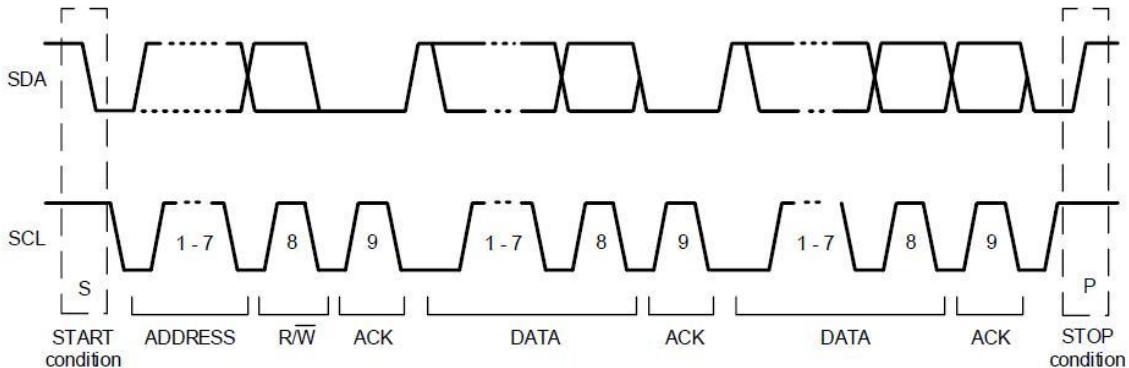
标示	参数	条件	最小值	最大值	单位
f_{scl}	时钟频率			400	kHz
t_{LOW}	时钟低脉冲维持时间		1.3		us
t_{HIGH}	时钟高脉冲维持时间		0.6		us
t_{SUDAT}	SDA 建立时间		0.1		us
t_{HDDAT}	SDA 保持时间		0.0		us
t_{SUSTA}	每次开始时的建立时间		0.6		us
t_{HDSTA}	开始条件保持时间		0.6		us
t_{SUSTO}	停止条件建立时间		0.6		us
t_{BUF}	两次通讯间隔时间		1.3		us

■ I²C 时序图



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。

■ I²C 协议



● 寄存器描述

地址	描述	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
0x06	DATA_MSB	R									0x00
0x07	DATA_CSB	R									0x00
0x08	DATA_LSB	R									0x00
0x09	TEMP_MSB	R									0x00
0x0A	TEMP_LSB	R									0x00
0x30	CMD	RW			Sleep_time<7:4>		sco			Measurement_ctrl<2:0>	0x00
0xA5	Sys_config	RW			Aout_config<7:4>		LDO_config	Unipolar	Data_out_c ontrol	Diag_on	OTP
0xA6	P_config	RW			Input Swap		Gain_P<5:3>			OSR_P<2:0>	OTP

Reg0x06-Reg0x08: 压力数据寄存器

Reg0x09-Reg0x0A: 温度数据寄存器

Reg0x30 (测量命令寄存器)

Measurement_control<2:0>(工作模式)

010b,组合采集模式 (一次温度采集后立即进行一次传感器压力信号采集)。

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)

Sleep_time<7:4>: 0001:62.5ms, 0010:125ms ... 1111: 1s, 0000:无意义。(仅在休眠工作模式下有效)

Sco:数据采集完成标志位。1, 开始数据采集; 0, 采集结束 (休眠工作模式除外)。

Reg0xA5

Aout_config<7:4>:模拟输出配置（建议保留默认配置）

LDO_config: 内部 LDO 配置。0, 配置成 1.8V; 1, 配置成 3.6V

Unipolar: 0, ADC 原始数据以有符号数格式输出; 1: ADC 原始数据以无符号格式输出。(仅当 'Data_out_control'=1 有效)

Data_out_control: 0, 输出校准数据; 1, 输出 ADC 原始数据 (默认配置为 0)

Diag_on: 0, 关闭诊断功能; 1, 开启诊断功能 (默认开启)

Reg0xA6

Input Swap: 在传感器内部交换差分信号极性。

Gain_P<5:3>:采集传感器信号时 PGA 增益, 000:增益=1X。001:增益=2X。010:增益=4X。011:增益=8X。100: 增益=16X。101:增益=32X。110: 增益=64X。111:增益=128X。

OSR_P<2:0>:采集传感器信号时的过采样, 000:1024X, 001:2048X, 010:4096X, 011:8192X, 100:256X, 101:512X, 110:16384X, 111:32768X。

工作模式说明:

