



# GZP6816D

## 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 <sup>2</sup> C 通讯协议	7
9. 一般读取指令	10
10. 换算	10
11. 使用注意事项	10
11.1. 焊接	10
11.2. 清洗要求	11
11.3. 存储和运输	11
11.4. 其他使用注意事项	11
12. 包装信息	13
<b>安全注意事项</b>	14
IIC Example Code（C51 Language）	15
<b>免责声明</b>	21



## 文件修订历史

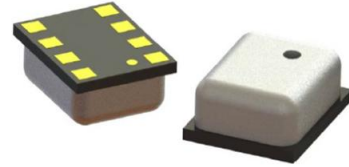
修订	描述	日期
V1.0	初始版本	2021.4.2
V1.1	增加封面、目录	2021.9.23
V1.2	增加寄存器、指令说明	2021.11.18
V1.3	更新程序量程、延时	2021.12.24
V1.4	调整产品归类	2022.03.16

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 30kPa ~ 110kPa
- 绝压型
- 电源电压: 1.8V ~ 3.6V
- 电流消耗: <80uA(最大过采样率一次测量)
- 休眠状态电流: 0.1uA (25°C)
- IIC 通讯
- 绝对压力精度:  $\pm 1\text{hPa}$  (8.3m)
- 相对压力精度:  $\pm 0.12\text{hPa}$  (1m)
- 产品尺寸: 2.5\*2.0\*0.98mm



## 2. 应用领域

- 无人机、气象站、导航
- 运动穿戴设备
- 手机等移动设备
- 钟表、家电
- 监护仪、制氧机等医疗设备
- 便携式和固定式气压计

## 3. 概述

GZP6816D 是一款具有高精度和低电流消耗的小型化数字式气体压力传感器，兼具压力和温度测量两种特点。内部信号处理器将压力和温度传感器的输出分别转换为 24 位、16 位数据。每个压力传感器已单独校准并且包含校准系数，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，传感器测量和校准系数可通过串行 I<sup>2</sup>C 接口获得。

## 4. 性能指标

供电电源: 1.8V~3.6V DC

测试参考: 温度@25°C, 压力@950~1050hPa



表 1. 性能指标

Parameter	Conditions	Min	Typ	Max	Units
测温范围	内部温度传感器	-40		150	°C
绝对压力精度		-1		1	hPa
相对压力精度		-0.12		0.12	hPa
温度测量精度		-1		1	°C
过载压力			2x		Rated
工作温度		-30		100	°C
补偿温度		-10		60	°C
储存温度		-40		150	°C

## 5. 电气特性

表 2. 电气特性

Parameter	Conditions	Min	Typ	Max	Units
	$V_{DD}=2V$ 时	32			dB
1Hz 转换率测量时的平均电流	OSR_P	过采样率 128x	80		μA
		过采样率 64x	42		
		过采样率 32x	23		
		过采样率 16x	13		
		过采样率 8x	8		
		过采样率 3x	6		
		过采样率 2x	4		
峰值电流	峰值电流		0.3		mA
待机电流	25°C 时休眠状态的待机电流		50	250	nA
单次测量时间 (包括测外部电桥和测温时间, 其中测温的 OSR 为 1024x)	OSR_P	过采样率 128x	203		ms
		过采样率 64x	105		
		过采样率 32x	56		
		过采样率 16x	31		
		过采样率 8x	19		
		过采样率 4x	13		
		过采样率 2x	10		
ADC 转换速率	OSR 为 2x ~ 128x	20		1350	Hz
I <sup>2</sup> C 时钟频率				3.4	MHz
温度分辨率			0.003		K/LSB
启动时间	$V_{DD}$ 上升至接口开始通讯的时间			1	ms
	$V_{DD}$ 上升至开始测量的时间			2.5	ms
唤醒时间	休眠状态至接口开始通讯的时间			0.5	ms
	休眠状态至开始测量的时间			2	ms

