

DWSPD 系列
MEMS 微差压传感器
($\pm 500\text{Pa}$ ~ $\pm 35\text{KPa}$)

Duwei[®]

产品概述

DWSPD是杜威智能研发的一款数字信号输出的微差压传感器。该系列使用了高性能ASIC来校准和补偿MEMS传感器元件，保证了产品的可靠性。测量范围由 $\pm 500\text{Pa}$ 到 $\pm 350\text{kPa}$ 的压力差信号转换为I²C数字输出信号。压力传感器可以集成在一个标准的印刷电路板上。

广泛用于空气或中性气体检测控制中，适用于各类通风、空调系统及设备的气体压力检测，洁净工作室、生物实验室、气体管道过滤器阻力的压差监测应用。

主要参数

- 压力范围
最小压差范围 -500Pa 至 500Pa
最大压差范围 -35kPa 至 35kPa
- 工作温度范围 $-20\text{s}^\circ\text{C}$ 至 $70\text{s}^\circ\text{C}$
- 初始误差范围
数字输出小于 $\pm 1.0\% \text{F.S}$
模拟输出小于 $\pm 1.5\% \text{F.S}$
- 24位I²C数字和16位模拟输出



场景应用

- 垂直端口
- 消防余压监测
- 呼吸机应用
- HVAC暖通空调监测
- 生物安全柜监测
- 微差压变送器应用

封装信息

型号	封装	尺寸
DWSPD	SOIC-16	10.26*7.52mm

目录

一、引脚的配置和功能.....	3
二、额定参数.....	4
三、工作范围.....	4
四、技术规格.....	5
1、电气特性.....	5
2、I ² C 时序图.....	5
3、I ² C 电气特性.....	6
五、功能描述.....	6
1、概述.....	6
2、数字输出计算方式.....	7
3、模拟输出计算方式.....	7
4、寄存器.....	8
5、I ² C 接口.....	8
六、典型应用电路.....	10
七、产品尺寸.....	11
八、传感器选型.....	12
九、焊接参数.....	13
1、回流焊（SMD 端子）.....	13
2、手动焊接.....	13
十、包装规格信息.....	14
十一、注释.....	15

一、引脚的配置和功能

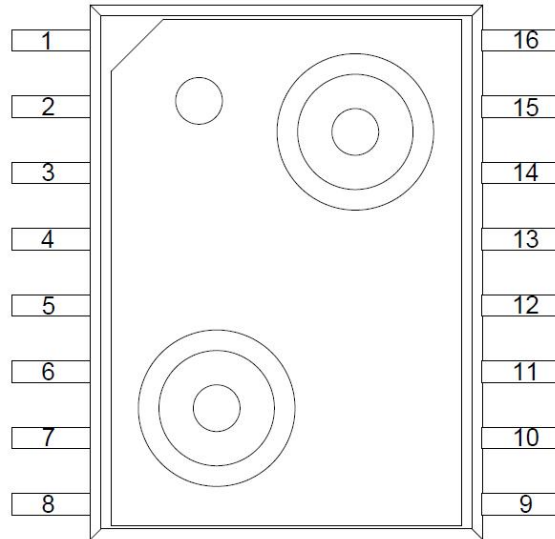


图1 DWSPD系列引脚定义（俯视图）

表1 DWSPD引脚说明

序号	符号	功能
1	NC	无连接
2	NC	无连接
3	NC	无连接
4	GND	地
5	VDD	电源
6	OUT	模拟输出
7	NC	无连接
8	NC	无连接
9	NC	无连接
10	SDA	I ² C 数据信号
11	SCL	I ² C 时钟信号
12	NC	无连接
13	NC	无连接
14	NC	无连接
15	NC	无连接

二、 额定参数

参数	符号	MIN	TYP	MAX	单位	备注
电源电压	VDDmax	-0.3		6.5	V	
模拟输出电流限制				25	mA	
数字引脚电压		-0.3		VDD+0.3	V	25°C
常规压力	Pproof	30				压力范围<10Kpa
		300			kPa	压力范围>10Kpa
瞬时压力	Pburst	50				压力范围<10Kpa
		500			kPa	压力范围>10Kpa
ESD 敏感性	HBM		2		kV	
储存温度	Tstg	-40		100	°C	

三、 工作范围

参数	符号	MIN	TYP	MAX	单位	备注
电源电压	VDD	3	3	3.6	V	VDD=3.3V
		4	5	5.5	V	VDD=5V
工作压力	Pamb			350	kPaD	差压
工作压力范围	Prange	0.2		450	kPa	Pmax-Pmin
数字输入/输出时的高电平电压	VIH	2			V	
数字输入/输出时的低电平电压	VIL			0.8	V	
1°C 时钟频率	Fsclk			400	kHz	
工作温度	Topr	-20		70	°C	

