

RS-ZS-BZ-* 噪声板载模块 使用说明书 (I2C)

文档版本: V1.2





声明

1. 本说明书版权属山东仁科测控技术有限公司（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。
2. 感谢您使用山东仁科的系列产品。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。如果用户不依照本说明书使用或擅自去除、拆解、更换设备内部组件，本公司不承担由此造成的任何损失。
3. 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利。使用本说明书时，请确认其属于有效版本。
4. 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。

山东仁科测控技术有限公司



1. 产品介绍	4
1.1 产品概述	4
1.2 功能特点	4
1.3 主要技术指标	4
1.4 产品选型	6
2. 硬件说明	6
2.1 设备定义	6
2.2 设备尺寸	7
2.3 模块 PCB 封装	7
3. 设备通信	8
3.1 I ² C 序列	8
3.2 I ² C 命令	8
3.3 校验和计算	9
3.4 通讯例程	10
4. 注意事项	13
5. 质保声明	13
6. 联系方式	14
7. 文档历史	14



1. 产品介绍

1.1 产品概述

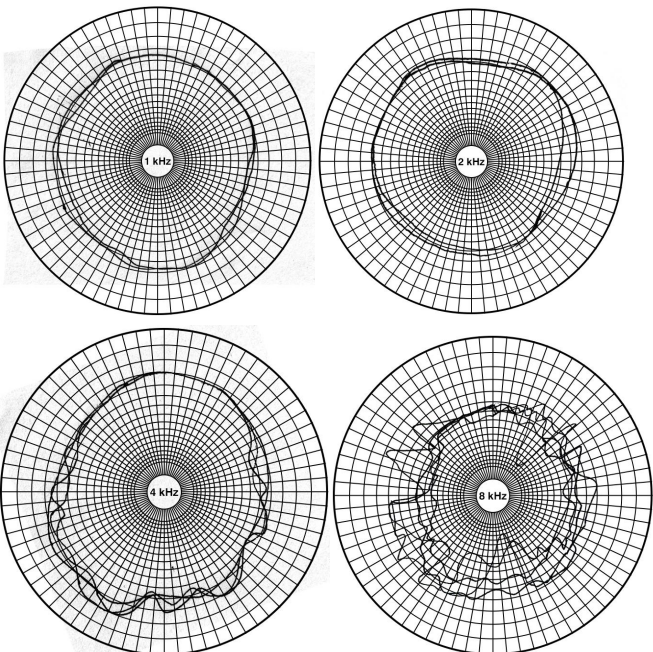
RS-ZS-BZ-*噪声板载模块主要用于环境噪声、交通噪声、作业场所噪声、建筑施工噪声和社会生活噪声等各类噪声的现场实时测量。有此款模块，客户不需要再为复杂的噪声信号处理而烦恼，可专注于擅长的领域，更快速的为客户创造价值。

