MH-440D infrared gas sensor

1. Introduction

MH-440D infrared gas sensor is a miniature universal intelligent sensor, which adopts NDIR theory to detect concentration of CH_4 in air and has good selectivity, stable performance, long life, also is independent of Oxygen. The inside temperature sensor could be used for temperature compensation. This miniature infrared gas sensor is developed by the tight integration of mature infrared absorbing gas detection technology, micro machine workout and superior circuit design.

It is convenient in use and also instead of catalytic component, widely used in various occasions with flammable and explosion hazard gas.

2. Features and main technical parameters

2.1 Features

High sensitivity

Standard output and signal output

Miniature figure

Fast response and resume

Temperature compensation

Good stability

Long life

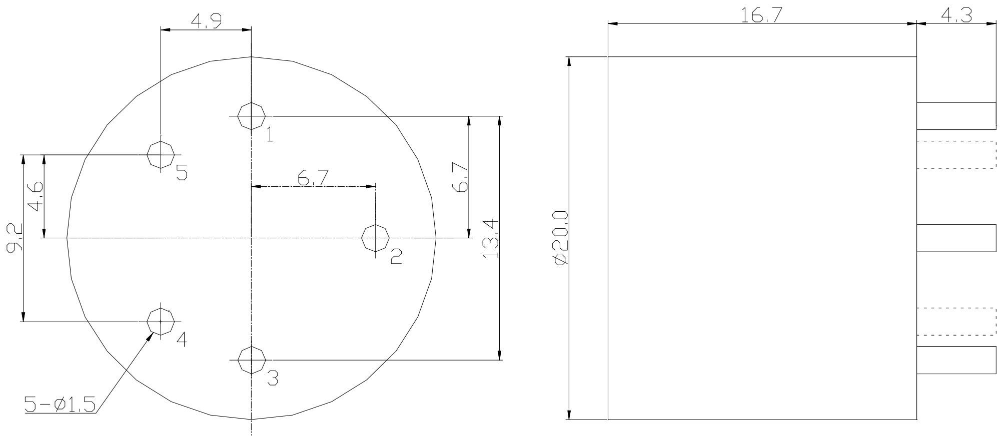
Anti-vapor interference

Instantly convert catalytic theory meter into infrared detection meter

2.2 Main technical parameters

Working voltage	3.5~5.5V dc
Working current	75~85mA
Detection range	0~5%vol（0~100%vol optional）
Output signal range	0.4~2V dc
Resolution	1%FSD
Warm-up time	90s
Response time	T90<30s
Repeatability	Zero < ±100ppm
	SPAN <±500ppm
Long-term drift	Zero < ±300ppm/month
	SPAN < ±500ppm/month
Temperature range	-20°C ~50°C
Humidity range	0~95%RH
Lifetime	>5 years
Dimension	20*16.6
Weight	15g

3. Configuration dimension



4. Pin definition

1． GND
2． Vout
3． VCC
4． TXD
5． RXD

5. Model selection

This product has two types for your selection:

MH-440V

3 pins—It is same as pins of catalytic sensor, and the linear output supplied is familiar with the ones of catalytic.

MH-440D

5 pins—The 3 pins of them are same as pins of catalytic sensor, and the other two pins are UART pins, which can communicate with circuit directly and offer more choices for users.

NDIR Gas sensor and modules MH-410D

Brief Introduction:

MH-410D is a miniature universal intelligent sensor, which adopts NDIR theory to detect concentration of CO2 in air and has good selectivity, stable performance, long life, also is independent of Oxygen. The inside temperature sensor could be used for temperature compensation.

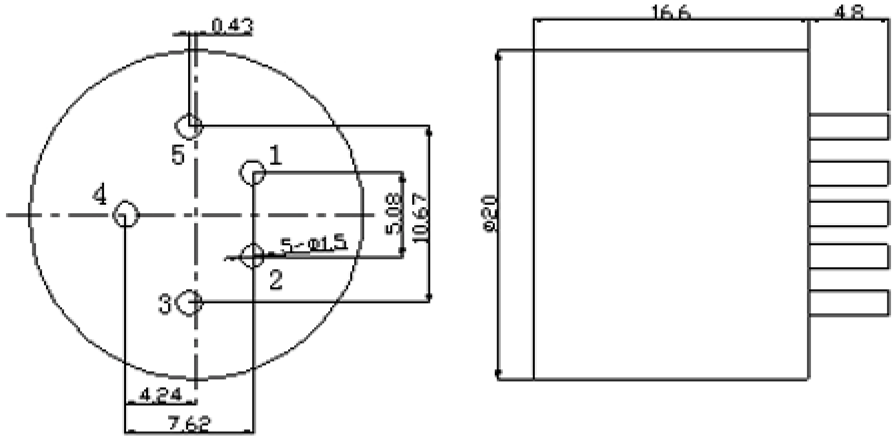
It could be used to replace catalytic component, widely used in occasions with flammable and explosion hazard gas.