组合数据采集模式: 设置'sco'=1 和'measurement_control'=010 进入组合数据采集模式。

芯片上电后先后进行一次温度数据采集和一次传感器数据采集, 完成后回到待机模式, 并自动将'sco'置 0。在组合采集模式下, "Data_out_control"寄存器必须设置为 0, 校准后的温度数据储存在 0x09~0x0A 寄存器, 压力数据储存在 0x06~0x08 寄存器。

休眠数据采集模式: 设置'sco'=1 和'measurement_control'=011 进入休眠数据采集模式。

芯片上电后, 以一定的时间间隔进行一次温度数据采集和一次传感器数据采集, 间隔时间由'sleep_time'设置, 范围为 62.5ms 到 1s。除非手动将'sco'置 0, 不然不会停止采集。在休眠数据采集模式下'Data_out_control'必须设置为 0, 校准后的温度数据储存在 0x09~0x0A 寄存器, 压力数据储存在 0x06~0x08 寄存器。

■ 组合模式读取数据按照如下指令顺序进行操作(其他模式参考 0x30 寄存器定义相应调整即可):

- 1、发送指令 0x0A(即 00001010)到 0x30 寄存器进行一次温度采集, 一次压力数据采集。
- 2、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, 否则等待延迟 20ms 读取。
- 3、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), 读取 0x09、0x0A 两个寄存器地址数据构成 16 位 AD 值 (温度数据 AD 值)。

4、按以下公式换算成实际压力、温度值:

(1) 最高位为“0”代表正压/正温度:

$$\text{Pressure} = \text{Pressure_ADC} / k;$$

$$\text{Temperature} = \text{Temperature_ADC} / 256;$$

(2) 最高位为“1”代表负压/负温度:

$$\text{Pressure} = (\text{Pressure_ADC} - 16777216) / k;$$

$$\text{Temperature} = (\text{Temperature_ADC} - 65536) / 256;$$

注：1、传感器校准后的输出可视为当前实际压力值；

2、传感器压力输出单位 Pa (默认)，若要显示其他单位，可在换算公式里输入相应的系数进行换算；

3、关于上述压力 ADC 换算公式中 k 值的选取可参照下表：

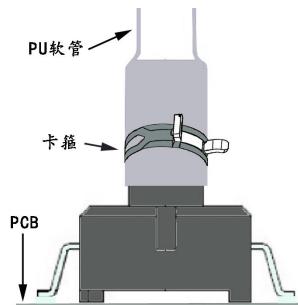
P 量程(kpa)	k 值
500 < P ≤ 1000	8
260 < P ≤ 500	16
131 < P ≤ 260	32
65 < P ≤ 131	64
32 < P ≤ 65	128
16 < P ≤ 32	256
8 < P ≤ 16	512
4 < P ≤ 8	1024
2 ≤ P ≤ 4	2048
1 ≤ P < 2	4096
P < 1	8192

量程 P 为测量区间的最大值，比如，测量-100~200kpa，P=200kpa，k 值为 32。

● 安装提示

建议采用以下步骤传导气体压力

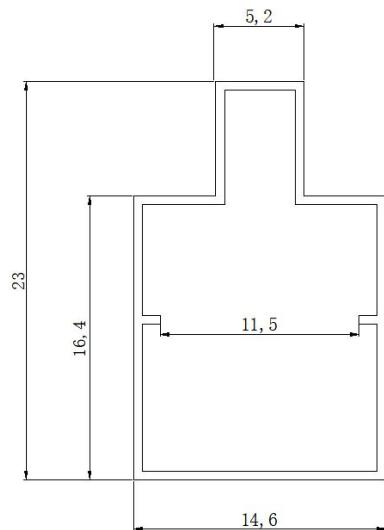
- 1、选择合适规格(内径 2.5MM)和强度的软管
- 2、如需要，采用卡箍固定软管(压力 ≥ 500kPa)
- 3、请勿堵住进气口
- 4、避免过度外力操作
- 5、如需卡箍，请联系厂家



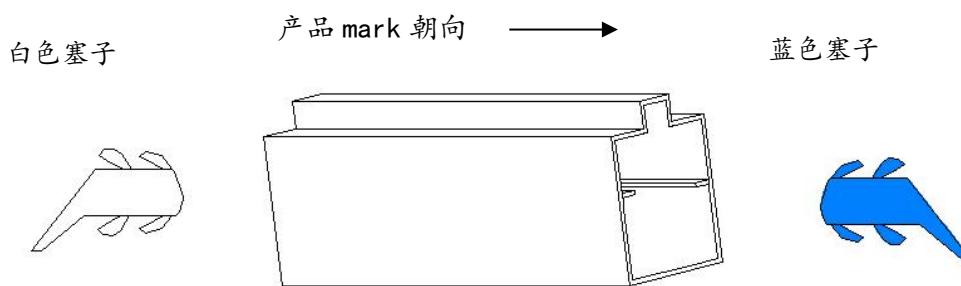
- 包装信息

料管信息 (单位为毫米)

