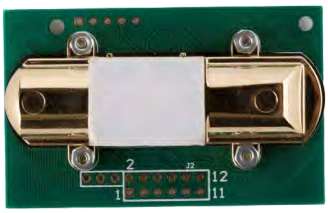
1% CO2 Sensor DataSheet

CO2Meter.com
CO2 Measurement Specialists

WIN-0005 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.

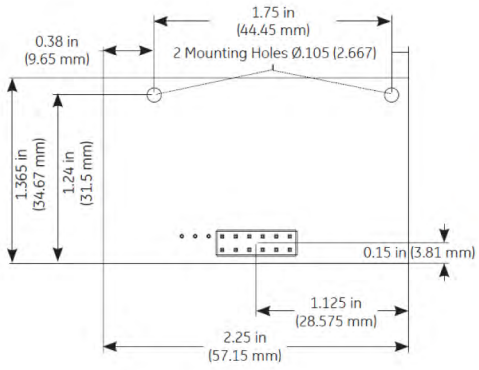


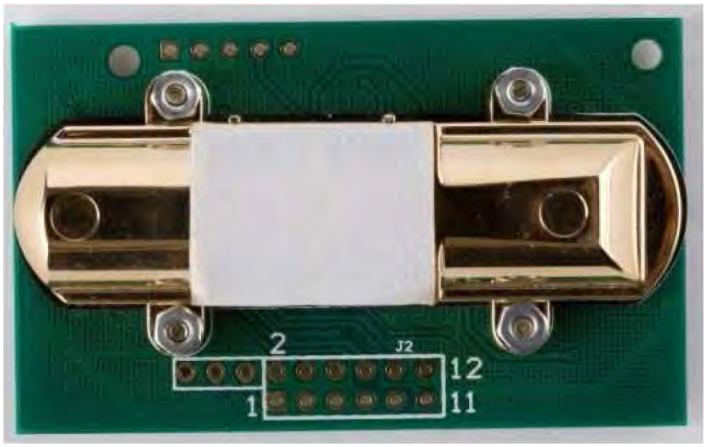
The WIN-0005 is a drop-in replacement for the Telaire T6615 Sensor Module.

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	4.5 ~ 5.5V DC
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

CO2 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 CO2 concentration which get through PWM:

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

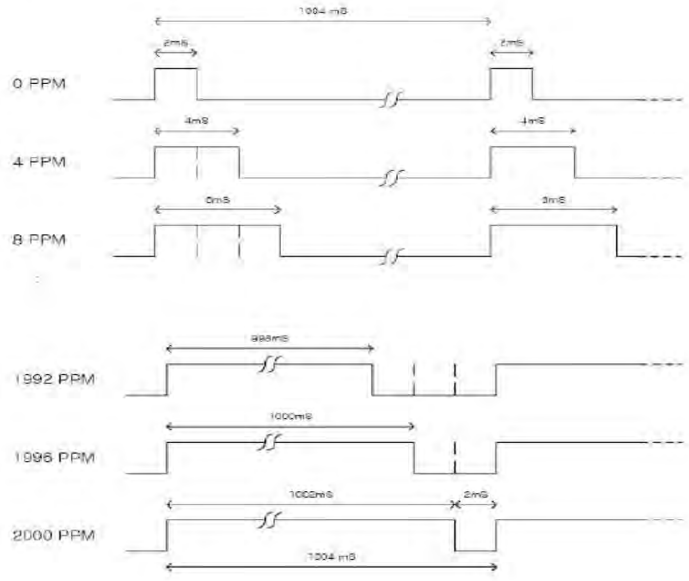
Among:

Cppm is calculated CO2 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.