四、技术规格

1、电气特性

参数	符号	MIN	TYP	MAX	单位	备注
重置电源	VDDpdr		2		V	
工作电流	Iavdd		2.4		mA	工作模式
				200	nA	待机模式
ADC 分辨率	RES		24		Bits	
PSRR	PSRR	90	120		dB	
DAC 分辨率			12		Bits	
输出负载电阻	Rload	1			kOhm	模拟输出
输出负载电容	Cload			15	nF	模拟输出
全寿命精度	ACC	-1.0		1.0	%FS	数字输出
	ACC	-1.5		1.5	%FS	模拟输出
精度	ACC	-1.0		1.0	%FS	
偏移	ACC	-0.5		0.5	%FS	
通电时间	Tup		100		ms	
EEPROM 数据保留	Tlive	10			年	@125°C

- 精度包括非线性滞后度、温度滞后度、压力滞后度、温度滞后度；
- 基于 500 小时的 HTOL、LTOL、HTSL、TH (40°C/90%RH)、PCT 和 3 倍回流试验的全寿命精度；

2、I²C 时序图

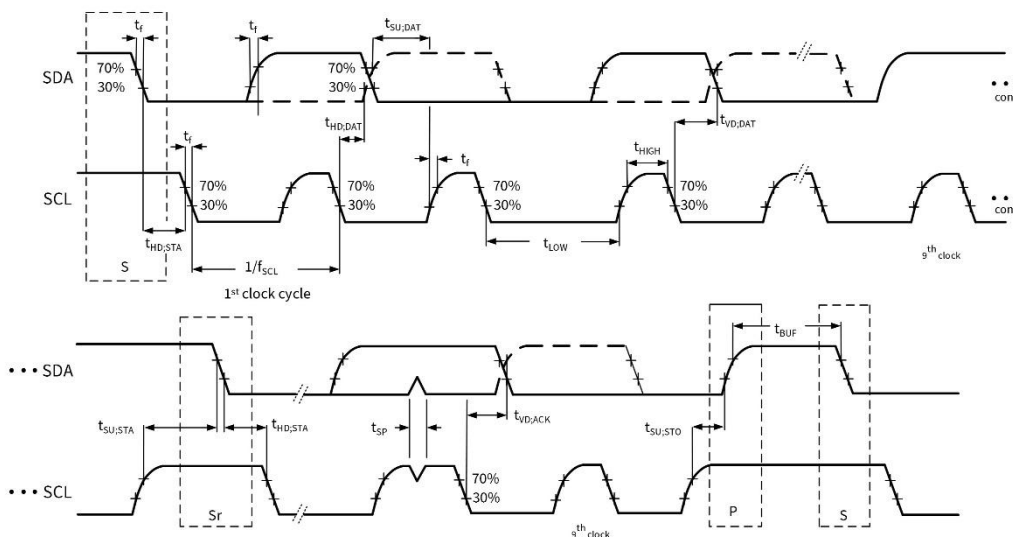


图2 I²C数据时序图

3、I²C 电气特性

参数	符号	MIN	TYP	MAX	单位	备注
时钟频率	Fbsc1b			400	kHz	

SCL 低脉冲	Tblowb	1.3			us	
SCL 高脉冲	Tbhighb	0.6			us	
SDA 设置时间	Tbsudatb	0.1			us	
SDA 保持时间	Tbhddatb	0			us	
重复启动条件的设置时间	Tbsustab	0.6			us	
保持一个开始的时间状况	Tbhdstab	0.6			us	
“停止”的设置时间状况	Tbsustob	0.6			us	
在一个新的传输可以启动之前的时间	Tbbufb	1.3			us	

五、功能描述

1、概述

DWSPD采用MEMS压阻式差压传感器元件作为压力敏感元件，提供与环境压力成比例的原始信号输出。内置的调理IC驱动敏感元件，放大、温度补偿和线性化原始信号，以输出与施加压力成比例的数字或电压信号。