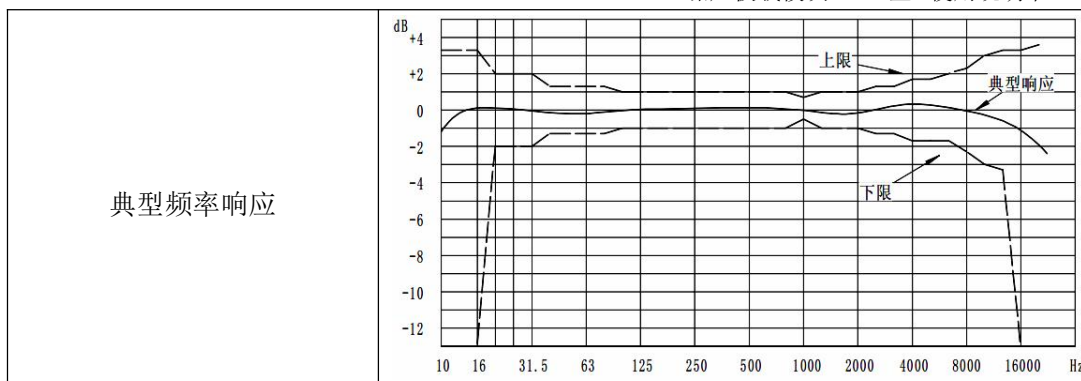
1.2 功能特点

- PCB 板载安装方式。
- 30~130dB 宽量程、20~12.5kHz 宽频率测量。
- 采用高性能预极化背极驻极体电容传声器，动态范围宽、性能稳定。
- 输出接口 I²C。
- 内置 25k Ω ~50k Ω 弱上拉电阻（典型值 40k Ω ）。
- 具有慢速和快速测量两种模式，满足不同的客户的要求。
- 供电电源 DC4.5V~5.5V、DC10~28V 可选。

1.3 主要技术指标

工作电压	DC4.5~5.5V（默认） DC10~28V（可选）	
功耗	18.9mA@5V 31.0mA@12V 27.8mA@24V	
模块电路工作温湿度	-20℃~+60℃，0%RH~90%RH（非结露）	
输出信号	I ² C	
I/O 数字逻辑特性	I/O 输入高电平电压	2~3.3V
	I/O 输入低电平电压	0~0.8V
	I/O 输出高电平电压	2.4~3.3V
	I/O 输出低电平电压	0~0.4V
测量范围	30dB~130dB	
频率计权	A 计权	
频率响应范围	20Hz~12.5kHz	
响应时间	快速模式	500ms
	慢速模式	1.5s
稳定性	使用周期内小于 2%	
参考校准点	94dB 校准，基准声压 20uPa,频率 1kHz	
噪声精度	±0.5dB（在参考音准，94dB@1kHz）	

防尘罩影响	在 50~115dB 范围内影响≤0.5dB 量程内其他范围内影响≤0.7dB																																				
工作环境	空气温度：-20~+60℃ 静压：65kPa~106kPa 周围无强烈的机械振动、冲击、强电磁场和腐蚀性气体存在																																				
声压响应与自由场响应的调整数据	通过声校准器产生的声压响应或用静电激励器产生的模拟声压响应，可通过下表的调整数据得到等效的自由场响应。 <table><tr><td>频率 / kHz</td><td>自由场增量 / dB</td><td>频率 / kHz</td><td>自由场增量 / dB</td></tr><tr><td>1</td><td>0.2</td><td>6.3</td><td>2.2</td></tr><tr><td>1.25</td><td>0.3</td><td>8</td><td>3.4</td></tr><tr><td>1.6</td><td>0.4</td><td>10</td><td>5.0</td></tr><tr><td>2</td><td>0.5</td><td>12.5</td><td>6.2</td></tr><tr><td>2.5</td><td>0.7</td><td>16</td><td>7.6</td></tr><tr><td>3.15</td><td>0.9</td><td>18</td><td>8.4</td></tr><tr><td>4</td><td>1.3</td><td>20</td><td>9.0</td></tr><tr><td>5</td><td>1.8</td><td>—</td><td>—</td></tr></table>	频率 / kHz	自由场增量 / dB	频率 / kHz	自由场增量 / dB	1	0.2	6.3	2.2	1.25	0.3	8	3.4	1.6	0.4	10	5.0	2	0.5	12.5	6.2	2.5	0.7	16	7.6	3.15	0.9	18	8.4	4	1.3	20	9.0	5	1.8	—	—
频率 / kHz	自由场增量 / dB	频率 / kHz	自由场增量 / dB																																		
1	0.2	6.3	2.2																																		
1.25	0.3	8	3.4																																		
1.6	0.4	10	5.0																																		
2	0.5	12.5	6.2																																		
2.5	0.7	16	7.6																																		
3.15	0.9	18	8.4																																		
4	1.3	20	9.0																																		
5	1.8	—	—																																		
模块在 1kHz、2kHz、4kHz、8kHz 的指向性																																					
驻极体参数																																					
自由场灵敏度	约为 8mV/Pa(自由场灵敏度级为-42 dB，以 1V 为参考)																																				
驻极体极头电容量	约为 15pF																																				

