

ZG09 二氧化碳模块

使用手册

目录

1 概述.....	3
2 设计特点.....	3
3 规格.....	3
4 UART.....	5
5 RS485 通讯.....	7
6 I2C 通讯.....	8
7 PWM 输出.....	8
8 DAV 输出.....	8
9 可靠度.....	8
10 响应时间.....	9
11 CO ₂ RMS 杂讯.....	9

1. 概述

本使用手册及设定指南，让您更熟悉如何安全及有效地使用 ZG09 二氧化碳模块。
我们已经作了最大努力以确保本操作手册中的信息为最新信息，但是，请注意，ZG09 二氧化碳模块的规范仍会不经通知即作出变更。

版权所有:

未经 ZyAura.com 书面允许，不得复制或传播本手册中的任何部分。本使用手册中所含信息如有更改，恕不另行通知。

有限保证:

使用手册中的特定产品讯息，仅使用于协助用户了解产品，它们不构成任何具有约束力的要约。

商标:

所有商标均为其各自所有者的财产。

2. 设计特点

使用双光束 NDIR 传感器

气体采样兼容扩散式与通气式

超越室内空气质量需求，量测范围可达 0~10,000ppm。

支持 UART、RS485、I2C、PWM、DAC 输出格式

3. 规格

CO ₂ 量测范围	0~10,000ppm
量测周期	2 秒
CO ₂ 精度	±50ppm±3% 读值
CO ₂ RMS 噪声	<10ppm @ 400ppm, <20ppm @ 1000ppm
CO ₂ 再现性	±20ppm
分辨率	1ppm
响应时间	扩散式约 1 分钟 (90%上升时间)
大气压力影响	每毫升汞柱 读数变化 0.13%
暖机时间 (冷启动)	30 秒 @ 25°C
供应电压	5.0VDC±0.5V
电流消耗	最大: (0.8 ms 截取测试) <190 毫安, 平均<34 毫安
红外线关闭平均耗电流	<10 毫安
操作温度	0~50°C
储存温度	-20~60°C
操作湿度	0-95% RH
储存湿度	0-95% RH
外观尺寸	32.2 x 20.2 x 13.7 公厘

通讯:

UART	3.3V Modbus RTU 协议
RS485	RS485 R/T 与 SCL 共享接脚
I2C	最高频率 400kHz
PWM	3.3V, 频率 1kHz。
DAC 输出	0~1V = 800~1,200ppm