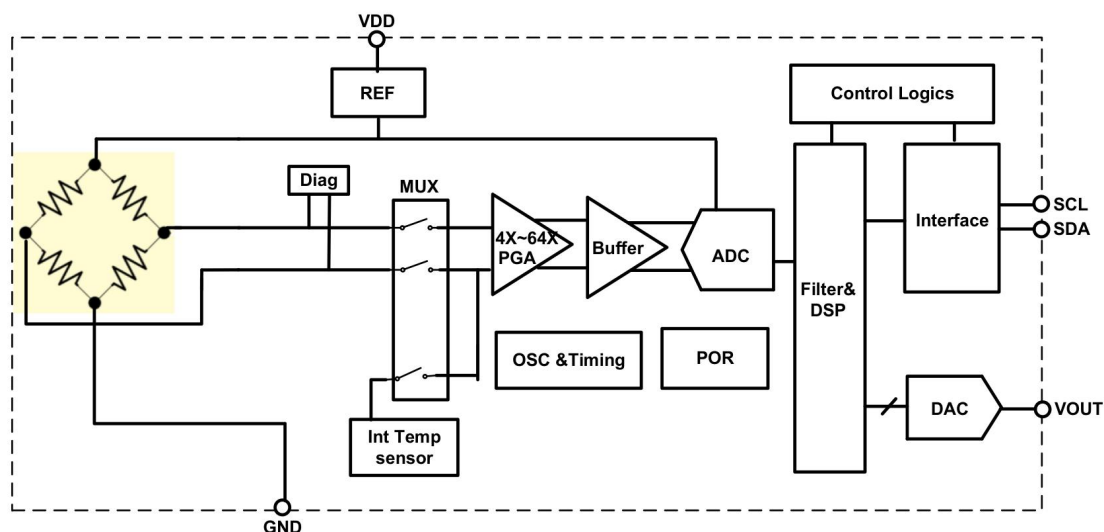


图3 DWSPD系列电路原理图

2、数字输出传递函数

$$P=A*\text{code}/8388608+B$$

Code: 寄存器0x06~0x08的值;

P: 压力值, 压差, 单位为KPa;

产品型号	压力范围(kPa)		输出范围		增益与偏移	
	PL	PH	OL	OH	A	B
DWSPD	-0.5	0.5	838861	7549746	1.250	-0.625

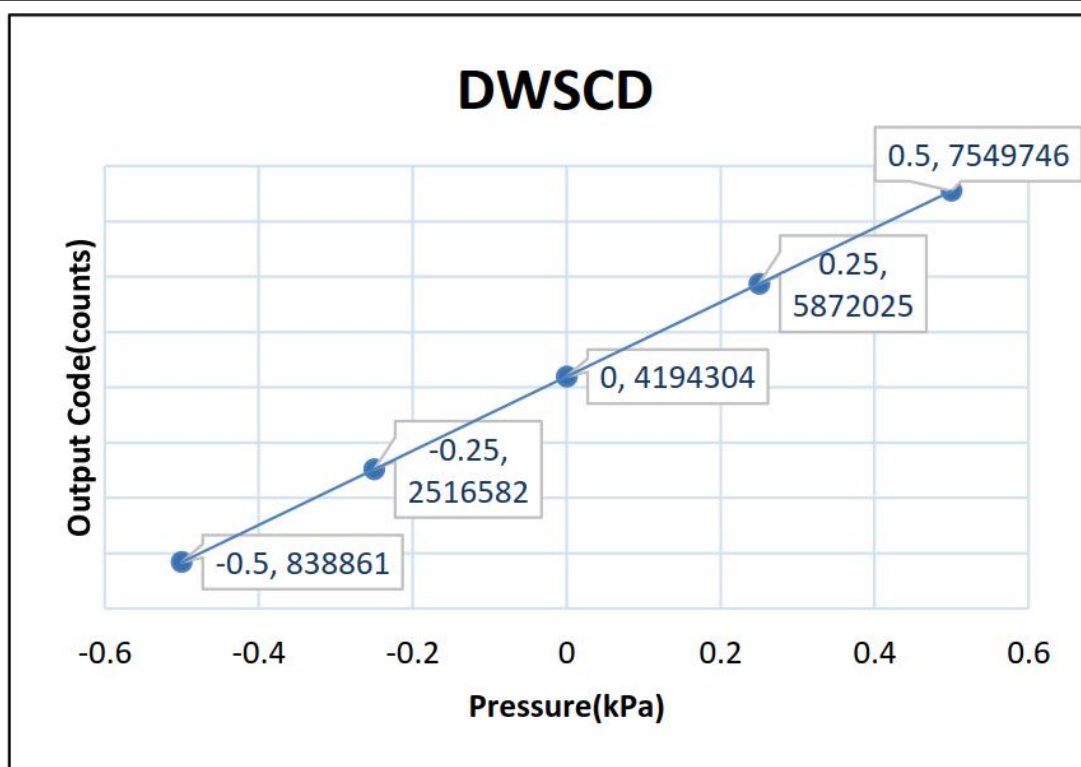


图4 DWSPD数字输出传递函数

3、寄存器

Addr	位添加器	描述	默认值	描述
0x30	7-4	保留	4'b0000	使用 0x0A 写入以开始转换, 在转换结束后自动返回到 0x02。
	3	S	1'b0	
	2-0	测量_ctrl<2:0>	3'b000	
0x06	7-0	PDATA<23:16>	0x00	输出压力数据。 代码=Data0x06*2^16+数据 0x07*2^8+数据 0x08;
0x07	7-0	PDATA<15:8>	0x00	

0x08	7-0	PDATA<7:0>	0x00
------	-----	------------	------

例如:

