

QZKJ800-PID-AD04

气体传感器智能模组

使用手册

深圳市前兆科技有限公司

一. 概述

AD04 气体传感器智能模组是专门针对气体分析仪表行业研发，可以完成 PID，电化学，红外等 4 系传感器模拟量输出的线性标定，同时内置各种传感器的响应曲线数据，实现在单点标定或多点标定情况下保证浓度转换输出的准确性。主要解决仪表安装现场没有标准气体无法标定，或者客户自行进行曲线转换后浓度值不准确等问题。

二. 技术参数

工作电压：DC 3.6~5.5V

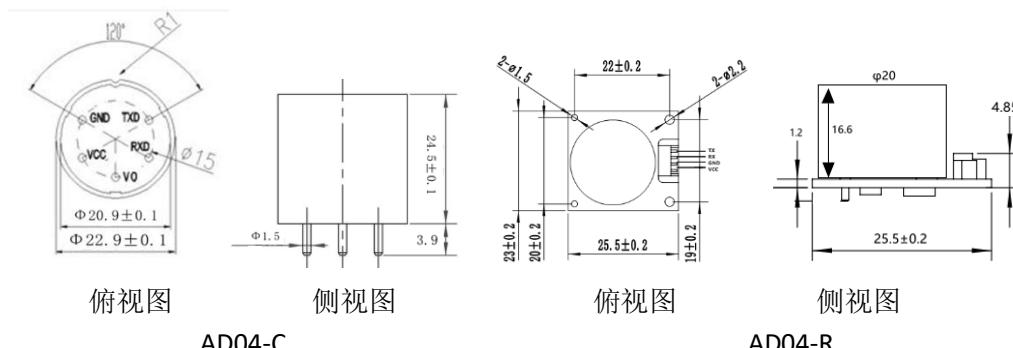
工作温度：-40~70°C (模组) / -20~55 (电化学原理)

工作湿度：0~95%RH 无凝露

工作电流：850uA (不含传感器)

信号输出：UART 输出 (TTL 电平, 3.3V) / 模拟输出 (0.4~2V)

外形尺寸：Φ23mm*24.5mm



俯视图

侧视图

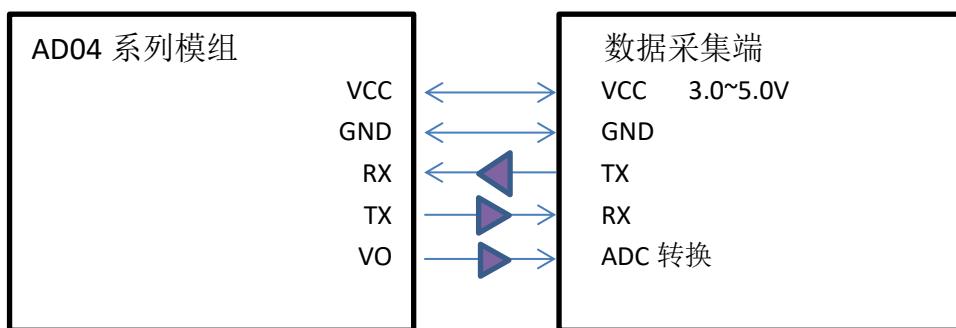
俯视图

侧视图

AD04-C

AD04-R

编号	AD04-C	AD04-R	管脚定义
1	+	+	电源正极
2	-	-	电源负极
3	RX	RX	UART RX 输入
4	TX	TX	UART TX 输出
5	VO	/	模拟电压输出



三. 通信协议

版本	发布日期	版本描述
V1.2	2025.11.19	对 EC 传感器模组通信协议作兼容，修改“DATAG”指令，使传感器模组能够正确输出量程值。
V1.1	2025.08.21	初稿

类型	值
波特率	9600 (默认, 其他可选)
数据位	8
起始位	1, 总是低的
停止位	1, 总是高的
校验位	无校验

通信过程中字母与数字中间的空格键将被忽略。下文中“ “，<>,()等字符均不包含在指令内容中。若收到错误指令，传感器将回复“Invalid Instruction”。建议通信频率不大于1Hz，间隔>1.1S。

常用通信指令：

(1) 数据读取: <DATAG> (hex: “44 41 54 41 47”)

