

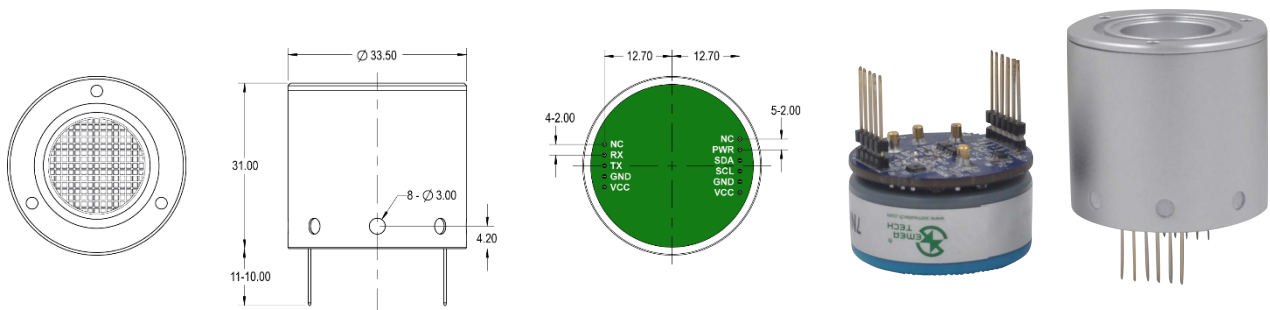
## 7 SMART SENSOR MODULE

### Description

This 7 Smart Sensor Module consists of a data collection and processing PCB assembly, a SemeaTech 7-Series electrochemical (EC) sensor, and a metal enclosure. The PCB assembly in the module collects the data from the gas sensor output and then processes it with amplification, sampling, filtration, and compensation through a built-in MCU to deliver stable and accurate digital output reflecting the actual target gas concentration. Any of SemeaTech 7-series EC sensors and 4-electrode EC sensors can be used to form a 7 Smart Sensor Module to detect the target gas with the digital output. This module provides the convenience and friendly user experience for users to quickly integrate gas sensors into their existing systems for a variety of gas detection applications.

- Unified electrical interface, mechanical dimensions and communication protocol;
- Working with all SemeaTech 7-series EC sensors and 4-electrode EC sensors;
- Built-in temperature and humidity sensors for better environmental independence;
- Multiple output interfaces, including USART, and I<sup>2</sup>C;
- Built-in power switch controllable through programming.
- Programmable I<sup>2</sup>C address for user friendly interfaces.

### Product Dimensions

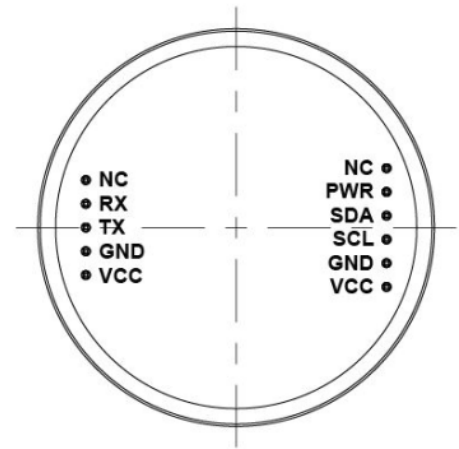


### Specifications

<b>Design for</b>	SemeaTech 7-Series EC Sensors	<b>Installation</b>	11 pins, plug-in
<b>Detection principle</b>	Electrochemical	<b>Operating temperature</b>	-20°C ~ +55°C
<b>Detection range</b>	Refer to sensor data sheet	<b>Operating humidity</b>	0%~90% RH Non-condensing
<b>Detection resolution</b>	Refer to sensor data sheet	<b>Operating pressure</b>	1 atm ±10%
<b>Measurement error</b>	Refer to sensor data sheet	<b>Enclosure material</b>	Aluminum alloy
<b>Operating voltage</b>	3.2 ~ 5.5 VDC	<b>Dimensions</b>	φ33.5x31 mm
<b>Operating current</b>	≤ 5mA @5V	<b>Weight</b>	30 g
<b>Output mode</b>	USART (3.3V TTL) I <sup>2</sup> C (3.3V TTL)	<b>Expected operating life</b>	PCBA: 5 years Sensor: Refer to sensor data sheet

## Pin Definitions

Pin	Function	Description
1	NC	Reserved pins (suspended)
2	RX	Serial port receiving
3	TX	Serial port sending
4	GND	0V-Ground
5	VCC	3.2-5.5 VDC
6	NC	Reserved pins (suspended)
7	PWR	Module power enable
8	SDA	I2C signal SDA
9	SCL	I2C signal SCL
10	GND	0V-Ground
11	VCC	3.2-5.5 VDC



\*Note: Two VCC signals are connected internally.