如果寄存器的值为 0x06、0x07、0x08 为 0x3F、0xFF、0xFF，根传递函数，代码=4194303，P (Pa)
=4194303/8388607*A+B，最终得到 0kPa 左右的压力值。

4、I²C 接口

I²C总线使用SCL和SDA作为信号线。两条线路在外部通过上拉电阻连接到VDD，这样当总线自由时，它们就会被拉高。设备地址如下图所示。

表2 I²C地址位

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

I²C接口协议具有特殊的总线信号条件。启动(S)、停止(P)和二进制数据条件如下所示。

在开始条件下，SCL较高，SDA有一个下降的边缘。然后，将发送从属地址。在这7个地址位之后，方向控制位R/W选择读取或写操作。当一个从设备识别到它正在被处理时，它应该通过在第9个SCL (ACK) 周期中拉低SDA来承认。

在停止条件下，SCL也很高，但SDA有一个上升的边缘。当SCL较高时，数据必须在SDA中保持稳定。只有在SCL值较低时，数据才能改变SDA处的值。

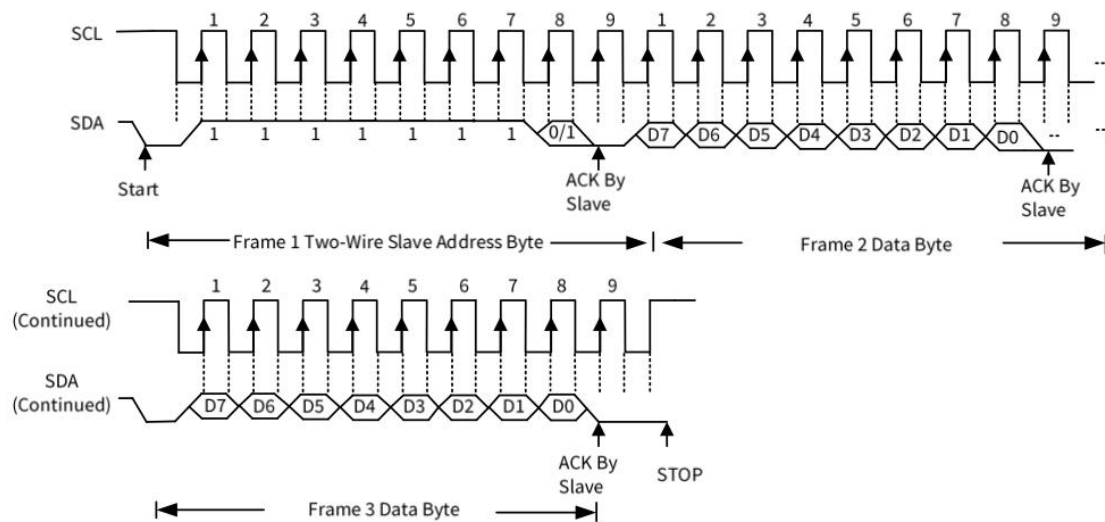


图5 I²C协议

字节写入

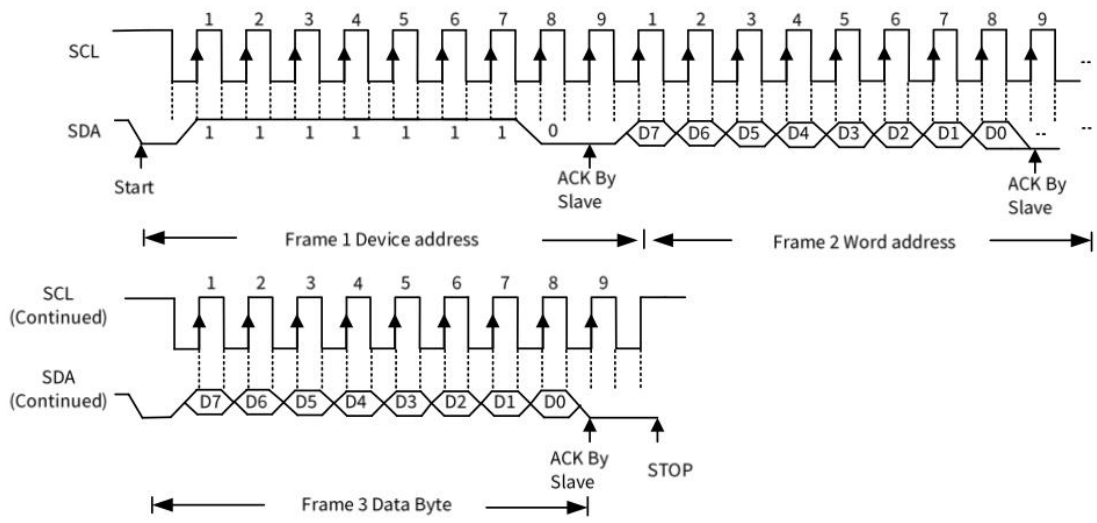


图6 I²C字节写入

随机读取

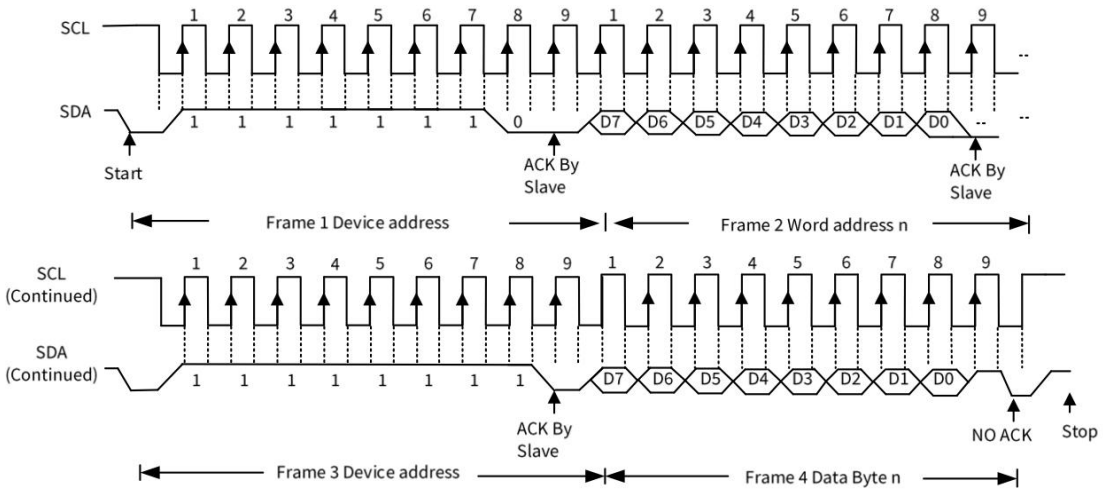


图7 I²C字节读取

六、典型应用电路

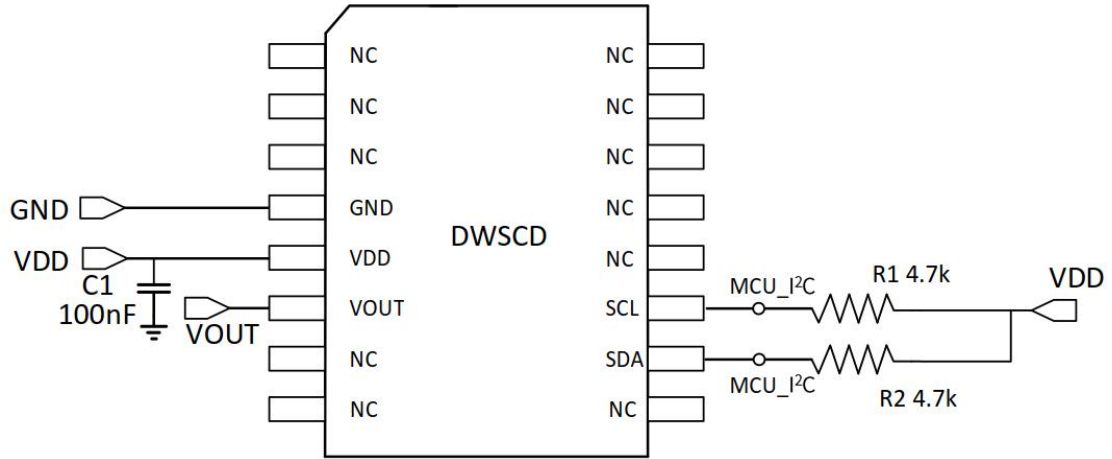


图8 典型应用电路

七、产品尺寸 (mm)

