

Product Specification

Product:	Digital Carbondioxide Sensor
Product model:	C8D

深圳市深晨科技有限公司
Shenzhen Shenchen Technology Co., Ltd

1. Main characteristics

- ◆ NDIR
- ◆ Double wavelength
- ◆ Industrial-grade Precision
- ◆ Minuscule Size



2. Overview

C8Dis a kind of digital NDIR sensor, which can acquire and calculate the concentration of carbon dioxide in the air continuously and output them in the form of digital interface. This sensor can be inserted into variable instruments related to the concentration of carbon dioxide in the air or other environmental improvement equipment to provide correct concentration data in time.

3. Working principle

NDIR gas sensor principle of work is according to the different gas molecules for absorption characteristic of near infrared spectra, by gas concentration and absorption strength relationship (Lambert-Beer law) analysis and determine the concentration of the gas.

This sensor adopts the non-spectrophotometric infrared absorption principle and adopts the design structure of single air chamber and single receive channels, it has the characteristic of small structure and high detection precision.

4. Technical Index

Parameter	Index	Unit
Accuracy effective Range (Keep Consistency Error)	0~2000/ 0~5000/ 0~10000	ppm
Maximum Range	0~10000	ppm
Resolution	1	ppm
Maximum Consistency Error (400~5000ppm)	$\pm (50\text{ppm} + 3\% \text{ of Reading})$	
Response Time (Environmental change)	<3 minutes for 90% step change typical	Minute(min)
Single Update	<3 (Typical 1s)	Second(s)
Update or Warmup Time	8s(update time) <25s (operational) <3min(90% accuracy) <10min (maximum accuracy)	
DC Power Supply	Typ:5.0 Min:4.5 Max: 5.5	Volt(V)
Active Current	150mA peak ,70mA average	Milliampere(mA)
Interface Level	L <0.8 @3.3 H >2.7@3.3	Volt(V)
Working Temperature Range	-10~+60	°C
Working Humidity Range	0~95% non condensed	
Storage Temperature Range	-40~+85	°C
Maximum storage Temperature & Humidity	70/85	°C/%
Life Time	≥ 10	Year(Y)
Physical Size	33×21.7×10	Millimeter(mm)

5. Pin Definition

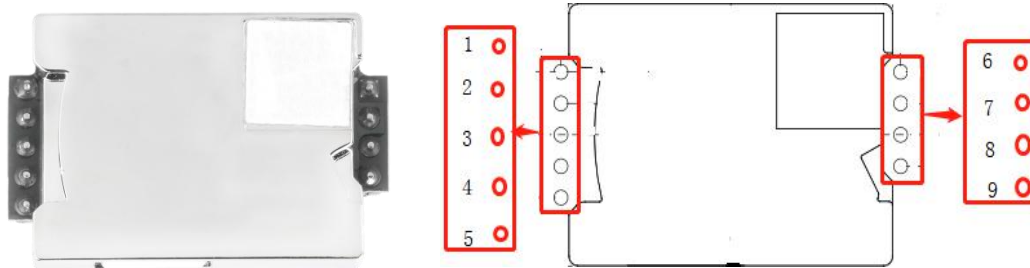
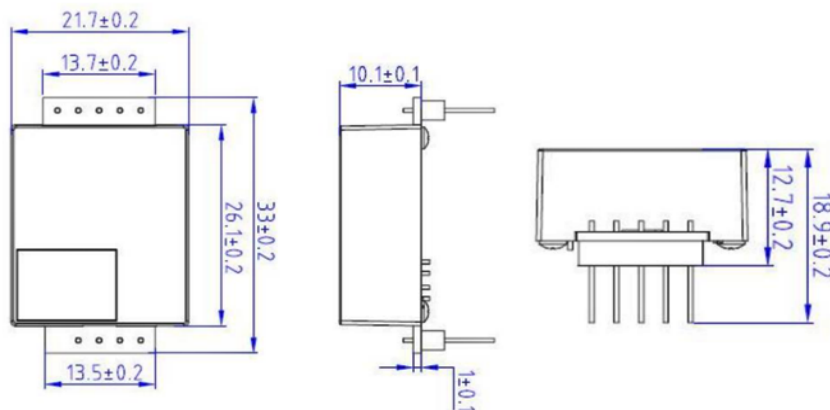