Technical Data:

Detected Gas	CO2
Working voltage	3.6-5V DC
Working current	75-85mA
Interface Level	3V
Detection range	0-5%vol(0-100%vol optional)
Output signal range	0.4-2V DC
Resolution	1%FSD
Warm-up time	90s
Response time	T90<30s
Repeatability	Zero < +/-100ppm
	SPAN < +/-500ppm
Long-term drift	Zero < +/-300ppm/month
	SPAN < +/-500ppm/month
Temperature range	-20°C -60°C
Humidity range	0-95%RH
Lifetime	>5 years
Dimension	20mm*16.6mm

Weight	15g

Dimension:



NDIR Gas sensor and modules MH-490W

Brief Introduction:

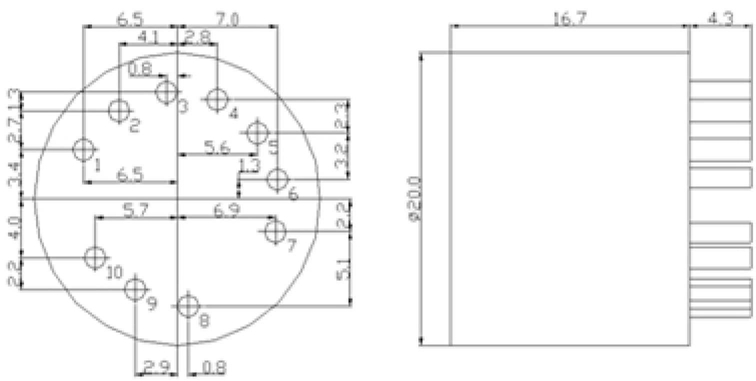
MH-490W is a miniature universal sensor, which adopts NDIR theory to detect concentration of CH_4 and CO_2 (or CH_4 and CO;CO and CO_2) in air and has good selectivity, stable performance, long life, also is independent of Oxygen. The inside temperature sensor could be used for temperature compensation, it also has EEPROM memorizer.

It could be widely used in occasions of fire detection and explosion hazard gas.

Technical Data:

Detected Gas	CH_4 and CO_2
Working voltage	3.5-5.5V DC
Working current	<100mA
Detection range	CH_4: 0-100%VOL CO_2: 0-50%VOL
Resolution	CH_4: 0.1%VOL CO_2: 0.1%VOL
Warm-up time	90s
Response time	T90<30s
Repeatability	Zero < +/-0.2%VOL
	SPAN < +/-1%VOL/month
Long-term drift	Zero < +/-0.6 VOL /month
	SPAN < +/-1%VOL /month
Temperature range	-20°C -50°C
Humidity range	0-99%RH
Lifetime	>5 years
Working Frequency	1Hz
Dimension	20mm*21mm
Weight	15g

Dimension:

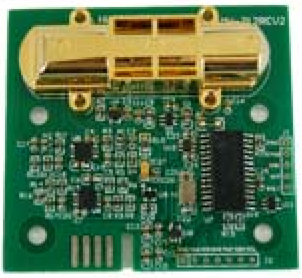


NDIR Gas sensor and modules MH-Z12

Brief Introduction:

MH-Z12 is a general, mini sensor, which adopts NDIR theory to measure concentration of CO_2 in air, it has good selectivity and long life. Temperature sensor inside could be for temperature compensation, it has both digital output and Analog voltage output.

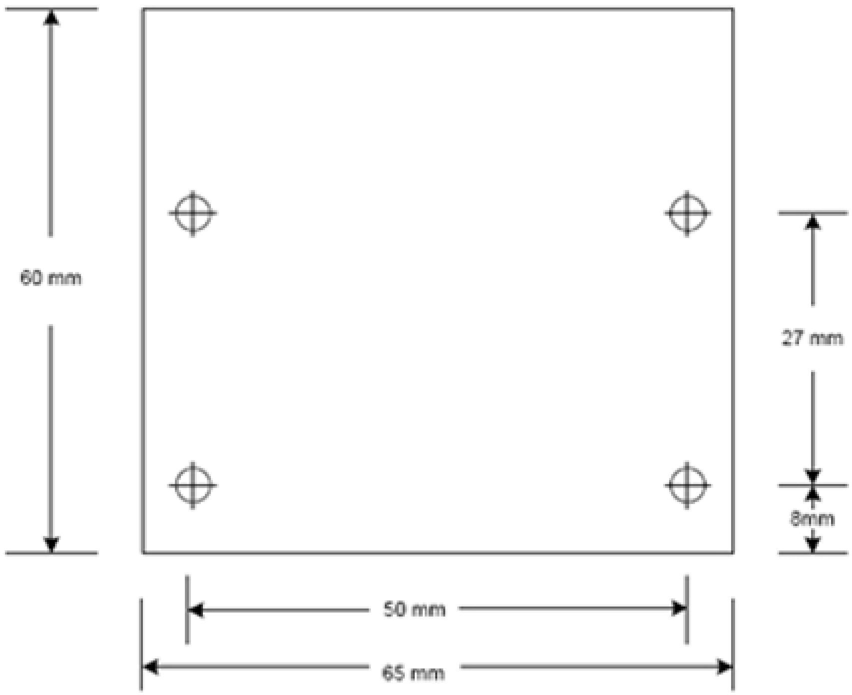
Widely used for HVAC, indoor and outdoor air quality control, industrial, agriculture and animal husbandry.



