

# MEMS数字乙醇传感器

JED107-001

## 产品说明

威海精讯畅通电子科技有限公司

# 声明

本说明书版权属威海精讯畅通电子科技有限公司（以下称本公司）所有，未经书面许可，本说明书任何部分不得复制、翻译、存储于数据库或检索系统内，也不可以电子、翻拍、录音等任何手段进行传播。

感谢您使用精讯畅通科技的系列产品。为使您更好地使用本公司产品，减少因使用不当造成的产品故障，使用前请务必仔细阅读本说明书并按照所建议的使用方法进行使用。如果您不依照本说明书使用或擅自去除、拆解、更换传感器内部组件，本公司不承担由此造成的任何损失。

您所购买产品的颜色、款式及尺寸以实物为准。

本公司秉承科技进步的理念，不断致力于产品改进和技术创新。因此，本公司保留任何产品改进而不预先通知的权力。使用本说明书时，请确认其属于有效版本。同时，本公司鼓励使用者根据其使用情况，探讨本产品更优化的使用方法。

请妥善保管本说明书，以便在您日后需要时能及时查阅并获得帮助。

威海精讯畅通电子科技有限公司

---

## 一、产品描述

数字型气体传感器是一款低功耗，小型化传感器。采用半导体气体传感器和高性能微处理器相结合，用于检测环境中的气体浓度。灵敏度高，小巧精密，采用 I2C 数字信号输出方式，方便用户使用和调试，大大缩短了用户的设计开发周期。可广泛应用于环境安全，便携仪器等诸多领域。

## 二、传感器特点

本产品采用MEMS工艺，结构坚固，抗震性好，具有尺寸小，低功耗，灵敏度高等优点。

## 三、主要应用

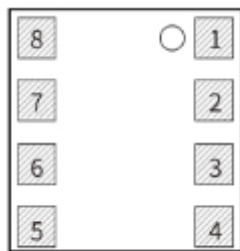
适用于便携仪器、医疗卫生等领域检测乙醇含量。

## 四、产品说明

### 4.1 性能参数

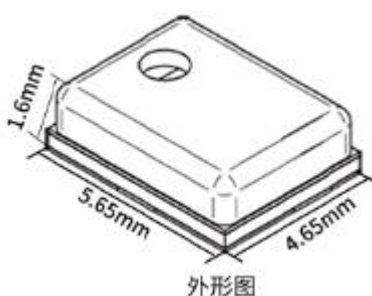
工作电压 (VCC)	3.3V	工作电流	≤25mA
最大加热功率	30mW	加热电压 (VDD)	1.4V
输出方式	I2C 从机模式	默认地址	0x54
I2C 速率	10-100kbps	上拉电阻	需外置上拉电阻
预热时间	30s	响应时间	≤60S
典型精度 (25°C/50%RH)	20% 读数	测量量程	0-500ppm

### 4.2 管脚定义图

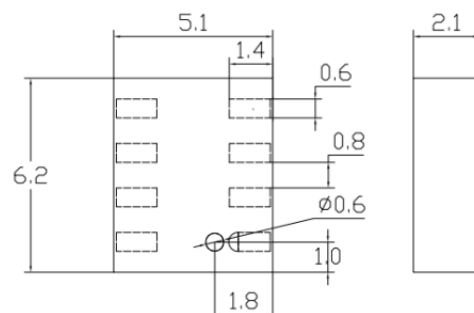


底视底视图

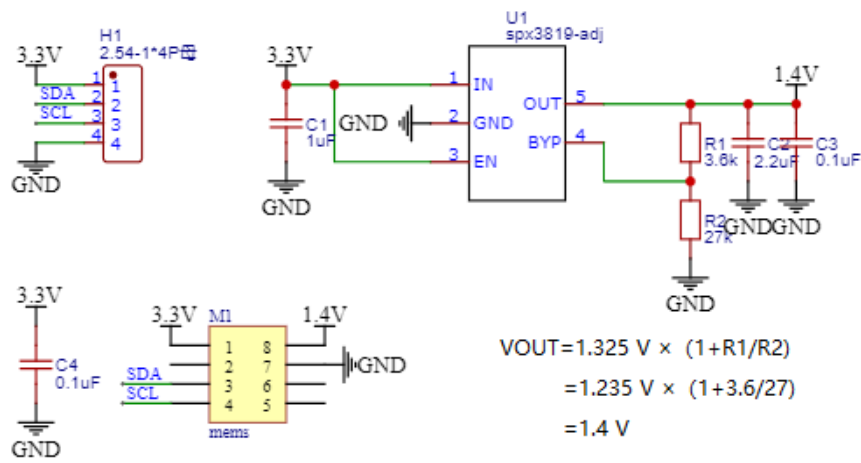
1	VCC	5	NC
2	NC	6	NC
3	SDA	7	GND
4	SCL	8	VDD(加热电压)