## 6. 外形结构 (单位为毫米)

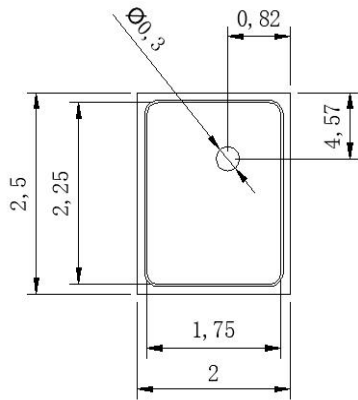


图 1. Top view

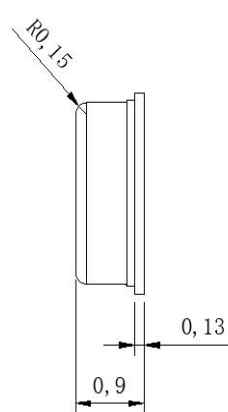


图 2.Side view

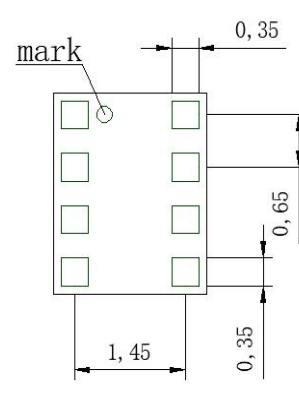


图 3.Bottom view

## 7. 电气连接

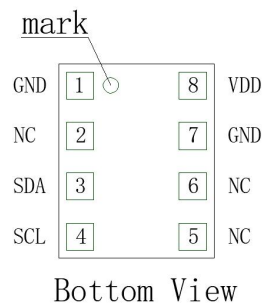


图 4. 引脚图对应关系(Bottom view)

表 3. 引脚对应关系

引脚	符号	功能
1	GND	接地
2	NC	无定义
3	SDA	I <sup>2</sup> C 双向数据线
4	SCL	I <sup>2</sup> C 时钟线
5	NC	无定义
6	NC	无定义
7	GND	接地
8	VDD	电源

### 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(3.6Vdc)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

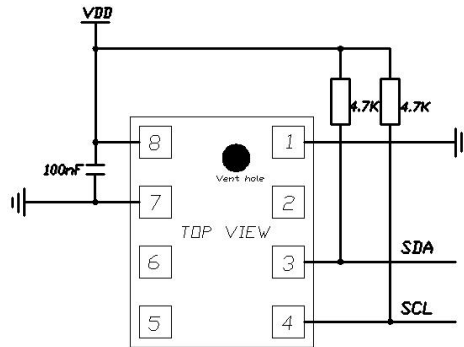


图 5. 典型应用

## 8. I<sup>2</sup>C 通讯协议

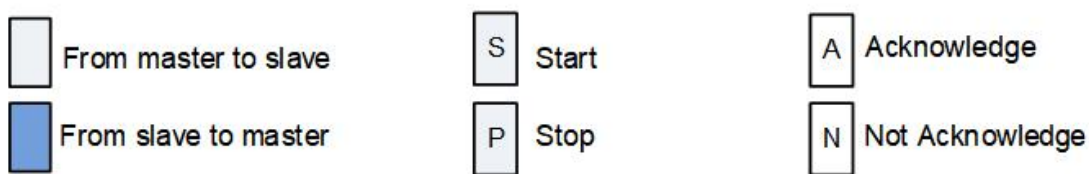
I2C 总线使用 SCL 和 SDA 作为信号线，这两根线都通过上拉电阻（典型值 4.7K）连接到 VDD，不通信时都保持为高电平。

IIC 设备地址为 0x78。

设备地址为 0x78 左移 1 位后为 0xF0,为 IIC 的写操作。

设备地址为 0x78 左移 1 位后再加 1 为 0xF1,为 IIC 的读操作。

IIC 通讯示意图如下所示：

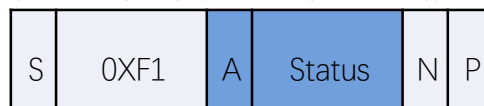


(1) 写命令中格式如下：



(2) 发送完写命令后需要等待一段时间再发送读命令，因为内部完成整个测量需要一段时间。等待的时长取决于压力过采样率和温度过采样率的设置（默认 OSR128X，单次测量时间为 203mS）。