Technical Data:

Detection gas	CO_2
Measurement range	0-5000ppm (Optional)
Resolution	5ppm(0-2000ppm) 10ppm(2000-5000ppm)
Accuracy	+/-50ppm
Repeatability	+/-30ppm
Response time	< 30s
Preheating time	3 min
Working voltage	4-6V
Working current	Max. 100mA, average less than 60mA
Working environment	Tem.: 0-60^{o}C
	Hum.: 0%-90%RH (No Dew)
Storage environment	Tem.: -20-60^{o}C
Lifetime	>5 year

Dimension:



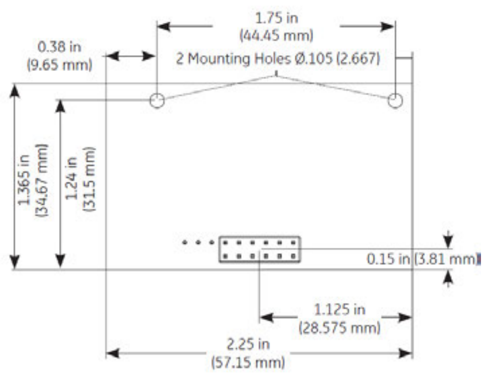
MH-Z14 CO2 Module

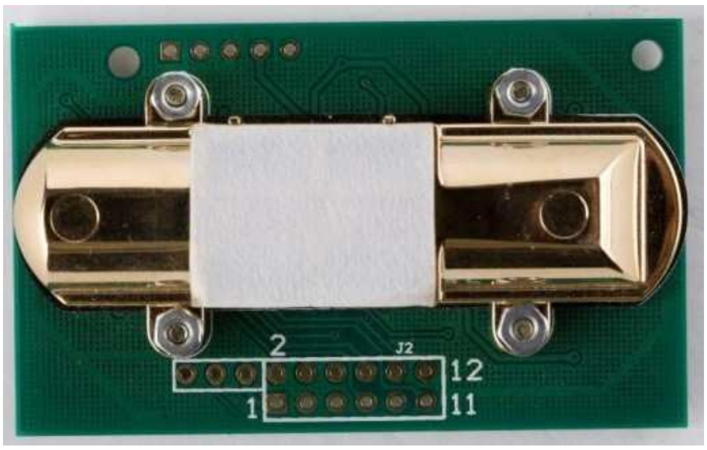
MH-Z14 NDIR Infrared gas module is a common type, small size sensor, using non-dispersive infrared (NDIR) principle to detect the existence of CO_2 in the air, with good selectivity, non-oxygen dependant, long life. Built-in temperature sensor can do temperature compensation; and it has digital output and analog voltage output. MH-490W integrate sophisticated infrared absorption gas detection technology, sophisticated light transmission design and sophisticated circuit design.
MH-Z14 NDIR Infrared gas module is applied in the HVAC, indoor air quality monitoring, industrial process, safety and protection monitoring, agricultrue and animal husbandry production process monitoring.

1. Technical specification:

Detection range	0～10000ppm（optional）
Resolution ratio	5ppm（0～2000ppm）
	10ppm（2000～5000ppm）
	20ppm（5000～10000ppm）
Accuracy	±50ppm±5%
Repeatability	±30ppm
Responsible time	<30S
Warm-up time	3min
Working temprature	0～50℃
Working humidity	0%～90%RH（No condensation）
Storage temprature	-20～60℃
Working voltage	4～6V
Working current	Max current <100mA，Average current <50mA
Usingage	>5year

2. Structure Dimension Chart





3. Signal output

Signal output：analog voltage output, PWM wave output, UART output.

Pad1、 Pad15：Vin（input voltage 4～6V）

Pad2、Pad3、 Pad12：GND

Pad4：DAC2

Pad5：DAC1

Pad6：PWM output

Pad7、 Pad8、 Pad9：NC

Pad10、Pad13：UART（RXD） 0～3.3V digital input

Pad11、Pad14：UART（TXD） 0～3.3V digital output

3.1 Analog voltage output

DAC1 output voltage range (0～2.5V)，corresponding gas concentration（0～full detection range）

DAC2 output voltage range（0.4～2V)，corresponding gas concentration（0~full detection range）

3.2 PWM output

CO_2 output range:	0ppm-2000ppm
Allowed max. current for OC:	5mA maximum
Cycle:	1004ms±5%
High level output for beginning:	2ms (in name)
Middle of cycle:	1000ms±5%
Low level output for ending:	2ms（in name）

Account formula for CO_2 concentration which get through PWM:

$$C_{ppm} = 2000 \times (T_H - 2ms) / (T_H + T_L - 4ms)$$

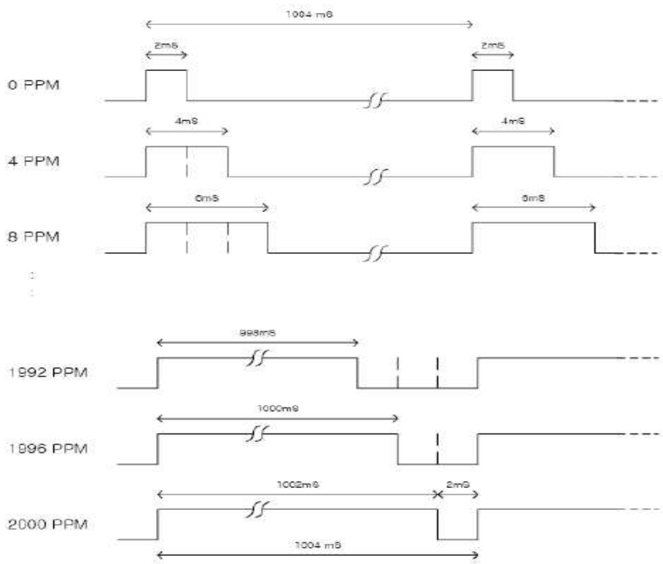
Among:

Cppm is calculated CO_2 concentration, unit is ppm;

TH is time for high level during an output cycle;

TL is time for low level during an output cycle.