ZG09 接脚尺寸图

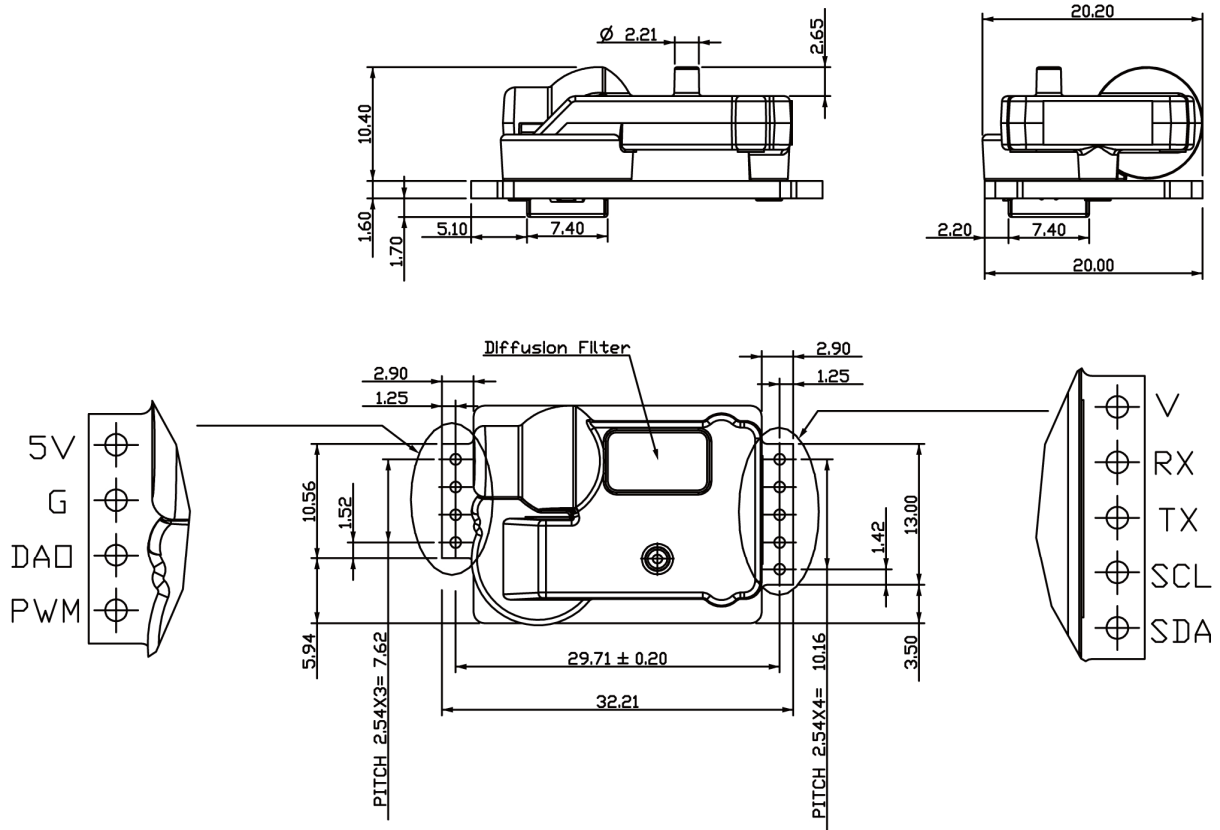


图 1: 模块外观尺寸图

注意: 图片尺寸仅供参考, 实际尺寸以标示为主。

接脚说明:

接脚名称	功能
PWM	CO ₂ ppm PWM 输出
DAO	DAC 输出
G	接地
5V	输入电压 +5V
SDA	I2C Data 接脚
SCL	I2C Clock 接脚(预设) 或 RS485 R/T 控制接脚
TX	UART TX
RX	UART RX
V	3.3V 输出

4. UART

UART 默认值:

BPS: 19200
Data Bit: 8
Parity bit: None
Stop bit: 1

通讯协议:

ZG09 UART 使用的是 Modbus RTU 协议, 指令结构分成发送(Send)与接收(Receive), 如下:

发送(Send)

编号 (1 Byte)	功能码 (1 Byte)	起始地址 (2 Bytes)	读回资料量 (2 Bytes)	检查 (2 Bytes)
FEH	03H	0000H (Note 1)	0003H	XXXX

接收(Receive)

编号 (1 Byte)	功能码 (1 Byte)	读回资料量 (1 Byte)	数据内容 (N Bytes)	检查 (2 Bytes)
FEH	03H	06H (Note 2)	6 * 8 bit (Note 3)	XXXX

注 1: 起始地址 = 想要读取该笔数据的地址, 例如: CO₂ 值 = 0x0B (参见起始地址表)

注 2: 读回数据量 = 数据内容的长度 (N)

注 3: 接收的数据内容的长度(N) = 发送的读取数据量 x 2

编号

用户可自行设定设备的编码, 出厂默认为 0x01, 0xFA 和 0xFE 为广播预留。

功能码

0x03 Read

0x06 Write

起始地址表

起始地址	Read / Write	说明
00(0x00) : SlaveID	R/W	ID 设定: 出厂预设 0x01 250 (0xFA)和 254 (0xFE)为广播预留, 无法设定。
01(0x01) : BPS	R/W	Baud rate 设定 (需要重新设定或重新上电后, 才会套用设定值。) 0x00 : 9600 0x01 : 19200 (预设) 0x02 : 38400 0x03 : 57600
02(0x02) : Uart Parity Set	R/W	Uart Parity 设定 (需要重新设定或重新上电后, 才会套用设定值。) 0x00:n82 0x01:n81 (预设) 0x02:e81 0x03:o81
11(0x0B) : CO ₂ Value	R	CO ₂ 浓度读数以 ppm 为单位