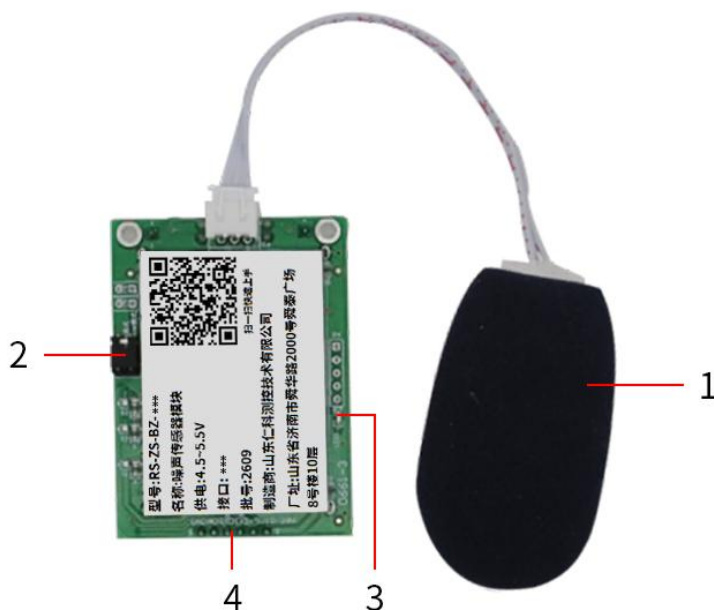


1.4 产品选型

RS-					公司代号
	ZS-				噪声模块
		BZ-			PCB 安装方式
			IIC-		I ² C 输出
				05	供电电压为 DC4.5~5.5V
				12	供电电压为 DC10~28V