外形图

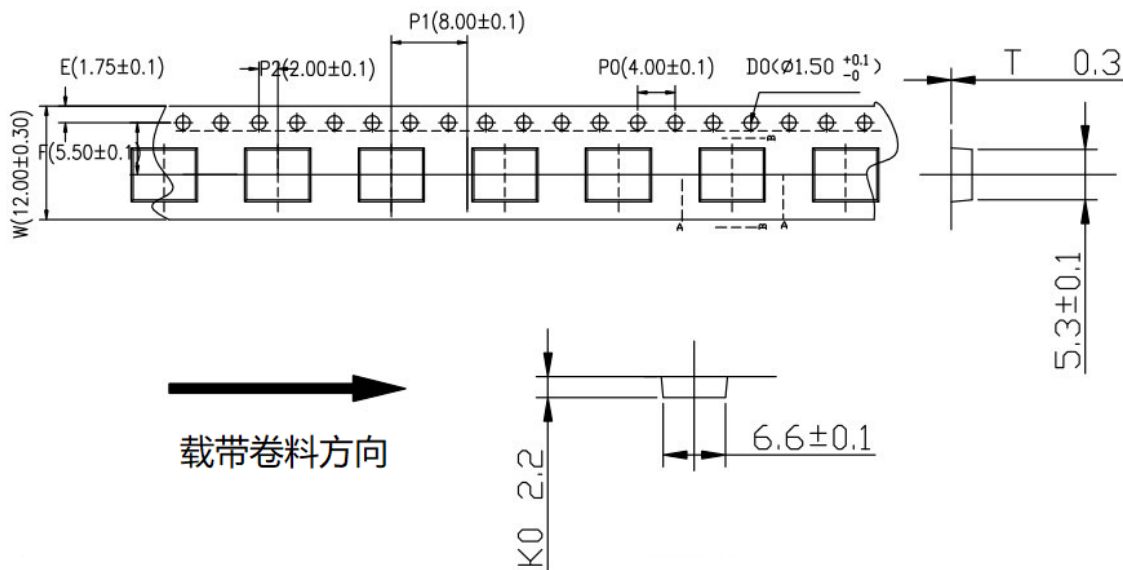


#### 4.3 基本电路

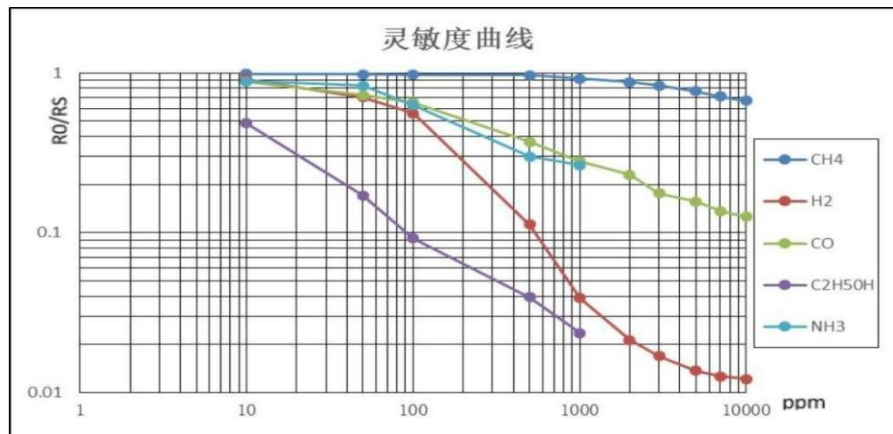


俯视时，芯片上小孔在右上方

#### 4.4 编带尺寸



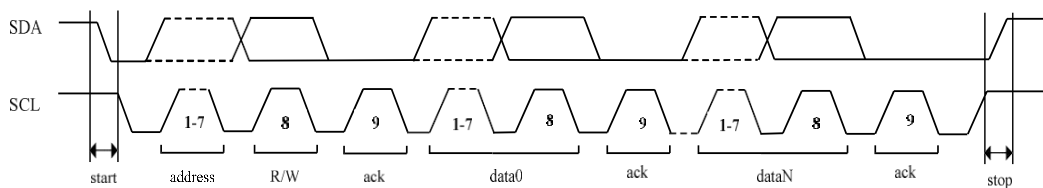
### 五、传感器特性描述



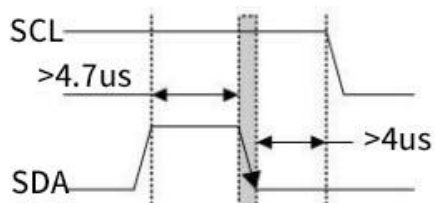
## 六、IIC 通讯协议

### (1) 总线描述

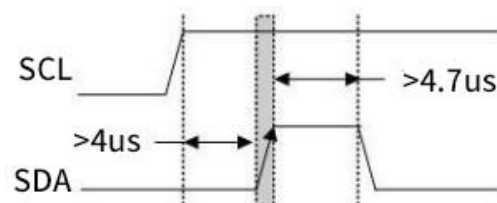
IIC 协议是一个特殊总线信号协议。由 start（开始信号）、stop（结束信号）、二进制数据等三部分组成，如下图。开始时，SCL 高，SDA 下降沿。之后，发送从机地址。在 7 位的地址位之后，是控制读写位，选择读写操作。当从机识别到与其对应的地址信息后，将向主机发送一个应答信号，在第 9 个时钟周期拉低 SDA。在停止时，SCL 保持高电平，SDA 上升沿