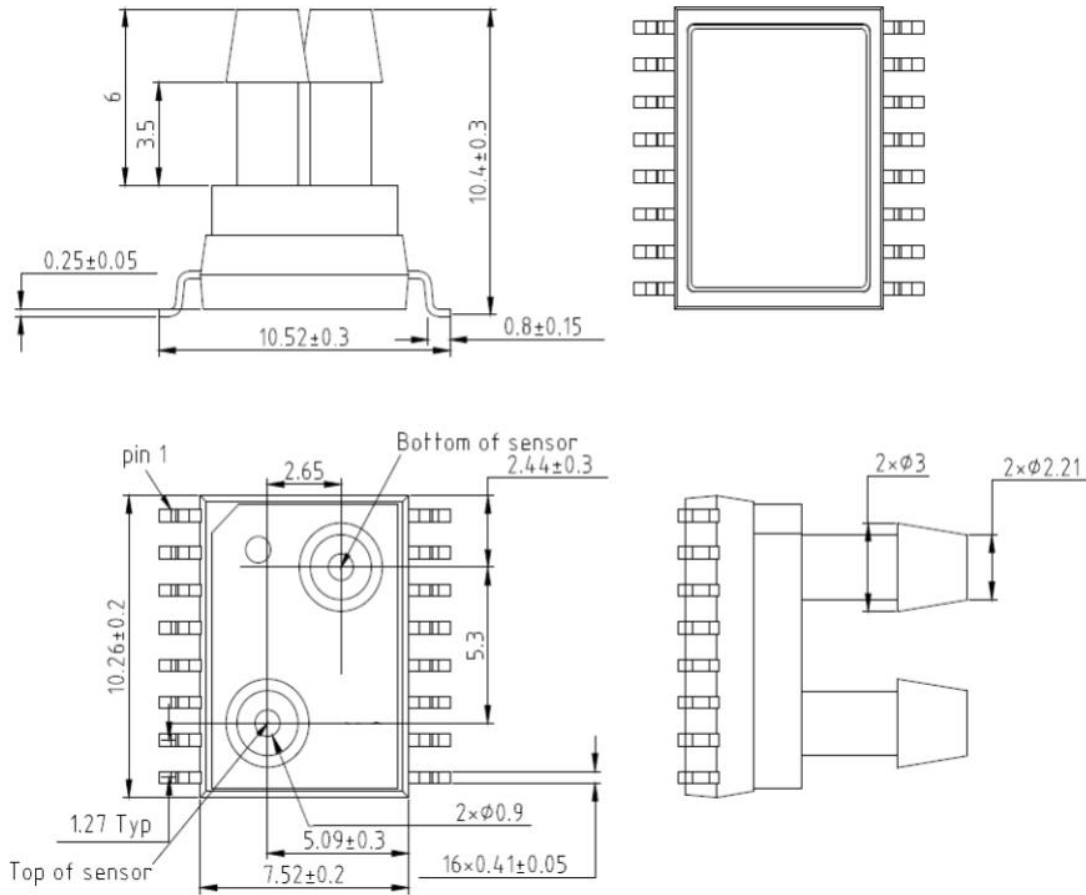


图9 SOIC 16封装外形 (mm)

- 传感器的顶部是连接到传感器芯片顶侧的引压口。顶侧压力为正压力。顶侧的压力增加将导致传感器输出值增加。
- 传感器底部是连接在传感器模具底侧的引压口。底侧压力为负压力。底侧的压力增加将导致传感器的输出值减少。

八、传感器选型

型号	输出类型	压力范围		供电	精度@-20~70°C	
		P _L	P _H		出厂精度	全寿命精度
DWSPD001	I ² C	-500Pa	500Pa	3.3V	±0.8%	±2.0%
DWSPD002	I ² C	-1000Pa	1000Pa	3.3V	±1.0%	±2.0%
DWSPD005	I ² C	-2500Pa	2500Pa	3.3V	±1.0%	±2.0%
DWSPD008	I ² C	-4000Pa	4000Pa	3.3V	±1.0%	±2.0%
DWSPD014	I ² C	-7000Pa	7000Pa	3.3V	±1.0%	±2.0%
DWSPD020	I ² C	-10000Pa	10000Pa	3.3V	±1.0%	±2.0%
DWSPD070	I ² C	-35000Pa	35000Pa	3.3V	±1.0%	±2.0%

型号定义

DW	SPD	001
杜威型号	传感器型号	量程代码

九、焊接参数要求

1、回流焊 (SMD 端子)

回流条件		无铅工艺要求
预热	温度最小值($T_s(\min)$)	150°C
	最大温度($T_s(\max)$)	180°C
	时间(最小至最大值)(t_s)	60-150 秒
平均上升速率液化出相线温度(T_L)至峰值		2°C/最大值
TS(最大值)-TL-提升率		2°C/最大值
软熔焊接	温度(T_L (液化))	210°C
	时间(最小至最大值)(t_L)	60-120 秒
峰值温度(T_P)		240°C
实际峰值温度的持续时间(5°C以内)		12-30 秒
温度降低率		6°C/最大值
25°C至峰值温度(T_P)所需时间		最多 230 秒
焊接温度上限		240°C

