

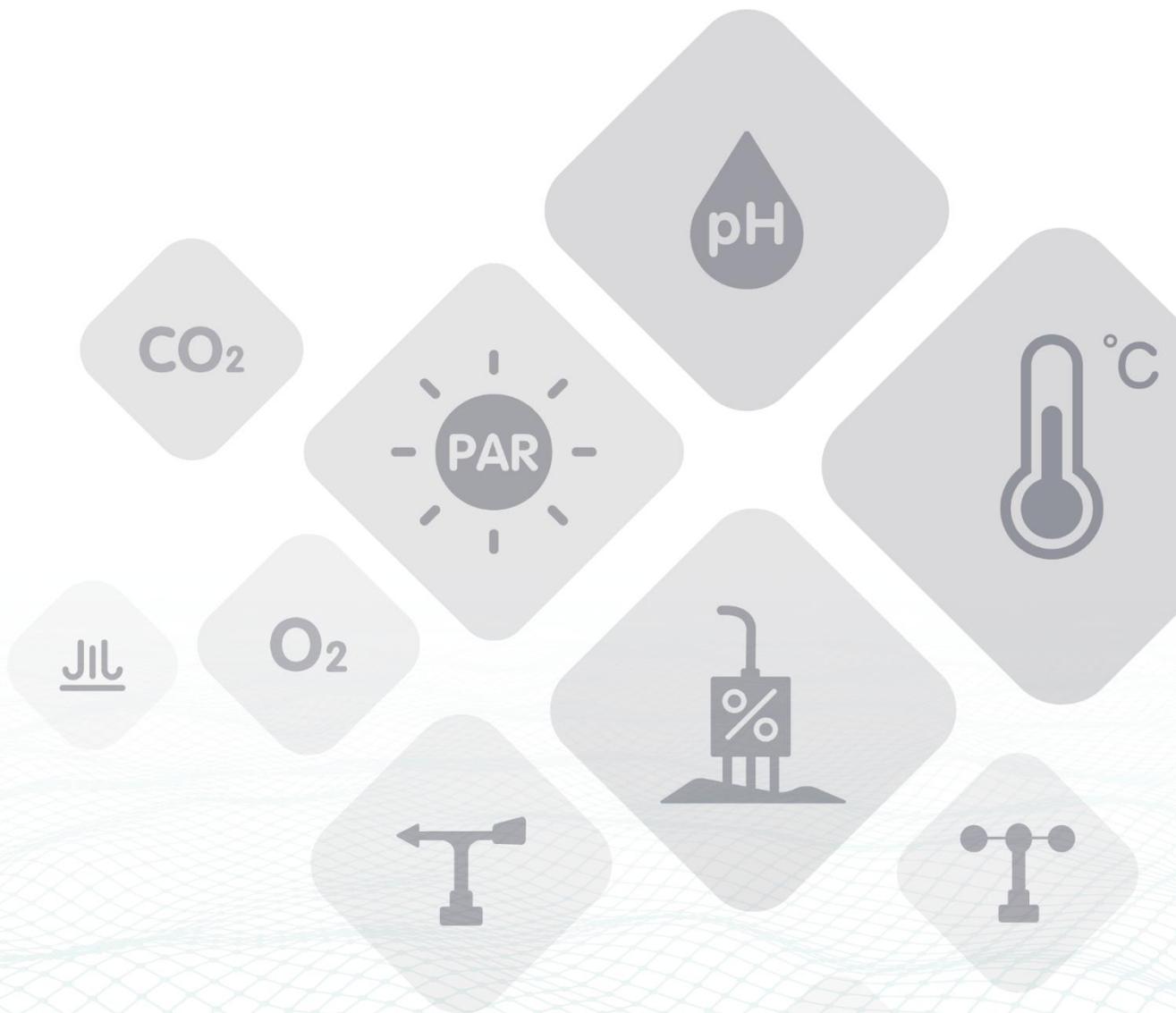


SENSECAP

# H<sub>2</sub>S Sensor Datasheet

Module: S-H2S-01

Version: V1.0



## Directory

1. Introduction .....	3
2. Technical parameters .....	6
3. RS-485 communication .....	8
3.1 Modbus-RTU RS-485 protocol summary .....	8
3.2 Modbus register .....	9
3.3 Description of Modbus register parameters .....	10
3.3.1 CRC validation algorithms and examples .....	10
3.4 Use Modbus to debug software communications .....	13
4. Precautions .....	15
5. Test the data .....	17