## USART Communication Protocol

### 1. Settings

Start bit – 1      Data bit – 8      Stop bit – 1      Check bit – None      Baud rate – 115200 bps

### 2. Frame Format (The format of each communication frame)

Header	Device Code	Function Code	Starting address	Data Length	Data	Check bit
H	ID	F	A	N	D	CRC16

H – 1Byte, fixed as 0x3A

ID – 1Byte, defaults as 0x10 and can be customized by users

F – 1Byte, for example (0x03)

A – 2Byte, for example (0x0001)

N – 1Byte, in two bytes, for example (0x02: 4 bytes)

D – N \* 2Byte, Big Endian for example (MSB LSB) is defined as signed short

CRC16 – 2Byte, using MODBUS\_CRC16 checking algorithm (see Appendix 1 for details)

### 3. Command Description

#### 3.1 Sensor type reading

Request from host device

Header	Device Code	Function Code	Starting address	Data Length	Data	Check
0x3A	0x10	0x01	0x0000	0x01	0x0000	0x82B0

Module response for correct data receiving

Header	Device Code	Function Code	Data	Check
0x3A	0x10	0x01	D (1byte Data)	CRC16

**D: Sensor code**

0: Not define 2: CO 3: O2 4: H2 5: CH4 6: C3H8 7: CO2 8: O3 9: H2S  
 10: SO2 11: NH3 12: CL2 13: ETO 14: HCL 15: PH3 16: HBR 17: HCN  
 18: ASH3 19: HF 20: BR2 21: NO 22: NO2 23: NOX 24: CLO2 25: SIH4  
 26: CS2 27: F2 28: B2H6 29: GEH4 30: N2 31: THT 32: C2H2 33: C2H4  
 34: CH2O 35: LPG 36: HC 37: C6H6 38: H2O2 39: CH3SH 40: C2H3CL  
 For example: 3A 10 01 0D CD 6C i.e. D=0x0D=13 It is ETO sensor

**3.2 Sensor data reading (unit: ug/m3)**

Request from host device

Header	Device Code	Function Code	Starting address	Data Length	Data	Check
0x3A	0x10	0x03	0x0000	0x02	0x0000	0x7352

Module response for correct data receiving

Header	Device Code	Function Code	Starting address	Data Length	Data	Check
0x3A	0x10	0x03	0x0000	0x02	D	CRC16

D: Received data, 4Byte, Big Endian

For example: 3A 10 03 00 00 02 00 00 00 5E 25 35  
 Sensor value (ug / m<sup>3</sup>): 00 00 00 5E i.e. 94ug / m<sup>3</sup>

**3.3 Sensor data reading (ppb)**

Request from host device

Header	Device Code	Function Code	Starting address	Data Length	Data	Check
0x3A	0x10	0x03	0x0002	0x02	0x0000	0x72EA

Module response for correct data receiving

Header	Device Code	Function Code	Starting address	Data Length	Data	Check
0x3A	0x10	0x03	0x0002	0x02	D	CRC16

D: Received data, 4Byte, Big Endian

For example: 3A 10 03 00 02 02 00 00 00 4C A4 DA  
 Sensor value (ppb): 00 00 00 4C i.e. 76ppb

**3.4 Sensor temperature data reading (°C)**

Request from host device

Header	Device Code	Function Code	Starting address	Data Length	Data	Check
0x3A	0x10	0x03	0x0004	0x01	0x0000	0x8262

Module response for correct data receiving

Header	Device Code	Function Code	Starting address	Data Length	Data	Check
0x3A	0x10	0x03	0x0004	0x01	D	CRC16

D: Received data, 2Byte, Big Endian and divided by 100 to get the temperature value