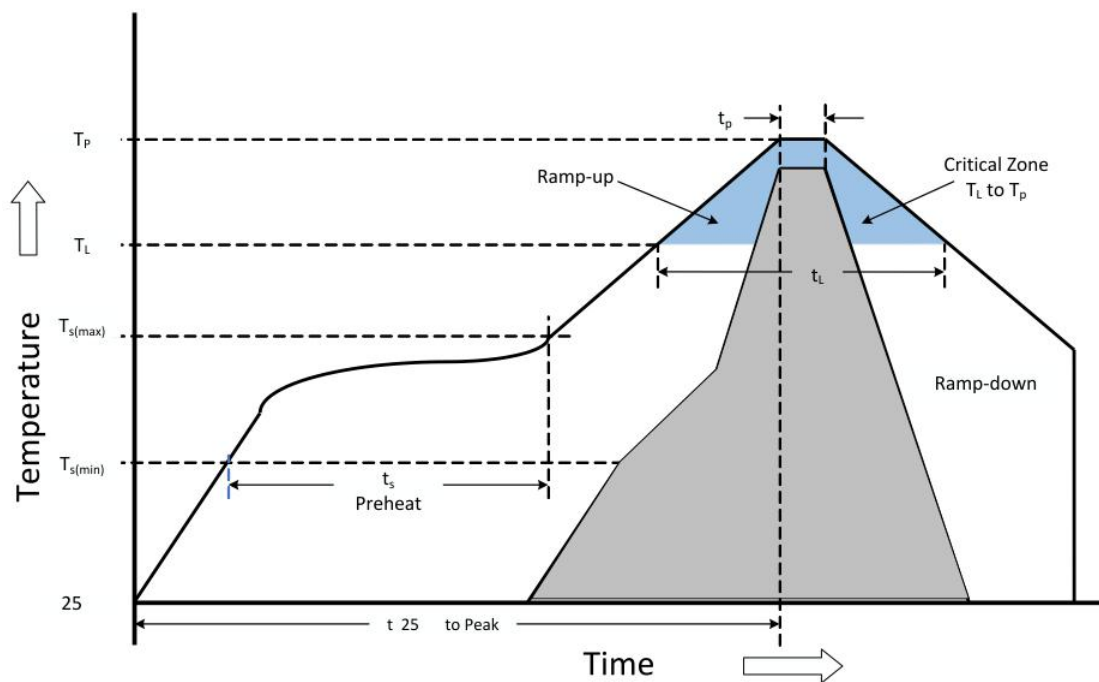
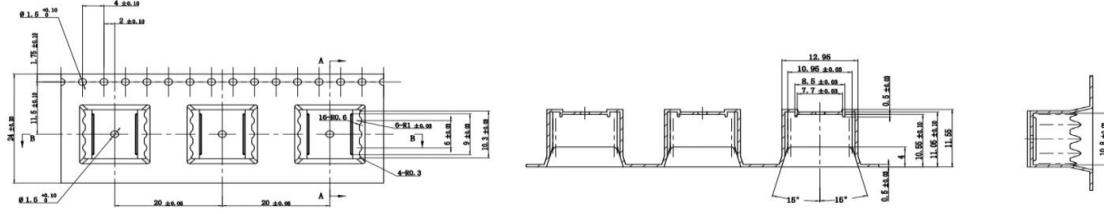