CO₂ 读取范例:

使用广播方式，发送指令 FA03 000B 0001 E043 到编号 250(0xFA)。

编号(ID) 0xFE 或 0xFA 被传感器定义为任意编号(ID)

编号(ID) (1 Byte)	功能码 (1 Byte)	起始地址 (2 Bytes)	读取数据量 (2 Bytes)	检查码 (2 Bytes)	注
(1) FAH	03H	000BH	0001H	E043H	
(2) FEH	04H	0003H	0001H	D5C5H	

回传资料 0103 0201 F8B8 56

编号(ID) (1 Byte)	功能码 (1 Byte)	读取数据量 (1 Byte)	数据内容 (N Bytes)	检查码 (2 Bytes)	注
(1) FAH	03H	02H	01ACH	5C7DH	428ppm
(2) FEH	04H	02H	0197H	ED1AH	407ppm

回传的数据内容为发送读取数据量的 2 倍

CO₂ 读值为 0x01ACH = 428ppm, 0x0197=407ppm

检查码

unsigned int Crc16(unsigned char * data, unsigned char length)

```

{
    int j;
    unsigned int reg_crc=0xFFFF;
    while(length--)
    {
        reg_crc ^= *data++;
        for(j=0;j<8;j++)
        {
            if(reg_crc & 0x01) /* LSB(b0)=1 */
                reg_crc=reg_crc >>1 ^ 0xA001;
            else
                reg_crc=reg_crc >>1;
        }
    }
    return reg_crc;
}
    
```

检查码 (查表法)

unsigned int Crc(unsigned char *ucTx, unsigned int Len)

```

{
    if (Len && ucTx)
    {
        unsigned char *ucPtr=ucTx;
        unsigned char ucCRCHi = 0xff;
        unsigned char ucCRCLo = 0xff;
        unsigned ulIndex;
        while(Len--)
        {
            ulIndex = ucCRCHi ^ *ucPtr++;
            ucCRCHi = ucCRCLo ^ ucCRCHigh[ulIndex];
            ucCRCLo = ucCRCLow[ulIndex];
        }
        return (unsigned int)(ucCRCLo << 8) | (unsigned int)ucCRCHi;
    }
    else
        return 0;
}
    
```