For example: (D = 0x0A3D = 2621 to get the temperature of 26.21 °C)

**3.5 Sensor humidity data reading (%RH)**

Request from host device

Header	Device Code	Function Code	Starting address	Data Length	Data	Check
0x3A	0x10	0x03	0x0005	0x01	0x0000	0x3292

Module response for correct data receiving

Header	Device Code	Function Code	Starting address	Data Length	Data	Check
0x3A	0x10	0x03	0x0005	0x01	D	CRC16

D: Received data, 2Byte, Big Endian and divided by 10000 to get the percentage humidity value  
 For example: (0x14 89 = 5257 to get the humidity of 52.57%)

### 3.6 Multiple parameters reading (address 0000-0005)

Request from host device

Header	Device Code	Function Code	Starting address	Data Length	Data	Check
0x3A	0x10	0x03	0x0000	0x06	0x0000	0x3293

Module response for correct data receiving

Header	Device Code	Function Code	Starting address	Data Length	Data	Check
0x3A	0x10	0x03	0x0000	0x06	D	CRC16

D: Received data, 12Byte

For example: 3A 10 03 00 00 06 00 00 00 8F 00 00 00 50 0A 70 16 11 35 96

Sensor value (ug/ m<sup>3</sup>): 00 00 00 8F ;

Sensor value (ppb): 00 00 00 50;

Temperature: 0A 70; Humidity: 16 11

### 3.7 Check error response

Header	Device Code	Function Code	Data	Check
0x3A	0x10	0x08	0x00	CRC16

For example: 3A1008000AF9

### 3.8 Zero Calibration

Request from host device

Header	Device Code	Function Code	Starting address	Data Length	Data	Check
0x3A	0x10	0x07	0x0000	0x01	0x0000	0x82D6

Module response for correct data receiving

Header	Device Code	Function Code	Starting address	Data Length	Data	Check
0x3A	0x10	0x07	0x0000	0x01	D	CRC16

D: 2Byte data

For example: 3A 10 07 00 00 01 04 7A 01 F5

Caution: Make sure the module in the zero-gas environment at least 5 minutes before calibration

### 3.9 Gas Calibration

Request from host device

Header	Device Code	Function Code	Starting address	Data Length	Data	Check
0x3A	0x10	0x09	0x0000	0x01	D	0x82D6

D: 2Byte gas concentration data, Big Endian, Unit: ppm, the value should be lower than 255ppm  
 For example: 3A 10 09 00 00 01 00 0A 03 FF i.e.: D=0x000A 10ppm gas is used for calibration

Module response for correct data receiving

Header	Device Code	Function Code	Data	Check
0x3A	0x10	0x09	0x00 (Success) 0x01 (Processing) 0x02 (Fail)	CRC16

For example: 3A 10 09 01 CAA9 (Calibration processing)

Caution: The calibration process for environmental detection is about 300 seconds, and for industry usage is about 60 seconds, please wait for the module to respond before starting the command

## I<sup>2</sup>C Communication Protocol

### 1. I<sup>2</sup>C Interface

Parameter	Definition	State	Min	Max	Unit
f <sub>ck</sub>	I2C clock frequency	Slave Mode		100	kHz
t <sub>su</sub>	Data input setup time	Slave Mode	6.5		ns
t <sub>h</sub>	Data input Holding Time	Slave Mode	15.5		ns