2. 硬件说明

2.1 设备定义

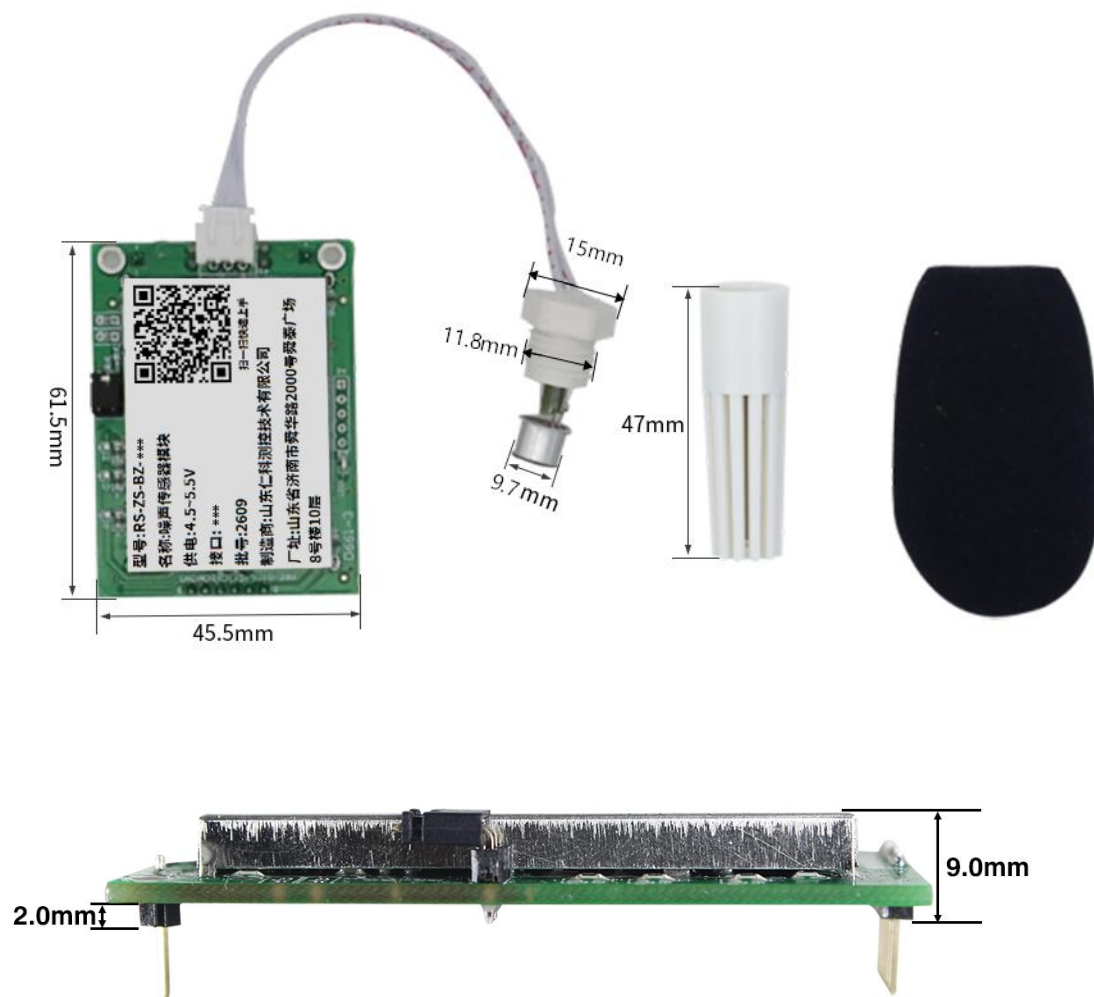


序号	名称	说明
1	声音探头	内含驻极体麦克风、塑料护套、防尘棉
2	模式选择端子	跳线帽短接则为 fast 模式，噪声更新时间为 500ms；不短接则为 slow 模式，噪声更新时间为 1.5s
3	运行指示灯	运行指示灯，1s 闪烁一次
4	管脚序号	名称说明

	1	10-28V	DC10-28V 电源输入 (选型供电电压为 DC10~28V 使用)
	2	+5V	DC5V 电源输入 (选型供电电压为 DC4.5~5.5V 使用)
	3	TXD/A	SCLK
	4	RXD/B	DATA
	5	AO	预留
	6	GND	电源负极