回复长度: 13 字节

回复内容：浓度值（4 字节）+温度 H+温度 L+湿度 H+湿度 L+量程 H+量程 L+实时 AD 值 H+实时 AD 值 L+BCC 校验码+0x0D

回复注释：浓度值单位为 PPB，校验码为 BBC 位之前所有字节的异或，校验码参照表 1.1

温度值计算: $T(^{\circ}\text{C}) = 175.0f * (\text{float}(\text{TH}*256+\text{TL}) / 65535.0f - 45.0f)$

湿度值计算: RH(%)=100.0f* (float) (RH*256+RL)/65535.0f

(表 1.1):

-BCC(Block Check Character)(计算从第 0 位到 BCC 位之前):

```
char BCC_CheckSum(char *buf, char len)
{
    char i;
    char checksum = 0;
    for(i = 0; i < len; i++)
    {
        checksum ^= *buf++;
    }
    return checksum;
}
```

(2) 零点设置<ZERO2> (hex: "5A 45 52 4F 32")

回复长度: 8 字节数据

回复内容: <ZERO OK> (hex:" 5A 45 52 4F 20 4F 4B 0A") 或
< ZERO FAULT> (hex:" 5A 45 52 4F 20 46 41 55 4C 54 0A")

回复注释: 将当前捕获传感器的 AD 值设置为零点

(3) 标定设置<CALB1 XXXXXX>(hex: "43 41 4C 42 31 20 --")

回复长度: 9 字节或 12 字节

回复内容: < CALB1 OK > (hex:" 43 41 4C 42 31 20 4F 4B 0A") 或
< CALB1 FAULT >(hex:" 43 41 4C 42 31 20 46 41 55 4C 54 0A")

回复注释: XXXXXX 为标定气体 PPM 值,默认一位小数点,例如使用 19.9ppm 异丁烯标定 PID-20 传感器, 使用 2 点标定, 即零点与满量程点(默认 0-20ppm 以内为线性, 线性只需标定两点), 则需要发送“CALB1 000199”, 标定成功回复“CALB1 OK”, 标定失败则回复“CALB1 FAULT”

建议传感器上电 1 分钟以上, 优先进行零点调整, 之后通入已知浓度的目标气体, 建议浓度为 40% FS ~60%FS (满量程), 流量大约 500ml/min, 待传感器示数稳定后, 使用本指令。

其余标定指令为< CALB2 XXXXXX>< CALB3 XXXXXX >< CALB4 XXXXXX>, 使用方法与<CALB1 XXXXXX>一致, 注意使用该指令前, 先进行调零指令, 确定使用 2 点标定还是多点标定(最多允许 5 点标定), 标定顺序随标定气体浓度从小到大进行标定, 例如使用 0-1000ppm 量程, 如果进行两点标定, 则只需调零和标定 1 点浓度, 如进行 4 点标定, 则先调零, 在标定 1, 2, 3 点处的浓度值, 如不按标定顺序进行标定, 则会标定失败。

(4) 修改量程<CRANGE XXXXX>(hex: "43 52 41 4E 47 45 20 -")

回复长度: 7-11 字节或 13 字节

回复内容: 若输入<CRANGE 00500>则输出<span: 500> (hex:"73 70 61 6E 3A 35 30 30 0A") (35 30 30 为 500)

或<CRANGE FAULT>(hex:" 43 52 41 4E 47 45 20 46 41 55 4C 54 0A")

回复注释: 量程默认为 0-20000ppm, 超出则修改失败

(5) 恢复默认<INIT>(hex: "49 4E 49 54")

回复长度: 8 字节

回复内容: < INIT OK > (hex:" 49 4E 49 54 20 4F 4B 0A")

回复注释: 将标定过的点, 零点, 量程恢复至默认值(量程默认为 20ppm), 使用该指令后需要重新上电

(6) 电压调整<VADJ XXXX>(hex: "56 41 44 4A 20 --")

回复长度: 7 字节

回复内容: < VADJ OK >{"56 41 44 4A 20 4F 4B"}

内容注释: 零点电压出现偏差时, 通过 VADJ 指令调整电压缩放倍率

例如: 当传感器零点电压输出是 0.41V 时, 把电压调整到 0.4V

通过计算 $0.4/0.41 \times 1000 = 975$

输入指令<VADJ 0975>调整;