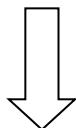
每管数量:38 PCS



料管截面示意图



(520mm x 24.6mm x 23mm, 38PCS)



(纸盒: 530mm x 145mm x 53mm, 760PCS)

包装示意图

备注: 包装信息仅供参考; 非标准包装数量发货, 包装细节略有不同, 请以实际包装为准。

● **使用注意事项** (未标注情况下, 以下描述为所有 CFSensor 产品通用注意事项)

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

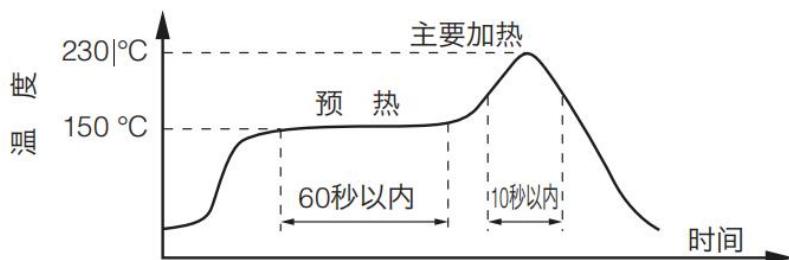
- 请使用头部温度在 260 ~ 300°C (30 W) 的电烙铁在 5 秒以内实施作业。
- 在端子上施加负载进行焊接的情况下, 输出可能会发生变化, 因此请注意。
- 请充分清洗电烙铁头。

2) DIP 焊接 (DIP 端子型)

- 在温度为 260 °C 以下的 DIP 焊锡槽内 5 秒以内实施作业。
- 安装在热容量较小的基板上时, 由于可能会发生热变形, 因此请避免采用 DIP 焊接。

3) 回流焊接 (SMD 端子型)

推荐的回流炉温度设置条件如下所示



- 由于无法做到自校准, 因此请慎重地对准端子与走线的位置。
- 设置的温度为端子附近的印刷电路板上所测得的值。
- 因为由于装置, 条件等原因, 压力导入口的先端因为高温会发生溶解和变形, 务必请在实际的贴装条件下, 进行确认测试。

4) 焊接部的修正

- 请一次性完成修正。
- 对搭焊进行修正时, 请使用头部形状较平滑的电烙铁, 请勿追加涂敷助焊剂。
- 关于电烙铁头部的温度, 请使用在规格书所记载的温度以下的电烙铁。
- 5) 在端子上施加过度的力后, 会引发变形损害焊接性, 因此请避免使产品掉落, 或进行繁杂的使用。
- 6) 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。
- 7) 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。
- 8) 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。
- 9) 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。

2 清洗

- 1) 部分产品内部未填充防水胶, 且产品为开放型结构, 因此请注意不要使清洗液侵入内部。
- 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。

3 环境

- 1) 本产品必须储存在清洁、无腐蚀性气体的室内环境中; 如长时间储存, 建议将其储存在氮气柜等无氧化环境中。
- 2) 本产品并非防滴构造, 因此请勿在可能溅到水等的场所中使用。
- 3) 请勿在产生凝露环境中使用。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。
- 4) 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。
- 5) 请避免采用超声波等施加高频振动的使用方法。

■ 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。

■ 关于其他使用

1) 压力范围，安装方法错误时，会造成事故，因此请注意。

2) 能够直接使用的压力媒介请参考规格书中介质描述。除此以外的媒介，尤其是在腐蚀性气体（有机溶剂气体，亚硫酸气体，硫化氢气体等）和异物的媒介中使用时，会造成故障和破损，因此请避免在上述环境中使用。

3) 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。另外，使用时请避免堵塞大气导入口。

4) 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。

5) 由于可能因静电而造成破坏，因此使用时请注意以下事项。

· 保存时，请使用导电性的材料使端子之间短路，或者用铝箔等整体包覆起来。由于塑料的容器容易带电，因此保存运输时请勿使用。

· 使用时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。

6) 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。

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。

· 请按照端子连接图进行接线。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。

· 为保证安全，特别是重要的用途，请务必考虑双重安全电路等的配置。

· 请勿施加最大施加压力以上的压力。此外，请注意不要使异物混入压力媒体。否则会造成产品废弃，或因媒体吹出而引发事故。

· 对产品进行固定和对压力导入口进行连接时请慎重。否则会因产品飞散，媒体吹出而引发事故。

· 由于部分产品前端较锐利，因此使用时请注意不要伤到身体。

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。CFSensor 保留对此处的任何产品进行更改的权利，恕不另行通知。CFSensor 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。