unsigned char ucCRCHigh[] =

```

{
0x00,0xc1,0x81,0x40,0x01,0xc0,0x80,0x41,0x01,0xc0,0x80,0x41,0x00,0xc1,0x81,
0x40,0x01,0xc0,0x80,0x41,0x00,0xc1,0x81,0x40,0x00,0xc1,0x81,0x40,0x01,0xc0,
0x80,0x41,0x01,0xc0,0x80,0x41,0x00,0xc1,0x81,0x40,0x00,0xc1,0x81,0x40,0x01,
0xc0,0x80,0x41,0x00,0xc1,0x81,0x40,0x01,0xc0,0x80,0x41,0x01,0xc0,0x80,0x41,
0x00,0xc1,0x81,0x40,0x01,0xc0,0x80,0x41,0x00,0xc1,0x81,0x40,0x00,0xc1,0x81,
0x40,0x01,0xc0,0x80,0x41,0x00,0xc1,0x81,0x40,0x01,0xc0,0x80,0x41,0x01,0xc0,
0x80,0x41,0x00,0xc1,0x81,0x40,0x00,0xc1,0x81,0x40,0x01,0xc0,0x80,0x41,0x01,
0xc0,0x80,0x41,0x00,0xc1,0x81,0x40,0x01,0xc0,0x80,0x41,0x00,0xc1,0x81,0x40,
0x01,0xc0,0x80,0x41,0x00,0xc1,0x81,0x40,0x01,0xc0,0x80,0x41,0x01,0xc0,
0x80,0x41,0x00,0xc1,0x81,0x40,0x01,0xc0,0x80,0x41,0x00,0xc1,0x81,0x40,0x01,
0xc0,0x80,0x41,0x01,0xc0,0x80,0x41,0x00,0xc1,0x81,0x40,0x01,0xc0,0x80,0x41,
0x00,0xc1,0x81,0x40,0x00,0xc1,0x81,0x40,0x01,0xc0,0x80,0x41,0x00,0xc1,0x81,
0x40,0x01,0xc0,0x80,0x41,0x01,0xc0,0x80,0x41,0x00,0xc1,0x81,0x40,0x01,0xc0,
0x80,0x41,0x00,0xc1,0x81,0x40,0x00,0xc1,0x81,0x40,0x01,0xc0,0x80,0x41,0x01,
0xc0,0x80,0x41,0x00,0xc1,0x81,0x40,0x00,0xc1,0x81,0x40,0x01,0xc0,0x80,0x41,
0x00,0xc1,0x81,0x40,0x01,0xc0,0x80,0x41,0x01,0xc0,0x80,0x41,0x00,0xc1,0x81,
0x40
};

unsigned char ucCRCLow[] =
{
0x00,0xc0,0xc1,0x01,0xc3,0x03,0x02,0xc2,0xc6,0x06,0x07,0xc7,0x05,0xc5,0xc4,
0x04,0xcc,0xc0,0x0c,0xd0,0xcd,0x0f,0xcf,0xce,0x0e,0x0a,0xca,0xcb,0x0b,0xc9,0x09,
0x08,0xc8,0xd8,0x18,0x19,0xd9,0x1b,0xdb,0xda,0x1a,0x1e,0xde,0xdf,0x1f,0xdd,
0x1d,0x1c,0xdc,0x14,0xd4,0xd5,0x15,0xd7,0x17,0x16,0xd6,0xd2,0x12,0x13,0xd3,
0x11,0xd1,0xd0,0x10,0xf0,0x30,0x31,0xf1,0x33,0xf3,0xf2,0x32,0x36,0xf6,0xf7,
0x37,0xf5,0x35,0x34,0xf4,0x3c,0xfc,0xfd,0x3d,0xff,0x3f,0x3e,0xfe,0xfa,0x3a,
0x3b,0xfb,0x39,0xf9,0xf8,0x38,0x28,0xe8,0xe9,0x29,0xeb,0x2b,0x2a,0xea,0xee,
0x2e,0x2f,0xef,0x2d,0xed,0xec,0x2c,0xe4,0x24,0x25,0xe5,0x27,0xe7,0xe6,0x26,
0x22,0xe2,0xe3,0x23,0xe1,0x21,0x20,0xe0,0xa0,0x60,0x61,0xa1,0x63,0xa3,0xa2,
0x62,0x66,0xa6,0xa7,0x67,0xa5,0x65,0x64,0xa4,0x6c,0xac,0xad,0x6d,0xaf,0x6f,
0x6e,0xae,0xaa,0xa6,0x6b,0xab,0x69,0xa9,0xa8,0x68,0x78,0xb8,0xb9,0x79,0xbb,
0x7b,0x7a,0xba,0xbe,0x7e,0x7f,0xbf,0x7d,0xbd,0xbc,0x7c,0xb4,0x74,0x75,0xb5,
0x77,0xb7,0xb6,0x76,0x72,0xb2,0xb3,0x73,0xb1,0x71,0x70,0xb0,0x50,0x90,0x91,
0x51,0x93,0x53,0x52,0x92,0x96,0x56,0x57,0x97,0x55,0x95,0x94,0x54,0x9c,0x5c,
0x5d,0x9d,0x5f,0x9f,0x9e,0x5e,0x5a,0x9a,0x9b,0x5b,0x99,0x59,0x58,0x98,0x88,
0x48,0x49,0x89,0x4b,0x8b,0x8a,0x4a,0x4e,0x8e,0x8f,0x4f,0x8d,0x4d,0x4c,0x8c,
0x44,0x84,0x85,0x45,0x87,0x47,0x46,0x86,0x82,0x42,0x43,0x83,0x41,0x81,0x80,
0x40
};