图10 回流焊曲线

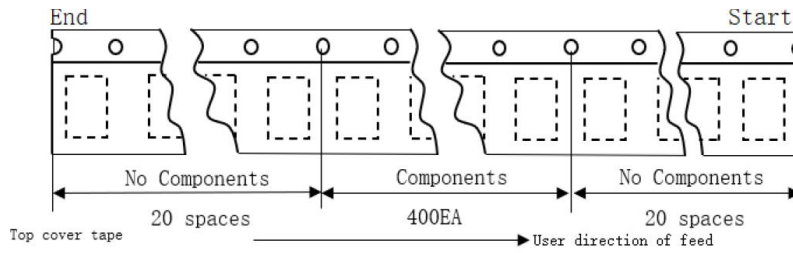
2、手动焊接

- 5秒内将焊头温度提高至260°C~300°C。在焊料桥上进行返工时，使用扁平焊头。
- 一次完成焊接工作。

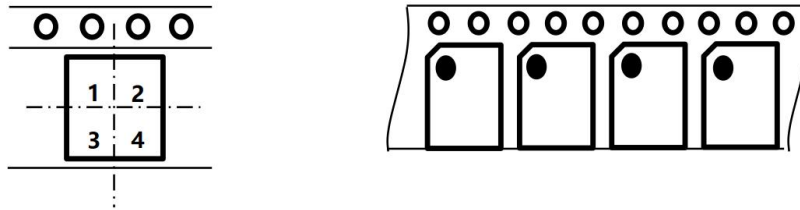
十、包装规格信息



每组盘带的头部和尾部没有组件，间距为20个空盘，如下图所示。



pin1位于第一个象限处，如下图所示。



最小订购量(MOQ): 400颗。

标准包装数量(SPQ): 400颗。

十一、 代码注释

```
#define ACK 1
#define NACK 0
uchar REG06=0,REG07=0,REG08=0;
uchar number=1;
uchar Reg30[1];
int PCode=0,Pdata=0;
float Pressure=0.0;
void IIC_Start(void) //Start the IIC,SDA High-to-low when SCL is high
{
    IIC_SCL(1); //SCL output high level
    SDA_OUT(1); //SDA output high level
    Delay_us(2); //Delay 2us
    SDA_OUT(0); //SDAoutputlowlevel
    Delay_us(2);
}
Void IIC_Stop(void) //StopthePC,SDALow-to-highwhenSCLishigh
{
    IIC_SCL(0);
    Delay_us(2);
    IIC_SCL(1);
    SDA_OUT(0);
    Delay_us(2);
    SDA_OUT(1);
    Delay_us(2);
}
voidIIC_ACK(void) //Send ACK(LOW)
{
    SDA_OUT(0);
    IIC_SCL(1);
    Delay_us(2);
    IIC_SCL(0);
}
Void IIC_NACK(void) //SendNoACK(High)
{
    SDA_OUT(1);
    IIC_SCL(1);
    Delay_us(2);
    IC_SCL(0);
}
```



```
uchar IIC_Wait_ACK(void) //CheckACK,ifreturn0,thenright,ifreturn1,thenerror
{
    Int ErrTime=0;
    SDA_IN(); //SDAsetasinput
    IIC_SCL(1);
    Delay_us(2);
    while(ReadSDA)
    {
        ErrTime++;
        if(ErrTime>200)
        {
            IIC_Stop();
            return1;
        }
        IIC_SCL(0);
        SDA_OUT(0);
        Delay_us(2);
        return0;
    }
}
Void IIC_Send(uchar IIC Data) //Send a byte to I2C
{
    uchar i;
    IIC_SCL(0);
    Delay_us(2);
    for(i=0;i<8;i++)
    {
        if((IIC_Data&0x80)>>7)
            SDA_OUT(1);
        else
            SDA_OUT(0);
        IIC_Data<<=1;
        IIC_SCL(1);
        Delay_us(2);
        IIC_SCL(0);
        Delay_us(2);
    }
}
uchar IIC_Receive(uchar ACK) //Receive a byte from I2C
{
    Uchar i,Receive_Data=0;
    SDA_IN()
    For (i=0;i < 8;i++)
```

```
{
    IIC_SCL(0);
    Delay_us(2);
    IIC_SCL(1);
    ReceiveData<<=1;
    if(Read_SDA==1)
        Receive_Data++;
    Delay_us(2);
}
IIC_SCL(0);
Delay_us(2);
if(ACK==0x01)
    IIC_ACK();
else
    IIC_NACK();
returnReceiveData;
}
void DWSPD_Write_Byte(uchar WriteAddr,uchar WriteData)
{
    IIC_Start();
    IIC_Send(0xFE|0x00);
    IIC_Wait_ACK();
    IIC_Send(WriteAddr);
    IIC_Wait_ACK();
    IIC_Send(WriteData);
    IIC_Wait_ACK();
    IIC_Stop();
}
void DWSPD_Read_Byte(uchar ReadAddr,uchar*pBuffer)
{
    IIC_Start();
    IIC_Send(0xFE|0x00);
    IIC_Wait_ACK();
    IIC_Send(ReadAddr);
    IICWait_ACK();
    IIC_Start();
    IIC_Send(0xFE|0x01);
    IIC_Wait_ACK();
    pBuffer[0]=IICReceive(0);
    IIC_Stop();
}
void DWSPD_Read_3Byte(uchar ReadAddr,uchar*pBuffer)
```

```
{
    IIC_Start();
    IIC_Send(0xFE|0x00);
    IIC_Wait_ACK();
    IIC_Send(ReadAddr);
    IIC_Wait_ACK();
    IIC_Start();
    IIC_Send(0xFE|0x01);
    IIC_Wait_ACK0;
    pBuffer[0]=IIC_Receive(ACK);
    pBuffer[1]=IIC_Receive(ACK);
    pBuffer[2]=IIC_Receive(NACK);
    IIC_Stop();
}
void main()
{
    uchar PData[3]={0,0,0};
    while(1)
    {
        DWSPD_Write_Byte(0x30,0x0A);
        while(1) //Check whether the conversion ends
        {
            if(number<=50)
            {
                number++;
                delay_ms(1);
                DWSPD_Read_Byte(0x30,Reg30);
                if(0x02==Reg30[0])
                {
                    number=1;
                    break;
                }
            }
            if(number>50)
            {
                number=1;
                //User can add his own error handler function
                break;
            }
        }
        DWSPD_Read_3Byte(0x06,PData);
        REG06=PData[0]; //Register 0x06
    }
}
```



DWSPD系列 MEMS微差压传感器

```
REG07=PData[1]; //Register0x07
REG08=PData[2]; //Register0x08
PCode=(REG06*65536+REG07*256+REG08);
//PCode=Data0x06*2^16+
//Data0x07*2^8+Data0x08

if(PCode>8388607)
    Pdata=PCode-16777216; //Symbolprocessing
else
    Pdata=PCode;
Pressure=float(1.25*Pdata/8388607-0.625); //P=A*PCode/8388607+B
//A=1.25,B=-0.625
//PNormalized=PCode/8388

607
    }
}
```