更多问题，请联系 CFsensor

电话(微信): 13855373761/18226771331

总机: 0553-3116860/0553-2167689

电子邮箱: Sales2@CFSensor.com

QQ: 445792528 / 337803064

IIC Example Code (C51 Language)

```
#include <reg52.h>
#include <math.h>

#define DELAY_TIME 600
#define TRUE 1
#define FALSE 0
#define uchar unsigned char
#define uint unsigned int

//----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);
    F0 = SDA;
    DELAY(DELAY_TIME / 2);
    SCL = 0;
    DELAY(DELAY_TIME);
    if (F0 == 1)
        return FALSE;
    return TRUE;
}
```

-----Write One Byte of Data-----

```
void WriteI2CByte(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 ReadI2CByte(void) reentrant
{
    char b = 0, i;
    for (i = 0; i < 8; i++)
    {
        SDA = 1;
        DELAY(DELAY_TIME);
        SCL = 1;
        DELAY(DELAY_TIME);
        F0 = SDA;
        DELAY(DELAY_TIME);
        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----
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(0x6D << 1 + 0); //IIC WRITE operation, SLAVER address bit: 0x6D
    acktemp = Check_Acknowledge(); //check the SLAVER
    Writel2CByte(addr); //address
    acktemp = Check_Acknowledge();
    Writel2CByte(thedata); //thedata
    acktemp = Check_Acknowledge();
    I2C_Stop(); //IIC STOP
}

//----Read One Byte of Data,Data from SLAVER to the MASTER----
uchar Read_One_Byte(uchar addr)
{
```

```
bit acktemp = 1;
uchar mydata;

I2C_Start();
WriteI2CByte(0x6D << 1 + 0);      //IIC WRITE operation, SLAVER address bit: 0x6D
acktemp = Check_Acknowledge();
WriteI2CByte(addr);
acktemp = Check_Acknowledge();
I2C_Start();
WriteI2CByte(0x6D << 1 + 1);      //IIC READ operation, SLAVER address bit: 0x6D
acktemp = Check_Acknowledge();
mydata = ReadI2CByte();
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++)
            ;
}

//----The main function----
void main(void)
{
    uchar pressure_H, pressure_M, pressure_L, temperature_H, temperature_L;
    //temporary variables of pressure and temperature
    //临时变量，用于保存从传感器中读出的与压力和温度相关的寄存器的数值
    long int pressure_adc, temperature_adc;
    //The value of pressure and temperature converted by the sensor's ADC
    //临时变量，用于保存传感器 ADC 转换后的压力值和温度值
    long float pressure, temperature;
    //The calibrated value of pressure and temperature
    //用于保存校准后的压力值和温度值
    Delay_xms(1000); //delay 1000ms

    while (1)
    {
        Write_One_Byte(0x30, 0x0A);
        //indicate a combined conversion (once temperature conversion immediately followed by once
        //sensor signal conversion)
        //0x30 里写入测量命令, 000: 单次温度测量; 001: 单次压力测量; 010: 组合: 单次压力和温度
        //测量; 011: 休眠方式 (以一定的时间间隔执行组合模式测量)
    }
}
```

```
while ((Read_One_Byte(0x30) & 0x08) > 0);
//Judge whether Data collection is over 判断数据采集是否结束
Delay_xma(20); //延时 20ms 后读取数据
pressure_H = Read_One_Byte(0x06);
pressure_M = Read_One_Byte(0x07);
pressure_L = Read_One_Byte(0x08);
// Read ADC output Data of Pressure 读取保存压力值的 3 个寄存器的值
pressure_adc = pressure_H * 65536 + pressure_M * 256 + pressure_L;
//Compute the value of pressure converted by ADC 计算传感器 ADC 转换后的压力值

if (pressure_adc > 8388608) //超过 8388606 为负压值，需在显示终端做正负号处理
    pressure = (pressure_adc - 16777216) / 64; //单位为 Pa
else
    pressure = pressure_adc / 64; //单位为 Pa
//The conversion formula of calibrated pressure, its unit is Pa 计算最终校准后的压力值

temperature_H = Read_One_Byte(0x09);
temperature_L = Read_One_Byte(0x0A);
//Read ADC output data of temperature 读取保存温度值的 2 个寄存器的值
temperature_adc = temperature_H * 256 + temperature_L;
//Compute the value of temperature converted by ADC 计算传感器 ADC 转换后的压力温度值
temperature = (temperature_adc - 65536) / 256; //单位为摄氏度
else
    temperature = temperature_adc / 256; //单位为摄氏度
//The conversion formula of calibrated temperature, its unit is Centigrade 计算最终校准后的温度值

Delay_xms(100); //delay 100ms
}
```