3.3 Output for PWM:



4. UART communication protocol

Data obtain procedure and hardware serial communication

Baud rate: 9600, 8 digit data, 1 digit stop bit, No parity bit

9 byte for each frame data, initially with 0xff, ending with check value

Check value= （in reverse（DATA1+DATA2+……+DATA7））+1

1) 1、Read concentration and temperature value of the sensor

Below order would be sent when host send concentration value of the sensor:

0	1	2	3	4	5	6	7	8
Start bit 0XFF	Detector No.	order 0x86	00	00	00	00	00	Check value

Format of data returned by subsidiary detector:：

0	1	2	3	4	5	6	7	8
Start bit 0XFF	0x86	High channel	Low channel	Tem. channel				Check value

Gas Concentration = High channel*256+low channel, No.of sensor: 0x01

Environment tem. value = Tem.channel

2) When make zero calibration, send value: 0xff, 0x87, 0x87, 0x00, 0x00, 0x00, 0x00, 0x00, 0xf2

The first byte (0xff) is beginning byte, the second byte (0x87) is repeated order, the third byte (0x87) is order, the last five bytes is arbitrary value, while the last byte (0xf2) is check sum. No return information.

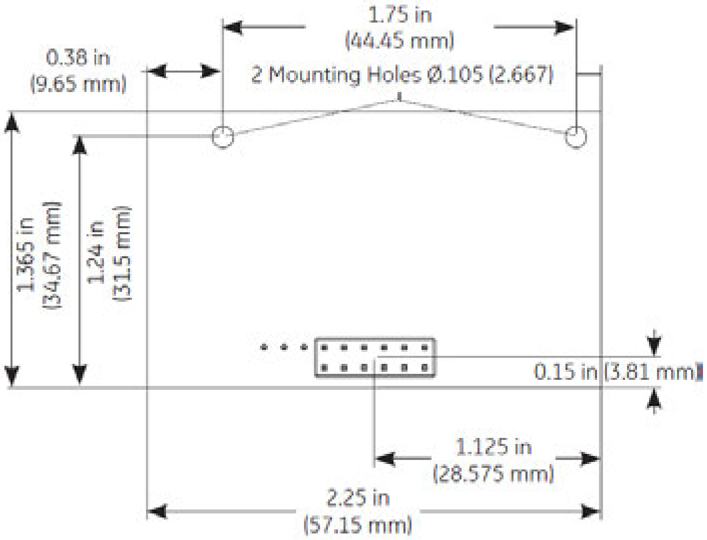
As it is CO2 sensor, please input Nitrogen gas for 5 minutes when make zero calibration.

3) When make span calibration, send value: 0xff, 0x88, 0x88, 0x00, 0x00, 0x00, 0x00, 0x00, 0xf0

The first byte (0xff) is beginning byte, the second byte (0x88) is repeated order, the third byte (0x88) is order, the fourth byte is span perch value, the fifth byte is span low value, last 3 bytes is arbitrary value, while the last byte (0xf0) is check sum. No return information.

5. Installation

There’s four installation holes

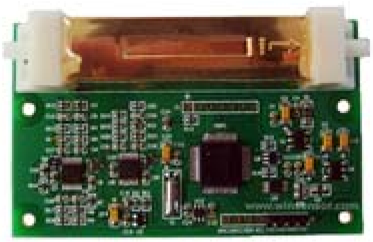


NDIR Gas sensor and modules MH-Z92

Brief Introduction:

MH-Z92 is a general and built-in sampling sensor, which adopts NDIR theory to measure concentration of CO2 and CH4 in air, it has good selectivity and long life. Temperature sensor inside could be for temperature compensation, it has both digital output and Analog voltage output.

Widely used for biogas, industrial and safety monitoring.



Technical Data:

Detected Gas	CO2 and CH4
Sampling method	Pump
Detection range	0-100%vol CH4 0-50%vol CO2
Error	<= +/-2% F.S
Response time	T90<10s
Working Temperature	0^oC -50^oC
Working Humidity	0-95%RH (No Dew)
Working voltage	3.5V-5.5V
Working current	Average 50mA, Max. 100mA
Lifetime	>5 years
Dimension	LXbXh, 83X51X18mm

MH-740A Intelligent Infrared Gas Sensor

User's Manual

Contents

1. Application

MH-740A sensor is a common intelligent sensor in small sized to detect CH_4 in air taking advantages of non-dispersive infrared principle. It has the advantages of good selectivity, oxygen independence, stable performance and long life. MH-740A is an infrared gas sensor in small sized combining mature technology for infrared absorption detection and micro-machining, superior circuit design closely. The sensor can be widely used in detection for fire disaster, explosive gases.