```

5. RS485 通讯

运用电路

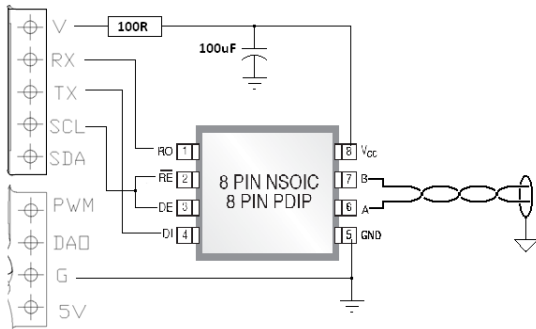


图 2

外接 RS-485/RS422 收发芯片(SP3072)可长距离监测环境 CO₂ 浓度，距离拉长必须使用隔离绞线，收发芯片可使用 ZG09 的 3V 输出电源，如上图 (图 2) 方式连接，接线尽可能缩短，通讯协议依照 ModBus 格式 (参阅 4. UART)。

6. I2C 通讯

I2C 界面具有 100KHz~400KHz 的通讯速率。

7. PWM 输出

可自定义 CO₂ 浓度范围的 PWM VCC (3.0~3.3V) 于 1 kHz 输出 0~10,000ppm, 运用电路如下图 (图 3), AL1=0~10,000ppm AL2=0~10,000ppm。AL1<AL2 可用于 HVAC; AL1>AL2 可用于农业; 使用于气体浓度输出 0~10,000ppm AL1=0ppm、AL2=10,000ppm

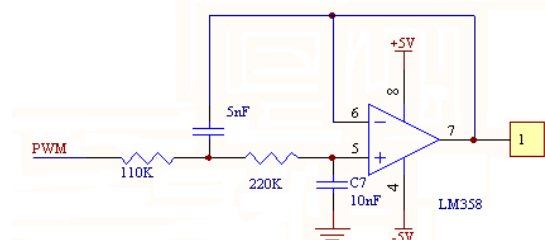


图 3

运用	AL1 0~10,000ppm	AL2 0~10,000ppm	备注
监测 CO ₂ 浓度	0ppm (0V)	10,000ppm (VCC)	
HVAC	预设 800ppm (<800ppm=0V)	预设 1200ppm (>1200ppm=VCC)	注 1
农业	自定义	自定义	注 2

注 1: 一般 HVAC 使用, 为环境 CO₂ 浓度大于 AL2 1200ppm(可自定义), 启动通风。当通风后, 环境 CO₂ 浓度降至 800ppm(可自定义), 停止通风。

注 2: 植物栽种使用, AL1>AL2 的方式设定。当植物环境 CO₂ 浓度低于 400ppm(可自定义), 增加植物环境 CO₂ 浓度; 植物环境 CO₂ 高于 AL2 1200ppm(可自定义) 时, 停止供应 CO₂。

8. DAV 输出

DAV 输出是可用户控制输出使用, 不建议做为高精密度 CO₂ 量测使用。输出 0~1V = 预设 800~1,200ppm (~210 阶), 可藉由 AL1 与 AL2 来控制范围, 运用端可以使用比较器来控制继电器。AL2>AL1 可运用于 HVAC, AL1>AL2 运用于农业。当 AL1 设置为 800ppm, 此时读值达 800ppm 时, DAV 输出 0V, 如果希望全范围输出, 可将 AL1 设置为 0ppm、AL2 设置为 10,000ppm。