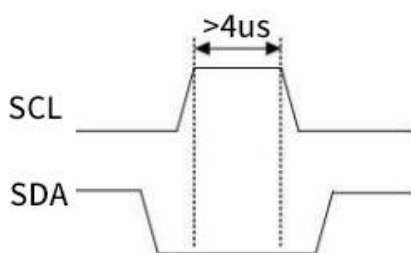
### (2) 典型信号模拟



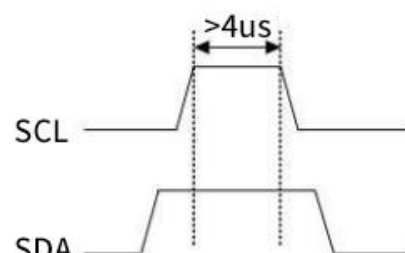
起始信号S



终止信号P



应答/“0”



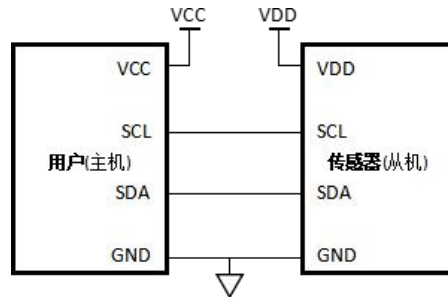
非应答/“1”

### (3) 从机地址

地址字节格式：高7位为模块地址，最低位为读/写操作，1代表读。

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

### (4) 硬件连接



### (5) 数据接口

电源引脚（VCC GND）：供电电压范围为3.3V

串行时钟输入（SCL）：SCL 引脚为 IIC 通信时钟线。

串行数据（SDA）：SDA 引脚为 IIC 数据线，用于读、写数据。

### (6) 数据帧格式

读取：

0	1	2	3	4
0X54	0XA1	0x55	DATA（高位）	DATA（低位）
从机地址(含读写位)	读取命令	从机地址(含读写位)	气体浓度高八位	气体浓度低八位