2. Product Mode & Explosion-proof Implication

Product Mode：MH-740A
Explosion Proof：ExmbⅡCT4

本产品生产制造依据：
GB3836.1-2000 《爆炸性气体环境用电气设备 第 1 部分：通用要求》
GB3836.9-2000 《爆炸性气体环境用电气设备 第 9 部分：浇封型“m”》
GB4208-93 《外壳防护等级（IP 代码)》
GB/T13384-92《机电产品包装应用技术条件》

3. Specification

Power Supply	4.5~5.5V DC
Working Current	120mA
Interface Power	5V
Detected Range	0~5%vol（0~100%vol can be specified）
Resolution	1%FSD
Warm-up Time	90s
Response Time	T90<30s
Repeatability	Zero < ±100ppm
	SPAN <±500ppm
Long-time Drift	Zero < ±300ppm/month
	SPAN < ±500ppm/month
Temperature	-40°C ~70°C
Humidity	0~95%RH
Life	>5 year
Explosion-proof	ExdmⅡCT4
Protection Class	IP6

4. Working Environment

Working Power：4.5~5.5V DC

Temperature：-40°C ~70°C

Humidity：0~95%RH

5. Structure

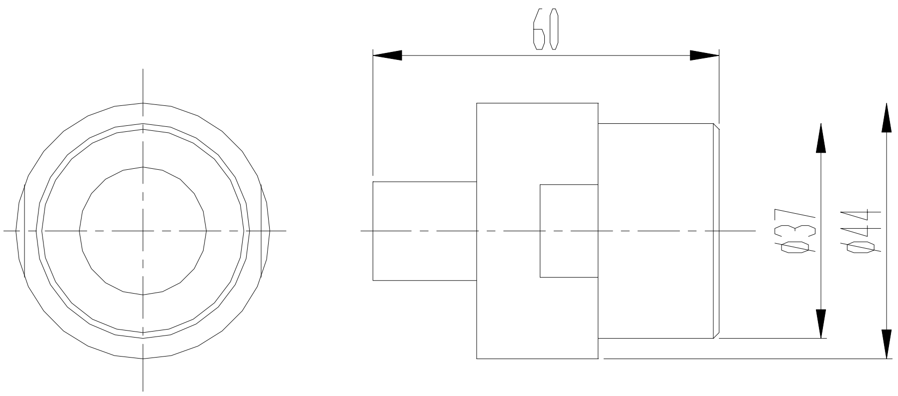


Fig 1：construction and dimension

5.1 Schematic Structure

5.2 Pin Definition

Red line：VCC

Yellow Line：SCL

Brown Line：SDA

Black Line：GND

5.3 Characteristics

High Sensitivity

5Vconstant power supply, low power consumption

Fast response and resume

Temperature compensation

Excellent stability

Long life

Anti-poisons

Water vapor-proof

6. Operation Instruction

6.1 Communication Protocol

MH-740A is communicated through IIC bus. Module works following slave mode and can be connected with MCU outside. The module address is 0xAA. The writing operation address is 0xAA and reading operation address is 0xAB. In the IIC communication, each frame data has 10 bytes, and the contents of the data vary according to different host command. The last byte of the data is checksum value. The recommended SCL clock frequency is less than 10K.

6.1.1 Writing Operation

The first sent byte of each writing operation is command byte. A full writing operation sequence is as below.

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command）→ receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum）→ receive acknowledge bit → send STOP signal

6.1.2 Reading Operation

A full reading operation sequence is as below.

send START signal → send module address（reading） → receive acknowledge bit → receive DATA0 → send acknowledge bit → receive DATA1 → receive acknowledge bit → …… → receive DATA9 （checksum）→ send non-acknowledge bit → send STOP signal

6.1.3 Checksum Value

Checksum Value =(converse（DATA0+DATA1+……+DATA8））+1

6.2 IIC communication Command

One writing operation must performance before reading operation. If the production data command for module is written, then performance reading operation, the production date can be read.

In the writing operation, the high-limit alarm value can be set through writing the high limit alarm value command (0x98) and high-limit alarm value together to the module.

For all the integer data, the higher is in front, the lower is in the post. Eg, DATA1 ~ DATA2 = high limit alarm concentration, it means DATA1 =the higher 8 bytes of high limit alarm concentration, ，DATA2 = the lower 8 bytes of high limit alarm concentration.

6.2.1 Version & Name of the Module Command: 0x90

send START signal → send module address（writing） → receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0x90

DATA1 ~ DATA8 is arbitrary value.