# 1. Introduction



## Key features:

- Sleep-type design, suitable for low-power IOT applications
- Combined with an intelligent algorithm, better environmental adaptability, more accurate detection, and stable zero point
- Good resistance to toxicity, no chemical material consumption, lifetime up to 5 years or more
- New micro-circuit design, strong anti-electromagnetic interference ability
- Fast response, fast return to zero, plug-and-play type
- Independent temperature and humidity digital sensor output
- Smallest size and lowest power consumption in the electrochemical field at present
- RoHS Compliant Design

## Scenarios:

Personal occupational safety and protection testing

Petrochemical field

Confined space operating environment detection applications

For odor and malodor gas detection

Sewage treatment plant application detection

Landfill and disposal environment H<sub>2</sub>S detection

Underground pipeline network working space safety protection real-time monitoring

Intelligent manhole cover

- Personal occupational safety and protection testing
- Petrochemical field
- Confined space operating environment detection applications
- For odor and malodor gas detection
- Sewage treatment plant application detection
- Landfill and disposal environment H2S detection
- Underground pipeline network working space safety protection real-time monitoring
- Intelligent manhole cover

**Physical and chemical hazards:**

- A slight odor begins at 0.1ppm
- 0.41 to 10ppm has an unpleasant odor
- Trachea stimulation and conjunctivitis above 10ppm
- Olfactory paralysis above 50ppm
- There is a risk of poisoning above 100ppm

The S-H2S-01 uses a sensor chip imported from Germany to measure the concentration of hydrogen sulfide, air temperature, and humidity. It outputs the data through the RS485 Modbus RTU protocol and has a quick response time, anti-interference technology, and can be calibrated for accuracy. The equipment is powered by 5V DC and can be easily integrated into larger systems without complicated installation.

Solid polymer electrochemical technology is an innovative method for detecting gases based on electrochemical catalytic reactions. This technology generates an electrical signal that is directly proportional to the gas concentration.

The sensor module comprises three catalytic electrodes, a solid electrolyte, and gas diffusion holes. The gas passes through the diffusion holes to reach the working electrode of the sensor, where an electrochemical redox reaction takes place on the porous micro-surface of the electrode. The solid electrolyte conducts electron transfer, and the resulting current signal characterizes the gas concentration.

## 2. Technical parameters

The basic parameters					
Product model	S-H2S-01				
Supply Voltage	3.3 ~ 5.5V DC (recommended 5V power supply).				
Supported Protocol	MODBUS-RTU RS485				
IP Rating	IP65 (The equipment should be protected from direct sunlight and rain over long periods of time).				
Operating Temperature	-40°C to 55°C (best operating temperature 20 to 35 °C).				
Operating Humidity	15 to 95% RH (non-condensation) (best operating humidity 50%RH).				
Ambient Pressure	Atmospheric pressure $\pm 10\%$ .				
Cable Length	2 meters				
Measurement Parameters					
	Measurement	Measurement accuracy	Resolution		
H <sub>2</sub> S	0~100 ppm	$\pm 5\%$ F.S	0.1 ppm		
Temperature	-40~85 °C	$\pm 0.2$ °C	0.1 °C		
Humidity	10~95 %RH	$\pm 2\%$	0.1 %RH		
Power					
Operating Current	<5mA				
Power Consumption	25mW @ 5V DC				
Performance reference					
	Parameter	Min	Typical	Max	Unit
RS-485 mode	Warm-up Time <sup>[1]</sup>	—	5 <sup>[2]</sup>	—	minutes
	Poll Rate <sup>[3]</sup>	—	1	—	seconds
	Response time <sup>[4]</sup>	—	30	—	seconds

[1] The time from when the sensor is powered on to when the data is read. Note the parameter when the sensor is powered on.