Figure 1 Connector Definition

CON5	①	NC	
	②	RX	UART Rx (input to sensor) TTL level@3.3V
	③	TX	UART TX (output from sensor) TTL level@3.3V,
	④	NC	
	⑤	NC	
CON4	⑥	+5V	Positive power 5V
	⑦	GND	Negative power
	⑧	NC	
	⑨	PWM	PWM output at approximately 1Hz

6. Physical Size

Unit:(mm)



7. PWM OUTPUTS

PWM OUTPUTS

In order to convert the pulse to a reading in PPM, the user should use the following equation:

$$C_{ppm} = (T_H - 2ms) * 5 \quad \text{for } 0 - 5000 \text{ p}$$

pm models

The slow ~1Hz PWM output option allows the user to measure the duration of the pulse, and correlate this to a CO2 measurements.

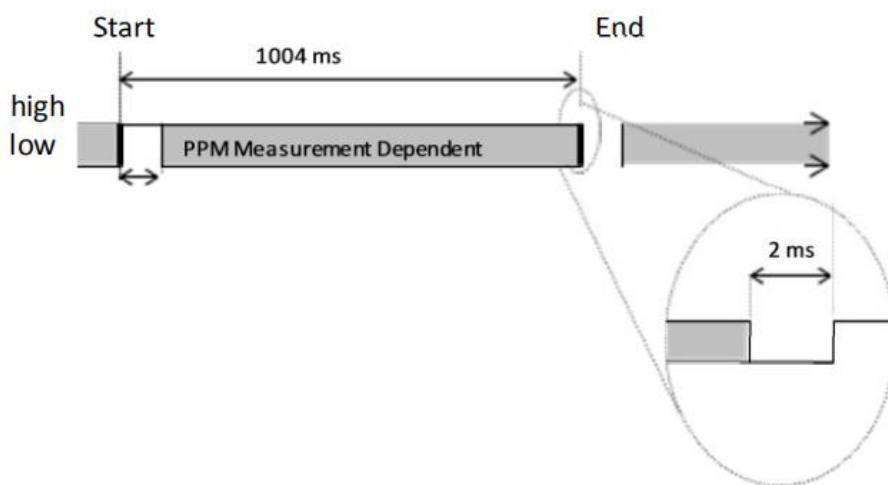
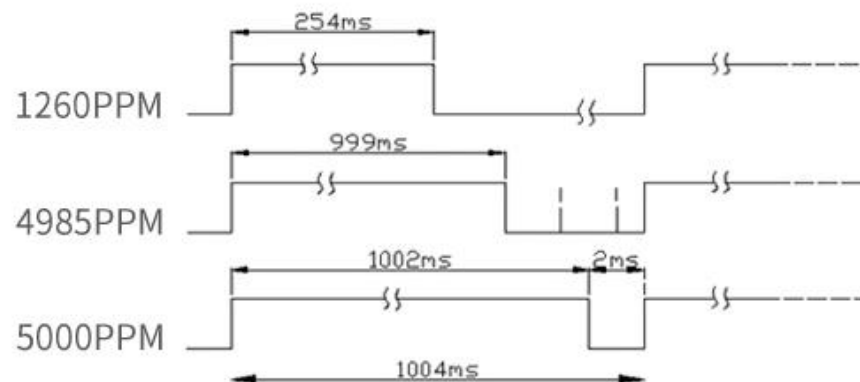


Figure 5 - Details of ~1Hz PWM



8. UART Serial Communications Interface

UART interface at 9600 baud, 8 data bits

There are two output modes of C8 sensor: active output and query output. After a query, the active output stops and needs to be powered up again before recovery.

After power on, the sensor actively outputs data: output frequency 1S/time.