【注意】设备已内置 25kΩ~50kΩ 弱上拉电阻 (典型值 40kΩ)，仍建议在使用时在通讯线上增加 10kΩ 外置上拉电阻。

2.2 设备尺寸

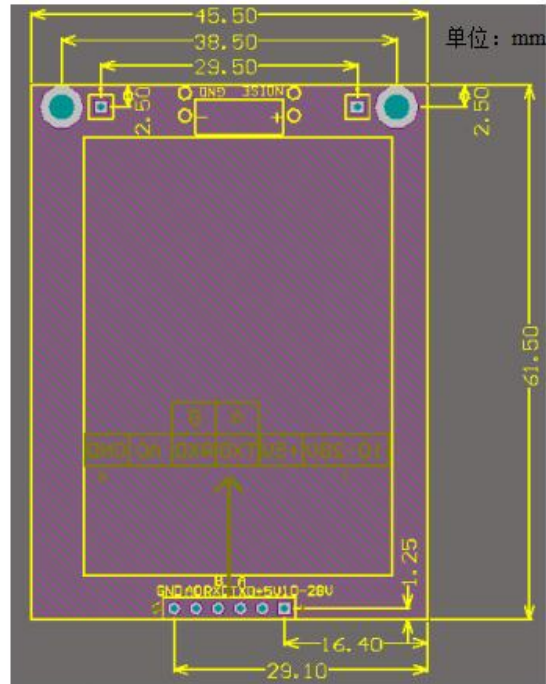


注：声音探头引线长度 10cm,若要夹住声音探头推荐面板开孔尺寸为 $\phi 13\text{mm}$ 。

2.3 模块 PCB 封装

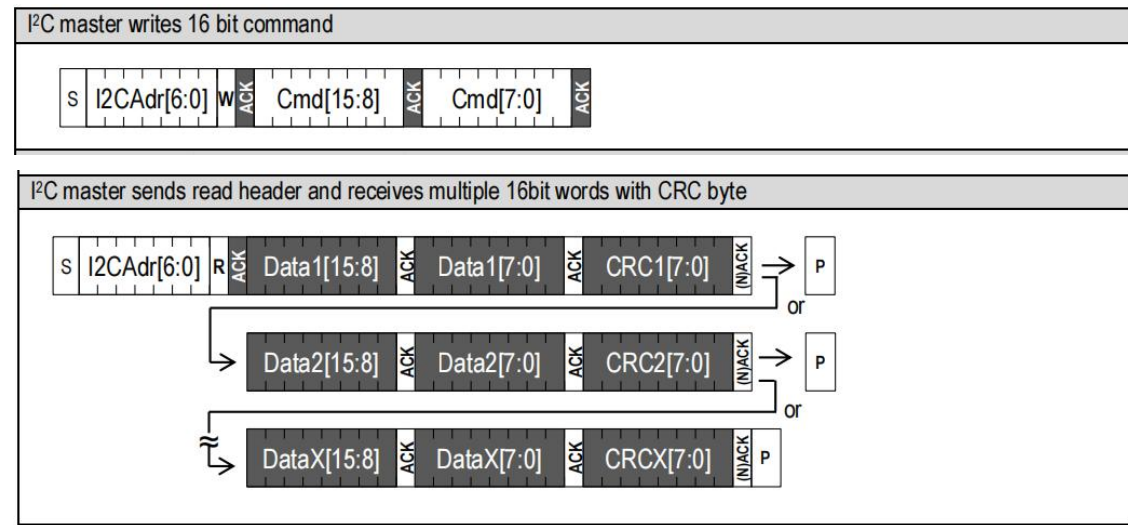
尺寸图如下：

Altium Designer 封装文件请下载资料包。



3. 设备通信

3.1 I²C 序列



3.2 I²C 命令

设备地址

设备地址为 0x10

读取命令

(先写入命令(确定读取哪个数据), 再读取数据(支持连读, 最后发 NACK 结束))

命令	命令代码 (Hex)	返回长度	描述
读取噪声	0x0000	3 (噪声值高位+噪声值低位+CRC)	读取噪声值 (扩大 10 倍)
读取版本	0x0009	3 (版本高位+版本低位+CRC)	读取版本号

3.3 校验和计算

校验和字节由具有以下属性的 CRC 算法生成：

项目	内容
名字	CRC-8
受保护数据	读取数据
长度	8 位
多项式	$0x31(x^8+x^5+x^4+1)$
初始化	0xFF
举例	CRC (0xBEEF) = 0x92

例：

```
unsigned char GenerateCRC8(unsigned char *data, unsigned char count)
{
    unsigned char crc = 0xFF;
    unsigned char current_byte;
    unsigned char crc_bit;

    /*calculates 8-Bit checksum with given polynomial*/
    for (current_byte = 0; current_byte < count; current_byte++)
    {
        crc ^= (data[current_byte]);
        for (crc_bit = 8; crc_bit > 0; --crc_bit)
        {
            if (crc & 0x80)
                crc = (crc << 1) ^ POLYNOMIAL;    //POLYNOMIAL = 0x31;    //P(x)=x^8+x^5+x^4+1 = 100110001
            else
                crc = (crc << 1);
        }
    }
    return(crc);
}
```