反之当电压输出为 0.39V (小于 0.4V) 时,

通过计算 $0.4/0.39 \times 1000 = 1025$

输入指令<VADJ 1025>调整;

其他指令:

(1) 查询数据<CHECK3>(hex:" 43 48 45 43 4B 33")

回复长度: >100 字节

回复内容: 见下表

回复注释: 见下表

Command	<CHECK3>{" 43 48 45 43 4B 33"}
Response	<p>The parameters are as follows:</p> <p>Zeropoint:255</p> <p>CALB1point:0 CALB2point:0 CALB3point:0 CALB4point:0</p> <p>CALB1adjustment:10000</p> <p>CALB2adjustment:10000</p> <p>CALB3adjustment:10000</p> <p>CALB4adjustment:10000</p> <p>spanvalue:500</p> <p>注释:</p> <ol style="list-style-type: none">1.Zeropoint: 零点 AD 值2. CALB1point: 标定点 1 处浓度 (单位 ppm)3. CALB2point: 标定点 2 处浓度 (单位 ppm)4. CALB3point: 标定点 3 处浓度 (单位 ppm)5. CALB4point: 标定点 4 处浓度 (单位 ppm)6. CALB1adjustment: 标定点 1 曲线调整, 通常在 10000 左右7. CALB2adjustment: 标定点 2 曲线调整, 通常在 10000 左右8. CALB3adjustment: 标定点 3 曲线调整, 通常在 10000 左右9. CALB4adjustment: 标定点 4 曲线调整, 通常在 10000 左右10. spanvalue: 量程值 (单位 ppm)

(2) 数据自动发送模式< DATAS >(hex:" 44 41 54 41 53")

回复长度: >30 字节

回复内容: 见下表

内容注释: 见下表

Command	< DATAS >(hex:" 44 41 54 41 53")
Response	ADCValue=255 Volt=0.021 Voltage =0.000 ZeroPoint=255 Span=500 C=0 ppb T 27.97 RH 49.49

本指令主要用于电脑端测试时使用，发送 1 次该指令，模块将以文本的形式输出传感器当前 AD 值+电压值+获取的绝对电压值+零点 AD 值+量程大小+浓度(单位 ppb)+温度+湿度。

再次发送该指令，将停止自动发送。

(3) 数据自动发送模式< AUTOD>(hex:" 41 55 54 4F 44")

回复长度：>30 字节

回复内容：见下表

内容注释：见下表

Command	< AUTOD >(hex:" 41 55 54 4F 44")			
Response	Span=500	C=0 ppb	T 27.97	RH 49.49

本指令主要用于电脑端测试时使用，发送 1 次该指令，模块将以文本的形式输出传感器量程大小+浓度（单位 ppb）+温度+湿度。

再次发送该指令，将停止自动发送。

(4) 修改通信波特率<BAUD X>

回复长度：8 字节

回复内容：无

内容注释：波特率调整指令“BAUD X”，其中 X 可以是 0, 1, 2, 3，重新上电或者恢复默认等操作均保持不变。更快的波特率可以带来更低的通信功耗。

X	Baudrate Value
0	Baudrate=9600
1	Baudrate=57600
2	Baudrate=19200
3	Baudrate=38400

附录一 兼容协议

参照当前客户使用的协议，传感器模组通信协议在原基础上做如下兼容

1. 数据格式

类型	值
波特率	9600
数据位	8
起始位	1, 总是低的
停止位	1, 总是高的
校验位	无校验

2. 通信模式

模式一：查询模式 此模式下，上位机发送指令进行查询指定数据。此模式为默认模式，在每次上电或从休眠中唤醒后，都将进入此模式。

模式二：自动上报模式 此模式下，每隔 1s 会自动上报一组数据，这组数据包括气体浓度值和量程。

两种模式可以使用指令进行切换，具体指令请参照【4.1 模式切换指令】

3. 传感器类型

PID 传感器的类型为 VOC，值（HEX）为 0x18

4. 通信指令与示例

4.1 通讯模式切换指令

4.1.1 从出厂默认的查询模式切换至自动上报模式指令，格式及其示例如下：

下行指令 (Tx) 格式：

Byte[0]	Byte[1]	Byte[2]	Byte[3]	Byte[4]	Byte[5]	Byte[6]	Byte[7]	Byte[8]
0xFF	0x01	0x78	0x40	0x00	0x00	0x00	0x00	0x47