Output Format: 16BYTE

Data Header: BYTE0 = 0X42; BYTE1=4D

BYTE 6 high degree, BYTE7 low degree, is the concentration of CO2

BYTE 15, data checksums. $BYTE15 = BYTE0 + BYTE1 + \dots + BYTE14$;

For example,: 42 4D 0C 51 09 A2 07 2B 01 35 05 81 20 08 00 AD;

The concentration of CO2= $BYTE6 \times 256 + BYTE7 = 07 \times 256 + 2B = 1835$;

The query to read:

Read CO2 ppm, Reply 522 ppm

Send: 64 69 03 5E 4E

Reply: 64 69 03 01 0A 02 00 00 00 00 00 00 9B F0

Reply 14 BYTE.

BYTE4, BYTE5 is the concentration of CO2, Convert to base 10 is the concentration ,BYTE5 is high degree, BYTE4 is low degree

0x0A is 16bit Integer low byte, 0x02 is high byte ,add up to 522

BYTE12,BYTE13 is CRC testing data

Send a query command, the sensor stops active output , send a query, output a data.

Co2 concentration values for single-point calibration

Send : 110,303 DF1 DF2 CS

Reply: 16 01 03 E6

Function: Single-point calibration of CO2 concentration value

Explain:

1. Single-point calibrated target value = $DF1 \times 256 + DF2$. In ppm, with a range of (400 to 1500 p p m)
2. Before performing the CO2 single-point calibration, please confirm that the current environmental CO2 value is a single-point calibration target value, and the minimum stability time is more than at least 2 minutes.

Example:

When you need to calibrate the module single point to 600ppm, send a command: 11 03 03 02 58 8F

Headecimal conversion to decimal: 02 is 02; 58 is 88

Co2 measurements = $02 \times 256 + 88 = 600\text{ppm}$

CRC calculate

```
uint16_t CO2ModbusComm::modbus_calcuCRC(uint8_t *dataarray, uint16_t datalen)
{
    uint8_t uchCRCHi = 0xFF; /* CRC High byte initialization*/
    uint8_t uchCRCLo = 0xFF; /* CRC Low byte initialization*/
    uint16_t uIndex; /* CRC Query table index*/
    uint16_t crc;

    const uint8_t auchCRCHi[] = {
        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,
        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, 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, 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
    };

    const uint8_t auchCRCLo[] = {
        0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4,
        0x04, 0xCC, 0x0C, 0x0D, 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, 0x6A, 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
    };

    while (datalen--) /* Complete the entire message buffer*/
    {
        uIndex = uchCRCLo ^ *dataarray++; /* count CRC */
        uchCRCLo = uchCRCHi ^ auchCRCHi[uIndex];
        uchCRCHi = auchCRCLo[uIndex];
    }
    crc = (uint16_t)uchCRCHi * 256;
    crc += (uint16_t)uchCRCLo;
    return crc;
}
```

9. Pre-caution items while sensor handling

Transport, handling or assembly may affect calibration. If abnormal data is found during installation or use, it can be adjusted back to the normal value by automatic calibration (continue working 24hours) .

10. Attention

- 10.1 The metal-plated case is connected with the internal ground, and it should not be short connected with other external panel circuit or chassis case, so as to avoid contact with metal parts and other voltage values, so as not to affect the work of the sensor or the breakdown of the sensor.
- 10.2 Since the sensor generates part of the heat during operation, the main body temperature is higher than the ordinary indoor ambient temperature, which can be kept constant by itself to realize data stability. Due to the strong thermal conductivity of the metal structure, if the direct contact with the sensor shell will cause the heat dissipation to break the original balance too quickly, thus affecting the data reliability, please avoid the metal parts contact with the sensor.
- 10.3 The sensor is a thermal sensitive device and adopts the way of diffused gas exchange. If there is a faster airflow above the sensor, the data reliability will be affected. When applied to the purifier, it should be avoided to be directly placed in the air duct of the purifier. If it cannot be avoided, a separate structure space should be set up to isolate the sensor from the air duct of the purifier.
- 10.4 The sensor is an integral component, which should not be disassembled by the user, including the gas shell and the lower metal shielding shell, in case of irreversible damage.