DATA9 = checksum（same with below）

send START signal → send module address（reading） → receive acknowledge bit → receive DATA0 → send acknowledge bit → receive DATA1 → send acknowledge bit → …… → receive DATA9 （checksum）→ send non-acknowledge bit → send STOP signal

DATA0 = version number

DATA1 = ID number

DATA2 = word ID number

DATA3 ~ DATA8 =sensor name

DATA9 = checksum (same with below）

6.2.2 Module Name 2, Command: 0x91

send START signal→ send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0x91

DATA1 ~ DATA8 is arbitrary value.

send START signal → send module address（reading） → receive acknowledge bit → receive DATA0 → send acknowledge bit → receive DATA1 → send acknowledge bit → …… → receive DATA9 （checksum）→send non-acknowledge bit → send STOP signal

DATA0 ~ DATA8 = sensor name 2

6.2.3 Module Production Date,　　Command: 0x92

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0x92

DATA1 ~ DATA8 is arbitrary value.

send START signal→ send module address（reading）→ receive acknowledge bit → receive DATA0 → send acknowledge bit → receive DATA1 → send acknowledge bit → …… → receive DATA9 （checksum）→ send non-acknowledge bit → send STOP signal

DATA0 ~ DATA6 = production date

DATA7 ~ DATA8 = validating date

6.2.4 Module Calibrating Date,　　Command: 0x93

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command）→ receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0x93

DATA1 ~ DATA8 is arbitrary value.

send START signal → send module address（reading）→ receive acknowledge bit → receive DATA0 → send acknowledge bit → receive DATA1 → send acknowledge bit → …… → receive DATA9 （checksum）→ send non-acknowledge bit → send STOP signal

DATA0 ~ DATA6 = calibrating date

DATA7 ~ DATA8 = validating date

6.2.5 Module Serial Number,　　Command: 0x94

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0x94

DATA1 ~ DATA8 is arbitrary value.

send START signal → send module address（reading）→ receive acknowledge bit → receive DATA0 → send acknowledge bit → receive DATA1 → send acknowledge bit → ……

→ receive DATA9 （checksum）→ send non-acknowledge bit → send STOP signal
DATA0 ~ DATA8 = serial number

6.2.6 Reading Alarm Value of Module, Command: 0x95

send START signal→ send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0x95

DATA1 ~ DATA8 is arbitrary value.

send START signal→ send module address（reading）→ receive acknowledge bit → receive DATA0 → send acknowledge bit → receive DATA1 → send acknowledge bit → …… → receive DATA9 （checksum）→ send non-acknowledge bit → send STOP signal

DATA0 ~ DATA1 = low-limit alarm value

DATA2 ~ DATA3 = high-limit alarm value

DATA4 ~ DATA5 = STEL alarm value

DATA6 ~ DATA7 = TWA alarm value

DATA8 = 0

6.2.7 Module Status, Command: 0x96

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0x96

DATA1 ~ DATA4 =current time (year, month, date, time)

DATA5 ~ DATA8 is arbitrary value.

send START signal → send module address（reading）→ receive acknowledge bit → receive DATA0 → send acknowledge bit → receive DATA1 → send acknowledge bit → …… → receive DATA9 （checksum）→ send non-acknowledge bit → send STOP signal

DATA0 = module status

DATA2 = unit

DATA3 = gas type

DATA4 = temperature

DATA5 ~ DATA6 = gas concentration

DATA7 ~ DATA8 = detected range

6.2.8 Low-limit alarm setting, Command: 0x97

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0x97

DATA1 ~ DATA2 = low-limit alarm value

DATA3 ~ DATA8 is arbitrary value.

6.2.9 High-limit alarm setting,　　Command: 0x98

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit →send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0x98

DATA1 ~ DATA2 = high-limit alarm value

DATA3 ~ DATA8 is arbitrary value.

6.2.10 STEL Alarm Setting,　　Command: 0x99

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit →send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0x99

DATA1 ~ DATA2 = STEL alarm value

DATA3 ~ DATA8 is arbitrary value.

6.2.11 TWA Alarm Setting　　Command: 0x9a

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0x99

DATA1 ~ DATA2 = TWA alarm value

DATA3 ~ DATA8 is arbitrary value.

6.2.12 Zero Calibration,　　Command: 0xa0

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0xa0

DATA1 ~ DATA8 is arbitrary value.

6.2.13 SPAN Calibration,　　Command: 0xaa

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0xaa

Data1~Data2: single calibration concentration

DATA3 ~ DATA8 is arbitrary value.

6.2.14 Decimal Bytes Reading,　　Command: 0x9b send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0x9b

DATA1 ~ DATA8 is arbitrary value.

send START signal → send module address（reading） → receive acknowledge bit → receive DATA0 → send acknowledge bit → receive DATA1 → send acknowledge bit → …… → receive DATA9 （checksum）→ send acknowledge bit → send STOP signal

DATA0 = decimal bytes

DATA1 ~ DATA8 is arbitrary value.

7. Notes for maintenance

7.1 The sensor should be calibrated regularly. The cycle time is better to be no more than 3 months.

7.2 Do not use the sensor in the high dusty environment for long time.

7.3 Please use the sensor with correct power supply.

7.4 Forbidden to cut the sensor pin.

8. Order Notes

Please provide the following information in order to purchase the specified product.

Detected range of sensor

Resolution of sensor

Sensor name: MH – 740A

MH-710 Intelligent Infrared Gas Sensor

User's Manual

1. Application

MH-710A sensor is a common intelligent sensor in small sized to detect CO2 in air taking advantages of non-dispersive infrared principle. It has the advantages of good selectivity, oxygen independence, stable performance and long life. MH-710A is an infrared gas sensor in small sized combining mature technology for infrared absorption detection and micro-machining, superior circuit design closely. The sensor can be widely used in detection for fire disaster, explosive gases.