示例解析：0xFF 0x01 0x78 0x40 0x00 0x00 0x00 0x00 0x47-----发送数据为固定格式

上行数据 (Rx) 格式 (每隔 1s)：

Byte[0]	Byte[1]	Byte[2]	Byte[3]	Byte[4]	Byte[5]	Byte[6]	Byte[7]	Byte[8]
0xFF	0x86	Concentration-2 (uint16)	Range (uint16)		Concentration-1 (uint16)	Checksum		

示例解析：0xFF 0x86 0x25 0xBC 0x03 0xE8 0x20 0xD0 0xBE

0xFF 0x86-----指令头

0x25 0xBC-----0x25BC=9660 Concentration-2

根据【4.2 获取模组参数】章节中的参数计算得到 9.66 mg/m3

0x03 0xE8-----0x03E8=1000 Range

根据【4.2 获取模组参数】章节中的参数计算得到 1000 ppm

0x20 0xD0-----0x20D0=8400 Concentration-1

根据【4.2 获取模组参数】章节中的参数计算得到 8.4 ppm

0xBE-----校验和 $(-(0x86+0x25+0xBC+0x03+0xE8+0x20+0xD0)) \& 0xFF$

4.1.2 从主动上传模式切换到查询模式指令

下行指令 (Tx) 格式:

Byte[0]	Byte[1]	Byte[2]	Byte[3]	Byte[4]	Byte[5]	Byte[6]	Byte[7]	Byte[8]
0xFF	0x01	0x78	0x41	0x00	0x00	0x00	0x00	0x46

切换到查询模式示例:

0xFF 0x01 0x78 0x41 0x00 0x00 0x00 0x00 0x46-----发送数据为固定格式

发送完下行命令，接下来 2s 内会有数据回复，即切换成功。

4.2 查询模组参数指令

4.2.1 查询模组指令方式一

下行指令 (Tx) 格式: Byte[0]: 0xD1

示例解析: 0xD1-----发送数据为固定格式

上行数据 (Rx) 格式:

Byte[0]	Byte[1]	Byte[2]	Byte[3]	Byte[4]	Byte[5]	Byte[6]	Byte[7]	Byte[8]
Sensor type	Range (uint16)		Unit type	Reserved			Digital place	Checksum

示例解析: 0x19 0x03 0xE8 0x02 0x00 0x00 0x00 0x30 0xE3

0x19-----传感器类型，请参阅【3.传感器类型表】

0x03 0xE8-----0x03E8=1000 Range

根据【4.2 获取模组参数】章节中的参数计算得到 1000 ppm

0x02-----Concentration-2 单位为 mg/m3, Concentration-1 的单位为 ppm, 具体请参阅【2.气体浓度单位表】

0x00 0x00 0x00-----保留

0x30-----小数位数, 0x30>>4=0x03, 则在计算 Concentration-2 和 Concentration-1 时需要将原始数据除以 1000

0xE3-----校验和 $(-(0x03+0xE8+0x02+0x00+0x00+0x00+0x30)) \& 0xFF$

4.2.2 查询模组指令方式二

此方式和 4.2.1 指令方式一的功能相同，指令格式不同，用户可以自行选择使用。

下行指令 (Tx) 格式: Byte[0]: 0xD7

示例解析: 0xD7-----发送数据为固定格式

上行数据 (Rx) 格式:

Byte[0]	Byte[1]	Byte[2]	Byte[3]	Byte[4]	Byte[5]	Byte[6]	Byte[7]	Byte[8]
0xFF	0xD7	Sensor type	Range (uint16)		Unit type	Digital place	Reserved	Checksum

示例解析: 0xFF 0xD7 0x19 0x03 0xE8 0x02 0x30 0x00 0xF3

0xFF 0xD7-----指令头

0x19-----传感器类型，请参阅【3.传感器类型表】

0x03 0xE8-----0x03E8=1000 Range

根据【4.2 获取模组参数】章节中的参数计算得到 1000 ppm

0x02-----Concentration-2 单位为 mg/m3 Concentration-1 的单位为 ppm

具体请参阅【2.气体浓度单位表】

0x30-----小数位数, 0x30>>4=0x03, 则在计算 Concentration-2 和 Concentration-1 时需要将原始数据除以 1000