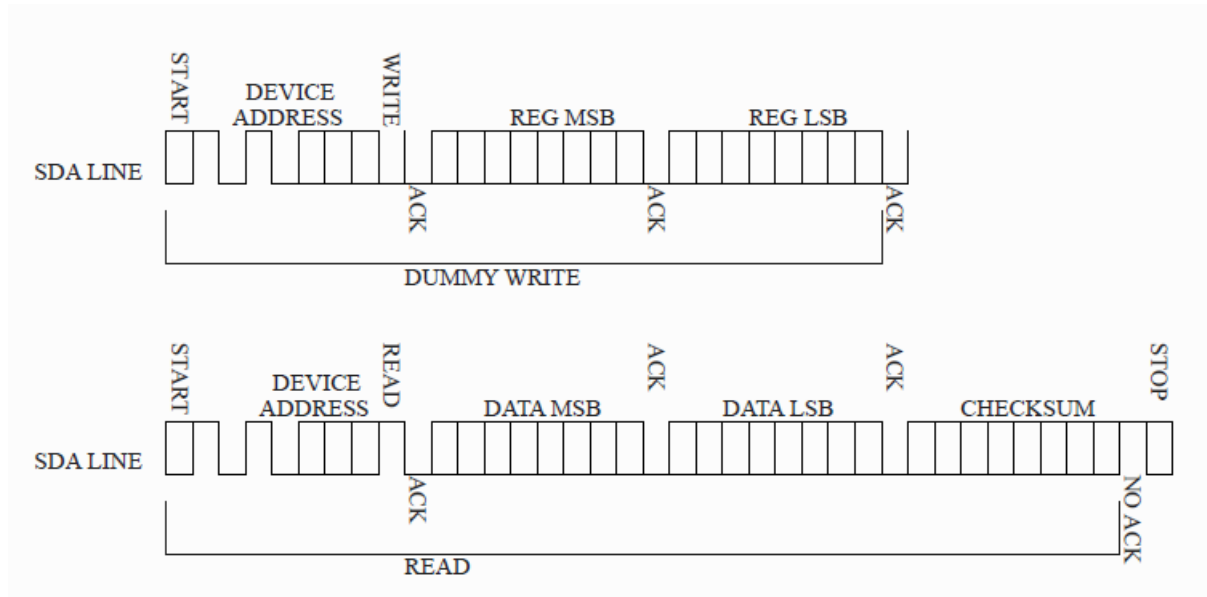
### 2. Slave Device Address

Slave device addresses can be customized by software tools

Default setting:

CO:	0	0	0	0	0	0	1	R/W
O3:	0	0	0	0	1	0	0	R/W
SO2:	0	0	0	0	1	0	1	R/W
NO2:	0	0	0	1	0	1	1	R/W

### 3. I<sup>2</sup>C Communication Protocol



CHECKSUM is cumulative and retrieve check. See Appendix 2 for details.

#### 4. Data Analysis

The REG parameters are shown below, which is the data address in the module.

REG	MSB	LSB
Sensor value (ug/m <sup>3</sup> )	0x00	0x00
Sensor value (ppb)	0x00	0x02
Temperature	0x00	0x04
Humidity	0x00	0x05

Data examples:

DATA	MSB	LSB	CHECKSUM	Actual value	Remarks
Sensor value (ug/m <sup>3</sup> )	0x0000	0x0005	0xFA	5ug/m <sup>3</sup>	Same with USART
Sensor value (ppb)	0x0000	0x0005	0xFA	5ppb	
Temperature	0x0A	0x3D	0xB8	26.21°C	Data Conversion
Humidity	0x14	0x89	0x62	52.57%	Method

Caution: To receive 5Byte data when reading sensor value, and to receive 3Byte data when reading temperature and humidity

### Warning!

- 1) This product does not have any intrinsic safety certification or explosion proof certification. Please do NOT use this product in any hazardous locations.
- 2) This product does not have reverse power protection and Electrostatic Discharge (ESD) protection. Please carefully verify the electrical polarity and make the ESD protection before each use or installation.
- 3) Please use a stable DC power supply for this gas sensor module. It is highly recommended to use a power supply with the output voltage fluctuation less than 1%.

### Appendix 1: MODBUS CRC16 algorithm

```
unsigned short modbus_CRC16(unsigned char *ptr, unsigned char len)
{
    unsigned short wrcrc=0XFFFF; //
    int i=0, j=0;
    for(i=0; i<len; i++)
    {
        wrcrc^=*ptr++;
        for(j=0; j<8; j++)
        {
            if(wrcrc&0X0001)
            {
                wrcrc=wrcrc>>1^0XA001;
            }
            else
            {
                wrcrc>>=1;
            }
        }
    }
    return wrcrc<<8|wrcrc>>8; //little endian (LSB first)
}
```

## Appendix 2: CHECKSUM Accumulation and Verification

```
unsigned char CheckSum(unsigned char *buf, unsigned char len) //return CheckSum value
{
    uint8_t i, ret = 0;

    for(i=0; i<len; i++)
    {
        ret += *(buf++);
    }
    ret = ~ret;
    return ret;
}
```