2. Product Mode & Explosion-proof Implication

Product Mode：MH-710A
Explosion Proof：ExmbⅡCT4

GB3836.1-2000
GB3836.9-2000
GB4208-93
GB/T13384-92

3. Specification

Power Supply 4.5~5.5V DC

Working Current	120mA
Interface Power	5V
Detected Range	0~2000ppm（0~100%vol can be specified）
Resolution	5ppm
Warm-up Time	90s
Response Time	T90<30s
Deviation	30ppm±5%
Repeatability	Zero < ±10ppm
	SPAN <±30ppm
Long-time Drift	Zero < ±30ppm/month
	SPAN < ±30ppm/month
Temperature	-40°C ~70°C
Humidity	0~95%RH
Life	>5 years
Explosion-proof	ExdmⅡCT4
Protection Class	IP6

4. Working Environment

Working Power：4.5~5.5V DC

Temperature：-40°C ~70°C

Humidity：0~95%RH

5. Structure

5.1 Schematic Structure

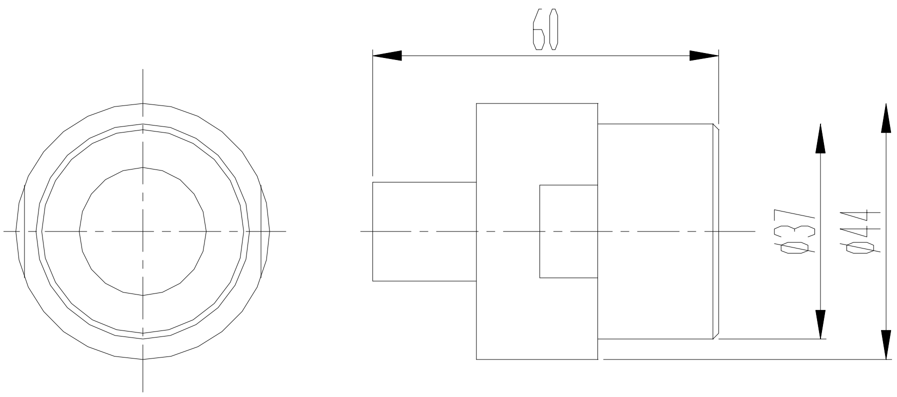


Fig 1：construction and dimension

5.2 Pin Definition

Red line：VCC
Yellow Line：SCL
Brown Line：SDA
Black Line：GND

5.3 Characteristics

High Sensitivity

5Vconstant power supply, low power consumption

Fast response and resume

Temperature compensation

Excellent stability

Long life

Anti-poisons

Water vapor-proof

6. Operation Instruction

6.1 Communication Protocol

MH-710A is communicated through IIC bus. Module works following slave mode and can be connected with MCU outside. The module address is 0xAA. The writing operation address is 0xAA and reading operation address is 0xAB. In the IIC communication, each frame data has 10 bytes, and the contents of the data vary according to different host command. The last byte of the data is checksum value. The recommended SCL clock frequency is less than 10K.

6.1.1 Writing Operation

The first sent byte of each writing operation is command byte. A full writing operation sequence is as below.

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

6.1.2 Reading Operation

A full reading operation sequence is as below.

A full reading operation sequence is as below.

send START signal → send module address（reading） → receive acknowledge bit → receive DATA0 → send acknowledge bit → receive DATA1 → receive acknowledge bit → …… → receive DATA9 （checksum） → send non-acknowledge bit → send STOP signal

6.1.3 Checksum Value

Checksum Value =(converse（DATA0+DATA1+……+DATA8））+1

6.2 IIC communication Command

One writing operation must performance before reading operation. If the production data command for module（0x92）is written, then performance reading operation, the production date can be read.

In the writing operation, the high-limit alarm value can be set through writing the high limit alarm value command (0x98) and high-limit alarm value together to the module.

For all the integer data, the higher is in front, the lower is in the post. Eg, DATA1 ~ DATA2 = high limit alarm concentration, it means DATA1 =the higher 8 bytes of high limit alarm concentration, ，DATA2 = the lower 8 bytes of high limit alarm concentration.

6.2.1 Version & Name of the Module Command: 0x90

send START signal → send module address（writing） → receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0x90

DATA1 ~ DATA8 is arbitrary value.

DATA9 = checksum（same with below）

send START signal → send module address（reading） → receive acknowledge bit → receive DATA0 → send acknowledge bit → receive DATA1 → send acknowledge bit → …… → receive DATA9 （checksum）→ send non-acknowledge bit → send STOP signal

DATA0 = version number

DATA1 = ID number

DATA2 = word ID number

DATA3 ~ DATA8 =sensor name

DATA9 = checksum (same with below）

6.2.2 Module Name 2,   Command: 0x91

send START signal→ send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0x91

DATA1 ~ DATA8 is arbitrary value.

send START signal → send module address（reading） → receive acknowledge bit → receive DATA0 → send acknowledge bit → receive DATA1 → send acknowledge bit → …… → receive DATA9 （checksum）→send non-acknowledge bit → send STOP signal