9. 可靠度

9.1 长期稳定度

取样 16 支经过 25°C、RH70%、500 小时长时间测试, 平均结果显示于(图 4), Y 轴为 CO₂ 浓度, X 轴为时间。

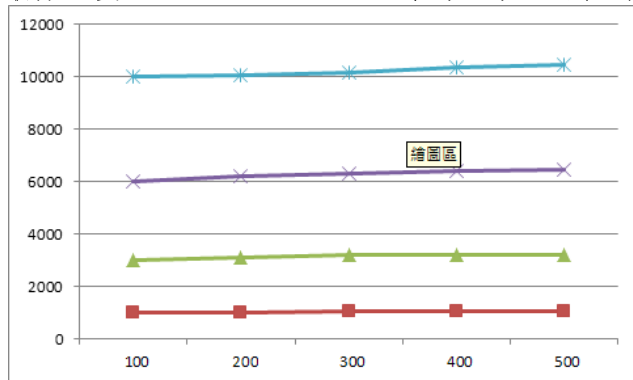


图 4

9.2 温度循环测试

取样 16 支经过 -20°C、+65、+60°C、RH85%、400 小时长时间测试，平均结果显示于(图 5)，Y 轴为 CO₂ 浓度，X 轴为时间。

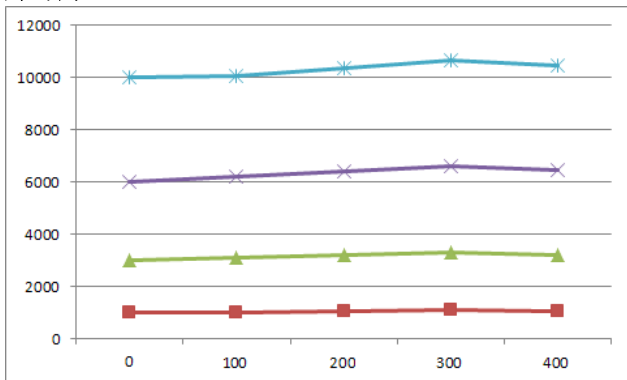


图 5

9.3 高低温测试

显示 -20°C ~+60°C 与 5V±0.25V 操作 CO₂ 读值的表现(图 6)，测试结果 4.75V~5.25V 读值表现一致，当电压降至 4.5V 才会对精度造成影响。

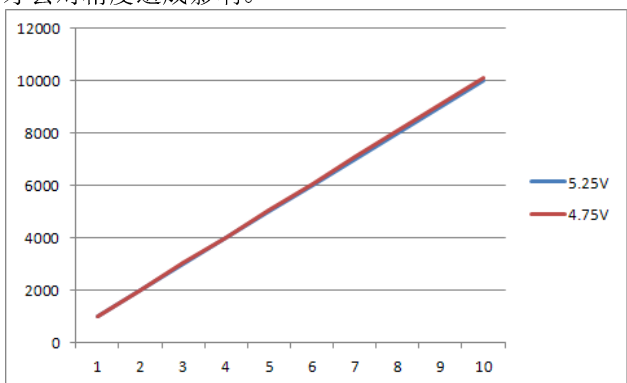


图 6

10. 响应时间

显示环境 CO₂ 改变多久时间可以平衡，读值与环境气体浓度一致 (表一)。测试方式由 1,000ppm 放置 10 分钟，快速移至 400ppm 环境与 400ppm 放置 10 分钟，快速移至 1,000ppm 环境。

(表一)

测试项目	时间
400ppm~1,000ppm	~60 秒 (90%上升时间)
1,000ppm~400ppm	~120 秒 (90%下降时间)

11. CO₂ RMS 噪声

显示于测试条件在 1,000ppm 与 25°C±2°C 下，量测 3 小时，量测的稳定度与变化小于±20ppm。

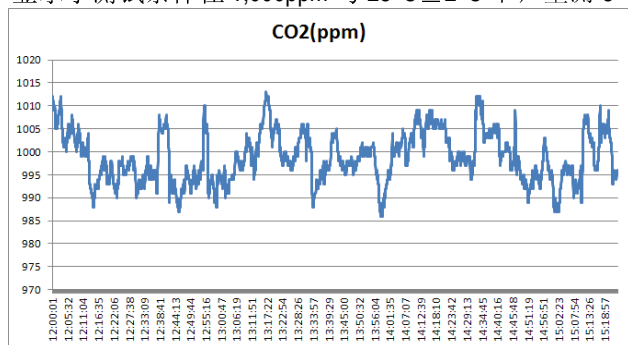


图 7