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

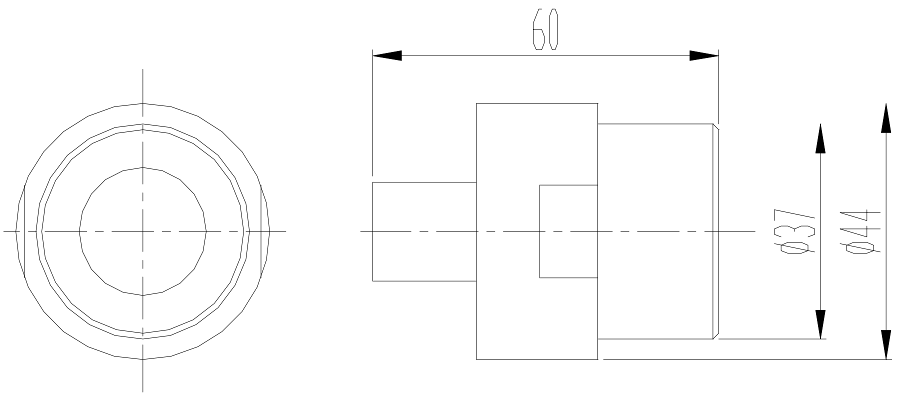

Fig 1：construction and dimension

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