3.4 通讯例程

```
/*IIC通讯示例*/

//根据实际使用的DATA、CLK线来设置拉高拉低
#define IIC_DAT_SET(dat) (dat?(P14 = 1):(P14 = 0))//操作数据线
#define IIC_CLK_SET(dat) (dat?(P13 = 1):(P13 = 0))//操作时钟线
#define IIC_DAT_READ() ((P14)?SET:RESET)//读数据线
#define IIC_CLK_READ() ((P13)?SET:RESET)//读时钟线

#define FALSE 0x00
#define TRUE 0x01
#define RESET 0x00
#define SET 0x01

#define NO_ERROR 0x00 // no error

#define ACK 0x00
#define NACK 0x01

typedef unsigned char INT8U;
typedef unsigned short INT16U;

const INT16U code POLYNOMIAL = 0x131; //P(x)=x^8+x^5+x^4+1 = 100110001

static INT8U IIC_CheckCrc(INT8U *tempdata, INT8U nbrOfBytes, INT8U checksum);
static INT8U WriteCommand(u16 command);

void I2CDelay (INT16U t) //t=1约1.5us
{
    while(t-->0)
    {
        _nop_0;
        _nop_0;
        _nop_0;
        _nop_0;
        _nop_0;
        _nop_0;
        _nop_0;
        _nop_0;
        _nop_0;
    }
}

//IIC起始信号
void I2CStart(void)
{
    IIC_DAT_SET(SET);
    IIC_CLK_SET(SET);
    I2CDelay(20);
    IIC_DAT_SET(RESET);
    I2CDelay(20);
    IIC_CLK_SET(RESET);
    I2CDelay(2);
}

//IIC停止信号
void I2CStop(void)
{
    IIC_DAT_SET(RESET);
    IIC_CLK_SET(RESET);
    I2CDelay(20);
    IIC_CLK_SET(SET);
    I2CDelay(20);
    IIC_DAT_SET(SET);
    I2CDelay(2);
}
```



INT8U I2C_Write_Byte(INT8U Write_Byte)

```
{
    INT8U i;
    for(i=0; i<8; i++)        //要传送的数据长度为8位
    {
        if(Write_Byte&0x80)    //判断发送位
        {
            IIC_DAT_SET(SET);
        }
        else
        {
            IIC_DAT_SET(RESET);
        }
        I2CDelay(5);
        IIC_CLK_SET(SET);      //输出SDA稳定后，拉高SCL给出上升沿，从机检测到后进行数据采样
        I2CDelay(20);
        IIC_CLK_SET(RESET);
        I2CDelay(5);
        Write_Byte <<= 1;
    }
    I2CDelay(20);
    IIC_DAT_SET(SET);          //8位发送完后释放数据线，准备接收应答位
    IIC_CLK_SET(SET);          //等待从机的应答信号
    I2CDelay(60);
    /*以下是判断I2C总线接收应到应答信号是ACK还是NACK*/
    if(IIC_DAT_READ()==SET)    //SDA为高，收到NACK
    {
        IIC_CLK_SET(RESET);
        I2CDelay(7);
        return NACK;
    }
    else
    {
        IIC_CLK_SET(RESET);
        I2CDelay(7);
        return ACK;
    }
}
```