可以通过发送 0XF1 命令读取传感器的状态值，来判断传感器数据采集是否已完成。





(3) 读数指令如下:

S	0XF1	A	Status	A	PressureD at[23:16]	A	PressureD at [15:8]	A	PressureD at [7:0]	A	TempDat [15:8]	A	TempDat [7:0]	N	P
---	------	---	--------	---	------------------------	---	------------------------	---	-----------------------	---	-------------------	---	------------------	---	---

I<sup>2</sup>C 读出 5 字节校准后的压力值和温度值

表 4. 比特位描述

Status 字节描述		
比特位	定义	描述
Bit7	保留	固定为 0
Bit6	上电指示	1 上电; 0 掉电
Bit5	采集结束标志	1 数据采集未完成 0 数据采集完成, 可读取数据
Bit4	保留	固定为 0
Bit3	保留	固定为 0
Bit2	保留	固定为 1
Bit1	保留	固定为 0
Bit0	保留	固定为 0

■ IIC 指令表

IIC 指令表 5.

Command (byte)	Return	Description	NOR Mode	CMD Mode
0x00~0x1F	16 位数据	从 OTP 中读取数据; 地址和命令是相同的	支持	支持
0x40~0x5F 跟随的命令字为 (0x0000 ~ 0xFFFF)	—	向 OTP 中写数据; 写地址是指令值减去 0x40 (地址是 0x00 到 0x1F)	支持	支持
0xA0~0xA7 跟随的命令字为 0xFFFF	24 位未校准 数据	<b>Get_Raw</b> 该指令用来执行一次传感器测量, 将测量到未 经过校准的 ADC 数据写入输出寄存器。	支持	支持
0xA8	—	<b>Start_NOM</b> 退出命令模式, 进入正常模式。	不支持	支持
0xA9	—	<b>Start_CM</b> 退出正常模式, 进入命令模式。	支持	不支持
0xAA	—	<b>Write_ChecksumC</b> 如果 CRC 校验值还没有写入 OTP 中, 指令对 OTP 中数据进行校验并将校验码写入 OTP 中。	支持	支持
0xAC	24 位校准后 的电桥值和 16 为校准后 的温度值	<b>Get_Cal</b> 使用 OTP 中的配置进行整体测量 (AZBM, BM, AZTM, 和 TM), 并且把校准后的 电桥值和温度值写入接口	支持	支持
0xB0~0xBF	24 位校准后 的电桥值和 16 为校准后 的温度值	(见表 6)	支持	支持





表 6 Get\_Cal\_S 命令

Command 0xBX(HEX)	Function	Detail
X 的第[3] Bit	测量温度时 ADC 的过采样率 OSR_T	0: 4x 过采样率    1: 8x 过采样率
X 的第[2:0] Bit	测量外部电桥时 ADC 的过采样率 OSR_P	000: 128x 过采样率    100: 8x 过采样率 001: 64x 过采样率    101: 4x 过采样率 010: 32x 过采样率    110: 2x 过采样率 011: 16x 过采样率    111: 1x 过采样率

Get\_Cal\_S 命令(0xB0 ~ 0xBF)与 0xAC 命令几乎一样，除了测量时 ADC 的过采样率的设定有所不同，例如，要设置温度 ADC 为 4 倍过采样，压力 ADC 为 1 倍过采样，命令格式为 0xB7，直接用 0xB7 代替 0xAC 即可。