0xF3-----校验和 $(-(0xD7+0x19+0x03+0xE8+0x02+0x30+0x00)) \& 0xFF$

4.3 查询传感器当前浓度值指令

下行指令 (Tx) 格式:

Byte[0]	Byte[1]	Byte[2]	Byte[3]	Byte[4]	Byte[5]	Byte[6]	Byte[7]	Byte[8]
0xFF	0x01	0x86	0x00	0x00	0x00	0x00	0x00	0x79

示例解析: 0xFF 0x01 0x86 0x00 0x00 0x00 0x00 0x00 0x79-----发送数据为固定格式

上行数据 (Rx) 格式:

Byte[0]	Byte[1]	Byte[2]	Byte[3]	Byte[4]	Byte[5]	Byte[6]	Byte[7]	Byte[8]
0xFF	0x86	Concentration-2 (uint16)	Range (uint16)	Concentration-1 (uint16)	Checksum			

示例解析: 0xFF 0x86 0x25 0xBC 0x03 0xE8 0x20 0xD0 0xBE

0xFF 0x86-----固定头

0x25 0xBC-----0x25BC=9660 Concentration-2

根据【4.2 获取模组参数】章节中的参数计算得到 9.66 mg/m3

0x03 0xE8-----0x03E8=1000 Range

根据【4.2 获取模组参数】章节中的参数计算得到 1000 ppm

0x20 0xD0-----0x20D0=8400 Concentration-1

根据【4.2 获取模组参数】章节中的参数计算得到 8.4 ppm

0xBE-----校验和 $(-(0x86+0x25+0xBC+0x03+0xE8+0x20+0xD0)) \& 0xFF$

4.4 查询传感器当前浓度值以及温湿度指令

下行指令格式:

Byte[0]	Byte[1]	Byte[2]	Byte[3]	Byte[4]	Byte[5]	Byte[6]	Byte[7]	Byte[8]
0xFF	0x01	0x87	0x00	0x00	0x00	0x00	0x00	0x78

示例解析: 0xFF 0x01 0x87 0x00 0x00 0x00 0x00 0x00 0x78-----发送数据为固定格式

上行数据格式:

Byte[0]	Byte[1]	Byte[2]	Byte[3]	Byte[4]	Byte[5]	Byte[6]	Byte[7]	Byte[8]	Byte[9]	Byte[10]	Byte[11]	Byte[12]
0xFF	0x87	Concentration-2 (uint16)	Range (uint16)	Concentration-1 (uint16)	Temperature (int16)	Rh (uint16)	Checksum					

示例解析: 0xFF 0x87 0x25 0xBC 0x03 0xE8 0x20 0xD0 0x07 0x3B 0x21 0x07 0x53

0xFF 0x87-----固定头

0x25 0xBC-----0x25BC=9660 Concentration-2

根据【4.2 获取模组参数】章节中的参数计算得到 9.66 mg/m3

0x03 0xE8-----0x03E8=1000 Range

根据【4.2 获取模组参数】章节中的参数计算得到 1000 ppm

0x20 0xD0-----0x20D0=8400 Concentration-1

根据【4.2 获取模组参数】章节中的参数计算得到 8.4 ppm

0x07 0x3B-----0x073B=1851 Temperature 固定 2 位小数, 18.51 摄氏度

0x21 0x07-----0x2107=8455 Rh 固定两位小数, 84.55% Rh

0x53----- $(-(0x87+0x25+0xBC+0x03+0xE8+0x20+0xD0+0x07+0x3B+0x21+0x07)) \& 0xFF$

4.5 查询温度和相对湿度指令

4.5.1 指令方式一

下行指令格式: Byte[0] : 0xD2

示例解析: 0xD2-----发送数据为固定格式

上行数据格式:

Byte[0]	Byte[1]	Byte[2]	Byte[3]
Temperature (int16)	Rh (uint16)		

示例解析: 0x07 0x3B 0x21 0x07

0x07 0x3B-----0x073B=1851 Temperature 固定 2 位小数, 18.51 摄氏度 0x21

0x07-----0x2107=8455 Rh 固定两位小数, 84.55% Rh

4.5.2 指令方式二

此方式和 4.5.1 指令方式一的功能相同, 指令格式不同, 用户可以自行选择使用。

下行指令 (Tx) 格式: Byte[0] : 0xD6

示例解析: 0xD6-----发送数据为固定格式

上行数据 (Rx) 格式:

Byte[0]	Byte[1]	Byte[2]	Byte[3]	Byte[4]
Temperature (int16)	Rh (uint16)		Checksum	

示例解析: 0x07 0x3B 0x21 0x07 0x96

0x07 0x3B-----0x073B=1851 Temperature 固定 2 位小数, 18.51 摄氏度 0x21

0x07-----0x2107=8455 Rh 固定两位小数, 84.55% Rh

0x96-----校验和 $(-(0x07+0x3B+0x21+0x07)) \& 0xFF$

4.6 查询软件版本号指令

下行指令 (Tx) 格式: Byte[0] : 0xD3

示例解析: 0xD3-----发送数据为固定格式

上行数据 (Rx) 格式:

Byte[0]	Byte[1]	Byte[2]	Byte[3]	Byte[4]	Byte[5]
Software version (BCD)					

示例解析: 0x20 0x23 0x11 0x08 0x14 0x54

0x20 0x23 0x11 0x08 0x14 0x54-----0x202311081454 按 BCD 码转换后 202311081454

4.7 恢复出厂标定指令

在操作标定过程中若出现误操作, 导致标定后的测量误差较大或不正确的测量数据, 以及其它需求时, 可以采
复出厂的标定数据。

下行指令 (Tx) 格式:

Byte[0]	Byte[1]	Byte[2]	Byte[3]	Byte[4]	Byte[5]	Byte[6]	Byte[7]	Byte[8]
0xFF	0x01	0x8E	0x00	0x00	0x00	0x00	0x00	0x71

示例解析: 0xFF 0x01 0x8E 0x00 0x00 0x00 0x00 0x00 0x71-----发送数据为固定格式

上行数据 (Rx) 格式:

Byte[0]	Byte[1]
0x4F	0x4B

示例解析: 0x4F 0x4B-----发送数据为固定格式

4.8 用户校准指令

气体传感器在使用过程中，当使用环境比较恶劣或者长时间的测量，会导致传感器信号衰减或测量超过误差值，以及用户有定期进行校准的需求，那么可以采用以下指令对模组的零点和灵敏度进行校准，确保其测量的准确度。

在校准时可以选择分别对零点和灵敏度校准：

- 1) 当只对零点校准时，那么灵敏度保持出厂校准数据或上一次校准的数据。

用户零点校准示例：写入 0ppm，该值按照 IEEE754 single precision 32-bit 转换后的十六进制整数写入模组。

- 2) 当对灵敏度校准时，建议零点保持出厂数据或上一次零点校准数据的基础上，再通入定量气体进行校准。

例如选用 10ppm 的标准气体进行校准灵敏度时，写入 10ppm，10.0f 按照 IEEE754 single precision 32-bit 转换后的十六进制整数为:0x41200000。

Float 类型数据转换请参考链接 (http://www.binaryconvert.com/convert_float.html)。

C 语言可参看如下联合体进行转换，举例：

```
typedef union {
    float data;
    uint32_t uint32_data;
    uint8_t uint8_data[4];
}FLOAT_DATA_U;
```

零点校准和灵敏度校准所采用的指令相同，通常先校准零点，再校准灵敏度，指令如下：

下行指令 (Tx) 格式：

Byte[0]	Byte[1]	Byte[2]	Byte[3]	Byte[4]	Byte[5]	Byte[6]	Byte[7]	Byte[8]
0xFF	0x01	0x8D	Concentration (float)(<=range/2)			0x00		Checksum

示例解析：0xFF 0x01 0x8D 0x41 0x20 0x00 0x00 0x00 0x11

0xFF 0x01 0x8D-----指令头

0x41 0x20 0x00 0x00-----0x41200000 - 10.0f (IEEE754 single precision 32-bit)

0x00-----保留

0x11-----校验和 (-(0x01+0x8D+0x41+0x20+0x00+0x00+0x00))&0xFF

上行数据 (Rx) 格式：

Byte[0]	Byte[1]
0x4F	0x4B

示例解析：0x4F 0x4B-----发送数据为固定格式