DATA0 ~ DATA8 = sensor name 2

6.2.3 Module Production Date,   Command: 0x92

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0x92

DATA1 ~ DATA8 is arbitrary value.

send START signal→ send module address（reading）→ receive acknowledge bit → receive DATA0 → send acknowledge bit → receive DATA1 → send acknowledge bit → …… → receive DATA9 （checksum）→ send non-acknowledge bit → send STOP signal

DATA0 ~ DATA6 = production date

DATA7 ~ DATA8 = validating date

6.2.4 Module Calibrating Date,   Command: 0x93

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command）→ receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0x93

DATA1 ~ DATA8 is arbitrary value.

send START signal → send module address（reading）→ receive acknowledge bit → receive DATA0 → send acknowledge bit → receive DATA1 → send acknowledge bit → ……

→ receive DATA9 （checksum）→ send non-acknowledge bit → send STOP signal

DATA0 ~ DATA6 = calibrating date

DATA7 ~ DATA8 = validating date

6.2.5 Module Serial Number, Command: 0x94

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0x94

DATA1 ~ DATA8 is arbitrary value.

send START signal → send module address（reading）→ receive acknowledge bit → receive DATA0 → send acknowledge bit → receive DATA1 → send acknowledge bit → …… → receive DATA9 （checksum）→ send non-acknowledge bit → send STOP signal

DATA0 ~ DATA8 = serial number

6.2.6 Reading Alarm Value of Module, Command: 0x95

send START signal→ send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0x95

DATA1 ~ DATA8 is arbitrary value.

send START signal→ send module address（reading）→ receive acknowledge bit → receive DATA0 → send acknowledge bit → receive DATA1 → send acknowledge bit → …… → receive DATA9 （checksum）→ send non-acknowledge bit → send STOP signal

DATA0 ~ DATA1 = low-limit alarm value

DATA2 ~ DATA3 = high-limit alarm value

DATA4 ~ DATA5 = STEL alarm value

DATA6 ~ DATA7 = TWA alarm value

DATA8 = 0

6.2.7 Module Status, Command: 0x96

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0x96

DATA1 ~ DATA4 =current time (year, month, date, time)

DATA5 ~ DATA8 is arbitrary value.

send START signal → send module address（reading）→ receive acknowledge bit → receive DATA0 → send acknowledge bit → receive DATA1 → send acknowledge bit → …… → receive DATA9 （checksum）→ send non-acknowledge bit → send STOP signal

DATA0 = module status

DATA2 = unit

DATA3 = gas type

DATA4 = temperature
DATA5 ~ DATA6 = gas concentration
DATA7 ~ DATA8 = detected range

6.2.8 Low-limit alarm setting,   Command: 0x97

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal
DATA0 = 0x97
DATA1 ~ DATA2 = low-limit alarm value
DATA3 ~ DATA8 is arbitrary value.

6.2.9 High-limit alarm setting,   Command: 0x98

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit →send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal
DATA0 = 0x98
DATA1 ~ DATA2 = high-limit alarm value
DATA3 ~ DATA8 is arbitrary value.

6.2.10 STEL Alarm Setting,   Command: 0x99

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit →send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal
DATA0 = 0x99
DATA1 ~ DATA2 = STEL alarm value
DATA3 ~ DATA8 is arbitrary value.

6.2.11 TWA Alarm Setting   Command: 0x9a

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal
DATA0 = 0x99
DATA1 ~ DATA2 = TWA alarm value
DATA3 ~ DATA8 is arbitrary value.

6.2.12 Zero Calibration,   Command: 0xa0

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal

DATA0 = 0xa0
DATA1 ~ DATA8 is arbitrary value.

6.2.13 SPAN Calibration, Command: 0xaa

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal
DATA0 = 0xaa
Data1~Data2: single calibration concentration
DATA3 ~ DATA8 is arbitrary value.

6.2.14 Decimal Bytes Reading, Command: 0x9b

send START signal → send module address（writing）→ receive acknowledge bit → send DATA0（command） → receive acknowledge bit → send DATA1 → receive acknowledge bit → …… → send DATA9（checksum） → receive acknowledge bit → send STOP signal
DATA0 = 0x9b
DATA1 ~ DATA8 is arbitrary value.
send START signal → send module address（reading） → receive acknowledge bit → receive DATA0 → send acknowledge bit → receive DATA1 → send acknowledge bit → …… → receive DATA9 （checksum）→ send acknowledge bit → send STOP signal
DATA0 = decimal bytes
DATA1 ~ DATA8 is arbitrary value.

7. Notes for maintenance

7.1 The sensor should be calibrated regularly. The cycle time is better to be no more than 3 months.
7.2 Do not use the sensor in the high dusty environment for long time.
7.3 Please use the sensor with correct power supply.
7.4 Forbidden to cut the sensor pin.

8. Order Notes

Please provide the following information in order to purchase the specified product.
Detected range of sensor
Resolution of sensor
Sensor name: MH – 710A

9. Contact

Address: No. 299, Jinsuo Road, National Hi-Tech Zone, Zhengzhou 450001, Henan
Tele: (86)0371-60932955 60932966 60932977
Fax: (86) 0371-60932988

Mail: sales@winsensor.com