表 7. OTP 寄存器表

Addr	Bit Range	Description	Notes/Explanations
0x00~0x13		校准系数	
0x14	15:14	温度过采样	00:8X    01:16 X    10: 32X    11:64X
	13:11	压力过采样	000: 128X    001: 64X    010: 32X 011: 16X    100: 8X    101: 4X
			110: 2X    111: 1X
	10:8	ADC 偏置	000 : 1/16 → [-1/16, 15/16]
			001 : 2/16 → [-2/16, 14/16]
			010 : 3/16 → [-3/16, 13/16]
			011 : 4/16 → [-4/16, 12/16]
			100 : 5/16 → [-5/16, 11/16]
101 : 6/16 → [-6/16, 10/16]			
110 : 7/16 → [-7/16, 9/16]			
111 : 8/16 → [-8/16, 8/16]			
7:6	保留		
5	传感器原始信号极性反转	1: 不反转, 0: 反转	
4:0	保留		
0x15~0x16		内部测试	
0x17	4	中断使能	0: disable, 1: enable
	3:2	中断 0 配置位	00 : 无效
			01 : 校准值高于预设上限(TH_H) 10 : 校准值低于预设下限(TH_L) 11 : 校准值高于预设上限(TH_H)或者低于预设下限(TH_L)
1:0	中断 1 配置位	00 : 无效	
		01 : 校准值高于预设上限(TH_H) 10 : 校准值低于预设下限(TH_L) 11 : 校准值高于预设上限(TH_H)或者低于预设下限(TH_L)	



## 9. 一般读取指令

表 8.读取指令

Command(byte)	Return	Description
0xAC	24 位校准后的压力值和16 位校准后的温度值	Get_Cal 使用 OTP 中的配置进行整体测量并且把校准后的电桥值和温度值写入接口

## 10. 换算

读到校准数据后，需要将以 AD 值形式表示的无符号数进行简单的换算。

为方便理解我们假设读到的校准数据为：0x04 0x9B 0xB0 0xC5 0x56 0xAA

0x04 为状态字 Bit5 为 1 表明最近一次 I2C 忙，需要等待一段时间。如果 Bit5 为 0 表明设备不忙，可以读取数据。关于状态字各比特的详细描述请参见表 4.比特位描述。

0x9B 0xB0 0xC5 三个字节为电桥校准值

0x56 0xAA 两个字节为温度校准值

实际某压力点电桥校准值换算：将 0x9B 0xB0 0xC5 转换为十进制数为 10203333，

本次计算假设校准时使用的量程为 30Kpa-110Kpa，对应的 AD 输出为 1677722~15099494 (10%AD~90%AD) ，

根据输入输出关系公式( $pressure = ((P_{MAX} - P_{MIN}) / (D_{MAX} - D_{MIN}) * (D_{test} - D_{MIN}) + P_{MIN})$ )得到：

实际压力值 =  $(110 - 30) / (15099494 - 1677722) * (10203333 - 1677722) + 20 = 70.816$  Kpa

温度校准值换算：将 0x56 0xAA 转换为十进制数为 22186，由于读取到的校准数据是以百分比形式表示的，这个百分比在数值上等于我们换算得到的十进制数与 16bits 无符号数的最大值 (65535) 之比，所以在换算百分比时可进行如下计算

$22186 / 65536 * 100\% = 33.85\%$

温度的校准范围规定为 -40℃~150℃ 所以校准值 =  $(150 - (-40)) * 33.85\% - 40 = 24.32$ ℃

注释：需要注意的是，温度传感器需校准后方可使用，未校准的温度值为原始值，不具备参考价值。

## 11. 使用注意事项

### 11.1. 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。
- 请保持电烙铁头洁净。

#### 2) DIP 焊接 (DIP 端子型)

- 在温度为 260 °C 以下的 DIP 焊锡槽内在 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

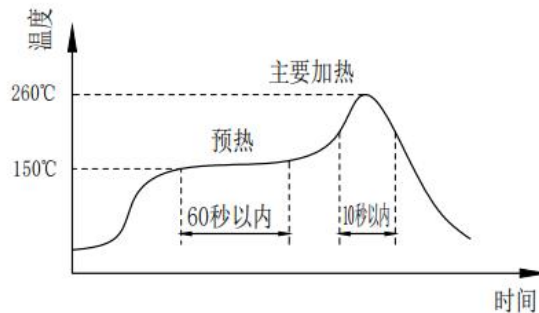


图 6.回流焊接

- 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。
  - 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。