INT8U I2C_Read_Byte(INT8U AckValue)

```
{
    INT8U i,RDByte=0;
    IIC_CLK_SET(RESET);        //置时钟线为低，准备接收数据位
    IIC_DAT_SET(SET);          //释放总线,置数据线为输入方式
    I2CDelay(30);
    for (i=0; i<8; i++)
    {
        IIC_CLK_SET(SET);      //SCL高电平期间，采集SDA信号，并作为有效数据 //置时钟线为高使数据线上数据有效
        I2CDelay(10);
        RDByte <<= 1;          //移位
        if(SET == IIC_DAT_READ())    //采样获取数据
        {
            RDByte |= 0x01;
        }
        else
        {
            RDByte &= 0xfe;
        }
        I2CDelay(5);
        IIC_CLK_SET(RESET);    //下降沿，从机给出下一位值
        I2CDelay(30);
    }
    /*以下是I2C总线发送应答信号ACK或者NACK*/
    IIC_DAT_SET((BitAction)AckValue);    //应答状态
    I2CDelay(15);
    IIC_CLK_SET(SET);
    I2CDelay(25);
    IIC_CLK_SET(RESET);        //清时钟线，钳住I2C总线以便继续接收
    I2CDelay(7);
    return RDByte;
}
```



```
//crc校验函数
static INT8U IIC_CheckCrc(INT8U *tempdata, INT8U nbrOfBytes, INT8U checksum)
{
    INT8U crc = 0xFF;
    INT8U byteCtr;
    INT8U i;
    for (byteCtr = 0; byteCtr < nbrOfBytes; byteCtr++)
    {
        crc ^= (tempdata[byteCtr]);
        for (i = 8; i > 0; --i)
        {
            if (crc & 0x80)
                crc = (u8)((crc << 1) ^ POLYNOMIAL);
            else
                crc = (u8)(crc << 1);
        }
    }
    if (crc == checksum)
    {
        return(TRUE);
    }
    else
    {
        return(FALSE);
    }
}

static INT8U IIC_WriteCommand(u16 command)
{
    u8 error;
    error = I2C_Write_Byte((u8)(command>>8));
    error |= I2C_Write_Byte((u8)(command&0xFF));
    return error;
}

unsigned char IICdata[3];
unsigned short wVersion = 0;

void IICReadTest(void)
{
    unsigned char i = 0;
    unsigned short wtempdata = 0;

    I2CStart(); //发送起始信号
    if(ACK == I2C_Write_Byte((0x10<<1) | 0x00)) //写入地址和写指令
    {
        if (NO_ERROR == IIC_WriteCommand(0x0009))//写入读版本号命令(读取噪声数值时为0x0000)
        {
            I2CStart();
            if (ACK == I2C_Write_Byte((0x10<<1) | 0x01)) //写入地址和读指令
            {
                IICdata[0] = I2C_Read_Byte(ACK); //版本号高字节
                IICdata[1] = I2C_Read_Byte(ACK); //版本号低字节
                IICdata[2] = I2C_Read_Byte(NACK); //CRC
                if (IIC_CheckCrc(IICdata,2,IICdata[2])) //校验
                {
                    wtempdata = IICdata[0] << 8;
                    wtempdata |= IICdata[1];
                    wVersion = wtempdata;
                }
            }
        }
    }
    I2CStop(); //停止IIC
}
```



4. 注意事项

1) 警告：人身伤害风险。本设备严禁用作安全装置或紧急停止装置，亦不得用于可能因设备故障导致人身伤害的其他用途。使用限制：仅限按预期授权用途使用。安装、操作或维修前必须查阅技术手册。未遵守上述指引可能导致死亡或严重伤害。

2) 用户不得自行拆卸，更不能触碰传感器芯体，以免造成产品的损坏。

3) 尽量远离大功率干扰设备，以免造成测量的不准确，如变频器、电机等，安装、拆卸变送器时必须先断开电源，禁止变送器内有水进入可导致不可逆变化。

4) 防止化学试剂、油、粉尘等直接侵害传感器，勿在结露、极限温度环境下长期使用、严防冷热冲击。

5. 质保声明

保修期限自购买日起 24 月内（以有效购买凭证为准），保修设备在保修期间，正常使用和维护的情况下，设备本身机件材料及工艺出现问题，发生故障，经查验属实，本公司将提供免费修理及更换零件。

超出质保期，终身提供维修服务。

符合以下情况之一则不在质保范围内：

- 1.产品因错误安装，操作而导致设备损坏。
- 2.曾经由非本公司的技术人员拆卸、修理、改动、改装或用户自行更换设备内任何部件。
- 3.疏忽使用或被水、其他物质掺入设备内造成损坏。
- 4.意外事件自然灾害导致的故障或损坏。
- 5.超出产品参数中列出的工作参数范围导致的故障或损坏。



6. 联系方式

山东仁科测控技术有限公司

营销中心：山东省济南市高新区舜泰广场 8 号楼东座 10 楼整层

邮编：250101

电话：400-085-5807

传真：（86）0531-67805165

网址：www.rkckth.com

云平台地址：www.0531yun.com



山东仁科测控技术有限公司 [官网](http://www.rkckth.com)



欢迎关注微信公众平台，智享便捷服务

7. 文档历史

- V1.0 文档建立。
- V1.1 修正使用环境。
- V1.2 更改通讯例程