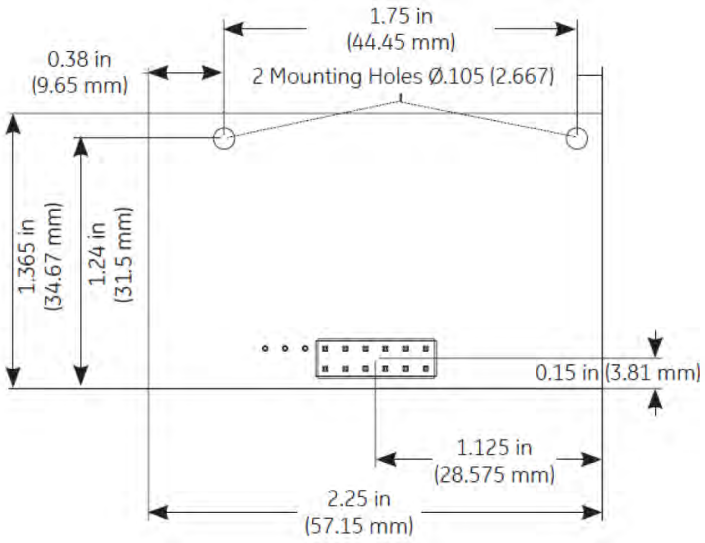
As it is CO2 sensor, please input Nitrogen gas for 5 minutes when make zero calibration.

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



6. Serial Host Protocol

6.1 General Settings

Baud Rate	9600
Data Bits	8
Stop Bits:	1
Parity(check bits):	0

6.2 Commands

Each command or Request:
-Consists of 9 bytes (byte0-byte8)
-Has the Sensor # in byte-1 (factory default is 0x01)
-End with Checksum (See page 5 for Checksum calculation)

0x86	Read CO_2 concentration
0x87	Calibrate CO_2 to zero
0x88	Calibrate CO_2 to known concentration

6.3 Read Co2 Concentration

Request								
Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
Start Byte	Sensor #	Command	-	-	-	-	-	Checksum
0XFF	0x01	0x86	0x00	0x00	0x00	0x00	0x00	0x79

Response(example)								
Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
Start Byte	Command	Concentration (High Byte)	Concentration (Low Byte)	-	-	-	-	Checksum
0XFF	0x86	0x02	0x60	0x47	0x00	0x00	0x00	0xD1

Percentage= (256 x High Byte) + Low Byte

6.4 Calibrate Co2 to Zero

Request								
Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
Start Byte	Sensor #	Command	-	-	-	-	-	Checksum
0XFF	0x01	0x87	0x00	0x00	0x00	0x00	0x00	0x78

6.5 Calibrate Sensor Span

Request								
Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
Start Byte	Sensor #	Command	Concentration (High Byte)	Concentration (Low Byte)	-	-	-	Checksum
0XFF	0x01	0x88	0x07	0xD0	0x00	0x00	0x00	0xA0

Concentration (High Byte) =concentration/256

Concentration (Low Byte) =concentration - concentration/256

6.6 Calibrate CO2 to Known Concentration

Calculating Checksum

The checksum is calculated as (NOT(Data1 + Data2 + Data 3+……Data7))+1

Example: Read Co2 Concentration

Request								
Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
Start Byte	Sensor #	Command	-	-	-	-	-	Checksum
0XFF	0x01	0x86	0x00	0x00	0x00	0x00	0x00	0x79

1. Add up all of the bytes 1-7 using 8-bit addition (byte-0 is NOT used in checksum calculation)
   0x1 + 0x86 + 0 + 0 + 0 + 0 + 0 = 0x87
2. We then get the Inverse or 'NOT' of the sum from step 1
   0xff – 0x87 = 0x78
3. Add 1 to the result
   0x78 + 0x01 = 0x79

6.7 C# Example

```
char getCheckSum(char *packet)
{
      char i, checksum;
      for( i = 1; i < 8; i++)
      {
            checksum += packet[i];
      }
      checksum = 0xff – checksum;
      checksum += 1;
      return checksum;
}
```


131 Business Center Drive
Ormond Beach, FL 32174
(386) 872-765 Sales
(386) 256-4910 Support
www.CO2Meter.com
Sales@CO2Meter.com