#### 11.2. 清洗要求

- 1) 由于产品为开放型，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。

#### 11.3. 存储和运输

- 1) 本产品为非防滴构造，因此请勿在可能溅到水等的场所中使用。
- 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。

#### 11.4. 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) 能够直接使用的压力媒介仅为干燥空气。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）和含有水分，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。
- 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。另外，使用时请避免堵塞大气导入口。
- 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。

## 12. 包装信息

载带信息 (单位为毫米)

每卷数量: 10,000 PCS

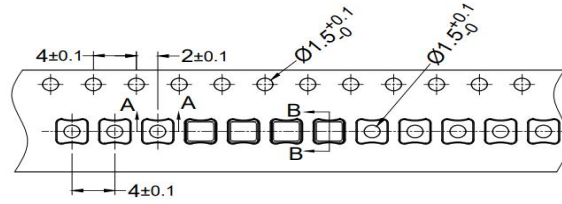


图 7. 载带

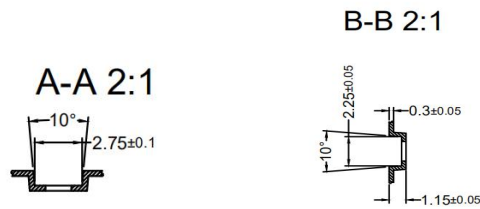


图 8. 载带口袋细节图

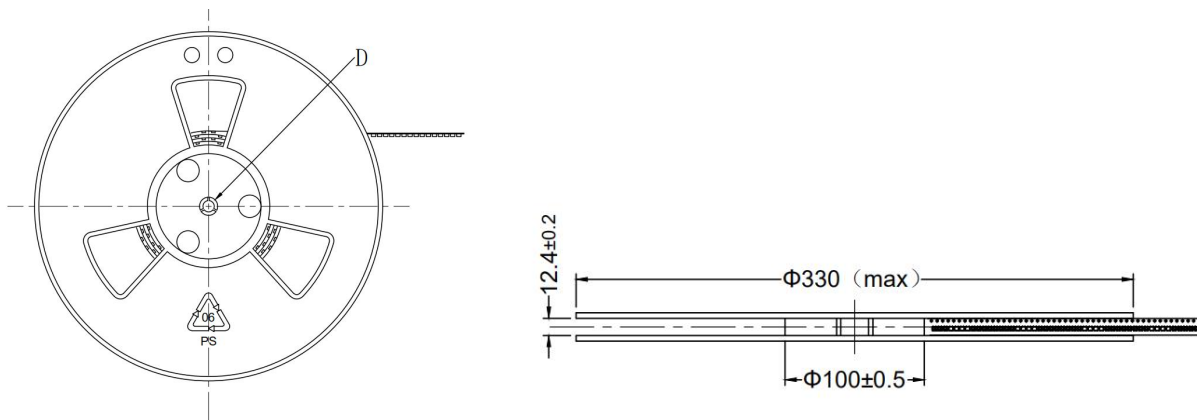


图 9. 载带转盘



## 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 (C51 Language)

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

#define DELAY_TIME 20

#define PMIN 30.0 // The minimum range pressure value for example 30Kpa 传感器最低量程
#define PMAX 110.0 //The full scale pressure value, for example 110Kpa 传感器最大量程
#define DMIN 1677722.0 //AD value corresponding to The minimum range pressure,
// for example 10%AD=2^24*0.1 最低量程对应的 AD 值
#define DMAX 15099494.0 //AD Value Corresponding to The full scale pressure value,
// for example 90%AD=2^24*0.9 最高量程对应的 AD 值

sbit SCL = P3 ^ 3; //IIC clock line IIC 时钟线
sbit SDA = P1 ^ 5; //IIC data line IIC 数据线