说明：读取出来的数据为当前气体浓度。 浓度值计算公式：气体浓度=DATA（高位）\*256+DATA（低位）

校零：

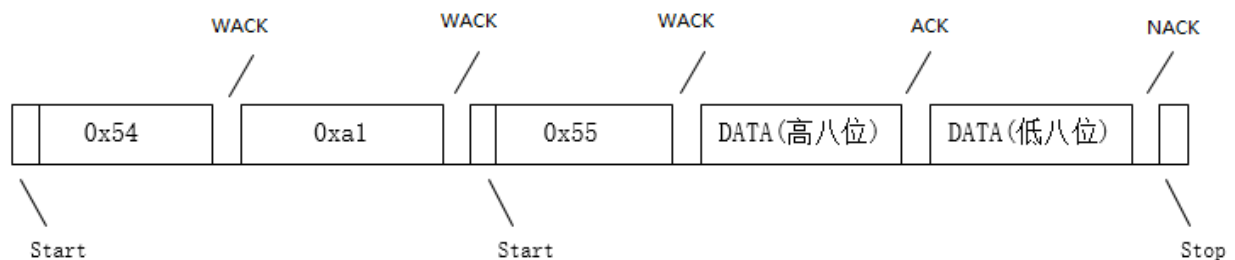
0	1	3	4
0X54	0X32	DATA（高位）	DATA（低位）
从机地址(含读写位)	写入命令	气体浓度高八位	气体浓度低八位

说明：写入的数据为传感器在通风环境空气中的数值，写入后对示数校零。写入后，再读取一次确认是否校零成功。

## 七、使用方法

该传感器上电后需要预热，时长 30 秒左右。预热完成后，进入正常工作状态。接入I2C 总线，主机发送读命令后进入读取状态，传感器会立即返回一个 16 位数据值，该数据表示当前检测的气体浓度值，浓度值计算公式：气体浓度=DATA（高位）\*256+DATA（低位）。

下图是一次I2C 通信过程的读取时序：起始信号 start，发送从机地址 0x54，接收到从机应答信号 ACK，发送读命令 0xA1，接收到从机应答信号ACK；起始信号 start，发送从机地址 0x55，接收到从机应答信号ACK，读取气体浓度高字节，发送到从机应答信号ACK，读取气体浓度低字节，结束信号Stop。



## 八、注意事项

### (1) 预热时间

在不通电情况下长时间贮存，其传感器电阻会产生可逆性漂移，使用前需进行预热以达到其传感器内部的化学平衡，贮存时间及对应的预热时间建议如下：

贮存时间	建议预热时间
1 个月以下	不低于 24 小时
1 - 6 个月	不低于 48 小时
6 个月以上	不低于 72 小时

### (2) 必须避免的情况

#### ① 暴露于可挥发性硅化合物蒸汽中

要避免暴露于硅粘接剂、发胶、硅橡胶、腻子或其它存在可挥发性硅化合物的场所。如果表面吸附了硅化合物蒸汽，传感器的敏感材料会被硅化合物分解形成的二氧化硅包裹，抑制传感器的敏感性，

并且不可恢复。

② 高腐蚀性的环境

暴露在高浓度的腐蚀性气体（如 乙醇，SOX，Cl<sub>2</sub>，HCl 等）中，不仅会引起加热材料及传感器引线的腐蚀或破坏，并会引起敏感材料性能发生不可逆的劣变。

③ 碱、碱金属盐、卤素的污染

该传感器被碱金属尤其是盐水喷雾污染后，或暴露在卤素如乙醇中，也会引起性能劣变。

④ 接触到水

溅上水或浸到水中会造成该传感器敏感特性下降。

⑤ 结冰

水在传感器敏感材料表面结冰会导致敏感层碎裂而丧失敏感特性。

⑥ 施加电压

由过载电压引起的过载加热功率会对传感器造成不可逆的损害，同时静电也会损坏设备，所以在接触设备时要采取防静电措施。

(3) 尽可能避免的情况

① 凝结水

在室内使用条件下，轻微凝结水对传感器性能会产生轻微影响。但是，如果水凝结在敏感层表面并保持一段时间，传感器特性则会下降。

② 处于高浓度气体中

无论是否通电，在高浓度气体中长期放置，均会影响其特性。如用打火机气直接喷向该传感器，会对其造成极大损害。

长期暴露在极端环境中无论是否通电，长时间暴露在极端条件下，如高湿、高温或高污染等极端条件，传感器性能将受到严重影响。

③ 振动

频繁、过度振动会导致该传感器内部产生共振而断裂。在运输途中及组装线上使用气动改锥/超声波焊接机会产生此类振动。

④ 冲击

如果传感器受到强烈冲击或碰撞会导致其内部断裂。

⑤ 焊接

- a. 回流焊接建议条件中性气氛；焊接温度  $250 \pm 10^{\circ}\text{C}$ ；避免助焊剂蒸汽。
- b. 手工焊接建议条件含氯最少的松香助焊剂；焊接温度  $\leq 350^{\circ}\text{C}$ ；持续时间  $\leq 3\text{s}$ 。