//The 7-bit IIC address of the sensor is 0x78 传感器 7 位 IIC 总线地址
unsigned char Device_Address = 0x78 << 1;

float pressure_kpa = 0.0; //变量，用于保存压力值，单位为 KPa
unsigned long pressure_pa = 0; //变量，用于保存压力值，单位为 Pa
float temperature = 0.0; //变量，用于保存温度值，单位为 °C

void Delay_Ms(unsigned char n) //Ms Time-Delay function Ms 延时函数
{
    unsigned char i,j;
    for(i=0;i<n;i++)
        for(j=0;j<123;j++);
}

void Start(void) //Start signal IIC 总线起始信号
{
    SDA = 1;
    Delay_Ms(DELAY_TIME);
    SCL = 1;
    Delay_Ms(DELAY_TIME);
    SDA = 0;
    Delay_Ms(DELAY_TIME);
    SCL = 0;
    Delay_Ms(DELAY_TIME);
}
```



```
void Stop(void) //Stop signal IIC 总线停止信号
{
    SDA = 0;
    Delay_Ms(DELAY_TIME);
    SCL = 1;
    Delay_Ms(DELAY_TIME);
    SDA = 1;
    Delay_Ms(DELAY_TIME);
    SCL = 0;
    Delay_Ms(DELAY_TIME);
}

unsigned char Check_ACK(void) //Read sensor's ACK signal 读取传感器的 ACK 信号
{
    unsigned char ack;
    SDA = 1;
    Delay_Ms(DELAY_TIME);
    SCL = 1;
    Delay_Ms(DELAY_TIME / 2);
    ack = SDA;
    Delay_Ms(DELAY_TIME / 2);
    SCL = 0;
    Delay_Ms(DELAY_TIME);
    return ack;
}

void Send_ACK(void) //Send ACK signal to sensor 向传感器发送 ACK 信号
{
    SDA = 0;
    Delay_Ms(DELAY_TIME);
    SCL = 1;
    Delay_Ms(DELAY_TIME);
    SCL = 0;
    Delay_Ms(DELAY_TIME);
    SDA = 1;
    Delay_Ms(DELAY_TIME);
}

void SendByte(unsigned char byte1) //Send one byte to sensor 向传感器发送 1 个字节
{
    unsigned char i = 0;
    do
    {
        if (byte1 & 0x80)
```





```
{ SDA = 1; }
else
{ SDA = 0; }
Delay_Ms(DELAY_TIME);
SCL = 1;
Delay_Ms(DELAY_TIME);
byte1 <<= 1;
i++;
SCL = 0;
} while (i < 8);
SCL = 0;
Delay_Ms(DELAY_TIME);
}
```

```
unsigned char ReceiveByte(void) //Receive one byte from sensor 从传感器读取 1 个字节
{
    unsigned char i = 0, tmp = 0;
    do
    {
        tmp <<= 1;
        SCL = 1;
        Delay_Ms(DELAY_TIME);
        if (SDA)
        { tmp |= 1; }
        SCL = 0;
        Delay_Ms(DELAY_TIME);
        i++;
    } while (i < 8);
    return tmp;
}
```

```
//Write several bytes of data to the sensor 向传感器写若干个字节
unsigned char GZP6816D_IIC_Write(unsigned char address, unsigned char *buf, unsigned char count)
{
    unsigned char timeout, ack;
    address &= 0xFE; // The lowest bit of address is 0 means writing 地址值最低位为 0 表示写
    Start();
    Delay_Ms(DELAY_TIME);
    SendByte(address); //Send the address to the sensor 向传感器发送地址值
    Delay_Ms(DELAY_TIME);
    timeout = 0;
    do //Getting the ACK response from the sensor means success, otherwise means failure
    { //如能获取传感器的 ACK 信号, 则表示 IIC 总线通信成功; 否则表示通信失败
        ack = Check_ACK();
```



```
        timeout++;
        if (timeout == 10)
        {
            Stop();
            return 1;
        }
    } while (ack);
    while (count)
    {
        SendByte(*buf); //Write bytes to the sensor 向传感器写字节数据
        Delay_Ms(DELAY_TIME);
        timeout = 0;
        do
        {
            ack = Check_ACK();
            timeout++;
            if (timeout == 10)
            {
                return 2;
            }
        } while (0);
        buf++;
        count--;
    }
    Stop();
    return 0;
}
```

//Read several bytes of data from the sensor 从传感器读取若干字节

unsigned char GZP6816D\_IIC\_Read(unsigned char address, unsigned char \*buf, unsigned char count)

```
{
    unsigned char timeout, ack;
    address |= 0x01; // The lowest bit of address is 1 means reading 地址值最低位为 1 表示读
    Start();
    SendByte(address);
    Delay_Ms(DELAY_TIME);
    timeout = 0;
    do //Getting the ACK response from the sensor means success, otherwise means failure
    { //如能获取传感器的 ACK 信号，则表示 IIC 总线通信成功；否则表示通信失败
        ack = Check_ACK();
        timeout++;
        if (timeout == 10)
        {
            Stop();

```



```
        return 1;
    }
} while (ack);
Delay_Ms(DELAY_TIME);
while (count)
{
    *buf = ReceiveByte();
    if (count != 1) //Except the last byte, send ACK to the sensor after receiving byte
        Send_ACK(); //除了最后一个字节外，每接收一个字节，向传感器发送 ACK 信号
    buf++;
    count--;
}
Stop();
return 0;
}

//Read the status of the sensor and judge whether IIC is busy
unsigned char GZP6816D_IsBusy(void) //读取传感器的状态变量，用于判断数据是否已准备好被读取
{
    unsigned char status;
    GZP6816D_IIC_Read(Device_Address, &status, 1);
    status = (status >> 5) & 0x01;
    return status;
}

void GZP6816D_get_cal(void) //The function of reading pressure and temperature from the sensor
{
    //从传感器 GZP6816D 读取压力值和温度值的函数
    unsigned char buffer[6] = {0}; //Temp variables used to restoring bytes from the sensor
    //临时变量，用于保存从传感器中读出的字节数据

    unsigned long Dtest = 0;
    unsigned int temp_raw = 0;

    buffer[0] = 0xAC; //Send 0xAC command and read the returned six-byte data
    GZP6816D_IIC_Write(Device_Address, buffer, 1); //发送 0xAC 命令，启动一次压力和温度测量

    Delay_Ms(DELAY_TIME);
    while (1)
    {
        if (GZP6816D_IsBusy()) //判断传感器数据是否已准备好被读取
            {Delay_Ms(DELAY_TIME); }
        else
            break;
    }
    GZP6816D_IIC_Read(Device_Address, buffer, 6); //读取传感器 6 个字节数据
```



```
//Computing the calibrated pressure and temperature values  计算校准后的压力值和温度值
Dtest = (unsigned long)(((unsigned long)buffer[1]) << 16) | (((unsigned int)buffer[2]) << 8) | ((unsigned
char)buffer[3]);
temp_raw = ((unsigned int)buffer[4] << 8) | (buffer[5] << 0);

//The calibrated pressure value is converted into actual values
if (Dtest != 0)
{ //将校准后的压力值转换成真实的压力值
    pressure_kpa = (float) ((PMAX-PMIN)/(DMAX-DMIN)*(Dtest-DMIN)+PMIN); //单位: KPa
    pressure_pa = (unsigned long) (pressure_kpa * 1000.0); //单位: Pa
}
else
{
    pressure_kpa = 0.0; //pressure value, its unit is KPa 压力值, 单位: KPa
    pressure_pa = 0; //pressure value, its unit is Pa 压力值, 单位: Pa
}
temperature = (float) temp_raw / 65536; //The calibrated temperature value is converted into actual
values
temperature = ((float) temp * 19000 - 4000) / 100; // its unit is °C
//将校准后的温度值转换成真实的温度值, 单位: °C
}

void main() //The main function
{
    Printf("CFSensor Ltd");
    while(1)
    {
        GZP6816D_get_cal();
        Delay_Ms(1000);
        Printf("The pressure from GZP6816D is %ld Pa\r\n", pressure_pa);
        Printf("The temperature from GZP6816D is %f CTG\r\n\r\n", temperature);
    }
}
```



## 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。