■ 注：违反以上使用条件将使该传感器特性下降。



## 附录：主机IIC程序

### (1) 读取浓度：

```
static uint16_t IIC_Read_Byte(u8 reg)
{
    uint8_t rec_data;
    I2C_Start();
    I2C_SendByte(0x54);
    I2C_WaitAck();
    I2C_SendByte(0xe1);
    I2C_WaitAck();
    I2C_Start();
    I2C_SendByte(0x55);
    I2C_WaitAck();
    rec_data = I2C_RecvByte();
    rec_data16=rec_data;
    I2C_SendACK(0);
    rec_data = I2C_RecvByte();
    rec_data16=rec_data16<<8|rec_data;
    I2C_Stop();
    return rec_data16;
}

/*起始信号*/
u8 I2C_Start(void)
{
    SDA1H;
    I2C_delay();
    SCL1H;
    I2C_delay();
    if(!SDAread)
        return 0;
    SDA1L;
    I2C_delay();
    if(SDAread)
        return 0;
    SCL1L;
    I2C_delay();
    return 1;
}

/*发送一个字节*/
void I2C_SendByte(u8 SendByte) //数据从高位到低位//
{
    u8 i=8;
    SCL1L;
    for (i=0; i<8; i++) //8位计数器
    {
        if(SendByte&0x80)//SDA准备
            SDA1H;
        else
            SDA1L;
        SCL1H; //拉高时钟，给从机采样
    }
}
```

```

    I2C_delay();          //延时保持IIC时钟频率，也是给从机采样有充足时间
    SCL1L;                //拉低时钟，给SDA准备
    I2C_delay();          //延时保持IIC时钟频率
    SendByte<<=1;         //移出数据的最高位
}
}
/*等待ACK*/
u8 I2C_WaitAck(void) //返回为:=1有ACK,=0无ACK
{
    uint16_t i=0;
    SDA1H;                //释放SDA
    SCL1H;                //SCL拉高进行采样
    while(SDAread)//等待SDA拉低
    {
        i++;              //等待计数
        if(i==1000)//超时跳出循环
            break;
    }
    if(SDAread)//再次判断SDA是否拉低
    {
        SCL1L;
        return 0;//从机应答失败，返回0
    }
    I2C_delay();//延时保证时钟频率低于40K，
    SCL1L;
    I2C_delay();//延时保证时钟频率低于40K，
    return 1;//从机应答成功，返回1
}
/*接受八位数据*/
u8 I2C_RecvByte(void) //数据从高位到低位//
{
    u8 i=8;
    u8 ReceiveByte=0;
    SDA1H; //释放SDA，给从机使用
    I2C_delay();//延时给从机准备SDA时间
    for (i=0; i<8; i++)          //8位计数器
    {
        ReceiveByte <<= 1;
        SCL1H;                  //拉高时钟线，采样从机SDA
        if(SDAread) //读数据
            ReceiveByte |=0x01;
        I2C_delay();           //延时保持IIC时钟频率
        SCL1L;                 //拉低时钟线，处理接收到的数据
        I2C_delay();           //延时给从机准备SDA时间
    }
    return ReceiveByte;
}

```

```
/*结束信号*/
void I2C_Stop(void)
{
    SCL1L;
    I2C_delay();
    SDA1L;
    I2C_delay();
    SCL1H;
    I2C_delay();
    SDA1H;
    I2C_delay();
}
/*IIC等待延时*/
void I2C_delay(void)
{
    int x=5;
    u8 i=100;
    x=i*x;
    while(x--);
}
void I2C_SendACK(u8 i)
{
    if(i==1)
        SDA1H;
    else
        SDA1L;
    SCL1H;
    I2C_delay();
    SCL1L;
    I2C_delay();
}
```