[2] If started in an environment with clean air it requires less time.

[3] The measurement data update interval, after the power-up warm-up time, if the power supply continues, the sensor periodically updates the reading at this interval.

[4]  $T_{90} < 30s$

Sensor wiring	
Yellow wire	RS485+/ A / T+
Green wire	RS485-/ B / T-
Red wire	VCC+
Black wire	VCC- (GND)

## 3. RS-485 communication

### 3.1 Modbus-RTU RS-485 protocol summary

The Modbus protocol is a common protocol for communication between electronic devices. It has become a common industry standard, and is widely used in data collectors, sensor equipment etc. Based on this protocol, devices produced by different vendors can communicate with each other which simplifies system integration with products from different manufacturers.

The Modbus protocol is a master-slave protocol. One node is the master, and the other nodes participating are slaves. Each slave device has a unique address. The sensor has RS485 interface and supports Modbus-RTU protocol. Sensing data and communication parameters can be acquired or modified by Modbus commands.

**Note:**

Default communication parameters: Baud rate 9600bps, one start bit,8 data bits, no verification, one stop bit.

## 3.2 Modbus register

Parameter	Register Address	Type	Function Code	Range and Description	Default
<b>Read-only registers</b>					
H <sub>2</sub> S	0x2000	Float32, read-only	3	Unit: ppm	N/A
Temperature	0x2004	int16, read-only	3	Units: °C, the actual value needs to be divided by 100	N/A
Humidity	0x2006	uint16, read-only	3	Unit: %RH, the actual value needs to be divided by 100	N/A
Max Range	0x2008	uint16, read-only	3	100ppm	100

- **Modify the address**

Default address: **0x10** (hex).

Use special instructions to modify the address, range 0x01 to 0XF7, out of range automatically reverts to 0x01

The instruction header	Modify the address
80 72 65 70 6F 6C 65 76 65 44	x

Example: The address needs to be changed to 0x02.

Send command: 80 72 65 70 6F 6C 65 76 65 44 02

**Note:**

When modifying the Modbus address, a single refresh is required (there is one device on the bus).

- **Error code**

According to the Modbus protocol, an error response is returned if an error occurs after a request for a feature code is initiated.

Address domain	Error function code	Error code	CRC check
1 byte	1 byte	1 byte	2 byte
	Request a function code +0x80	0x1: The function code is incorrect 0x2: Get length exceeded or register address error	

### 3.3 Description of Modbus register parameters

H <sub>2</sub> S Value		
Range	0 ~ 100 ppm	Register address: 0x2000
Read command: 10 03 20 00 00 02 CC 8A Return value: 10 03 04 <b>3C BB 08 2F</b> C1 5B		
The returned value is 3CBB282F (HEX), because the data type is 32-bit float, (byte[0]<<24) (byte[1]<<16) (byte[2]<<8) byte[3]), then 3CBB282F (HEX) = 0.02283105067908764 (DEC), the effective resolution is 0.1, then the H <sub>2</sub> S measurement value is 0 ppm		

Temperature value		
Range	-4000 to 8500 correspondence -40 to 85 °C	Register address: 0x2004
Read command: 10 03 20 04 00 01 CD 4A Returns value: 0F 03 02 <b>0A F9</b> 82 A5		
The returned value is 0AF9 (HEX). Since the data type is 16-bit signed integer, (byte[0]<<8 byte[1]), then 0AF9 (HEX)=2809 (DEC), the final value needs to be divided Take 100, that is, temperature value=2809/100=28.09 °C		

Humidity value		
Range	1000 to 9500 for 10 to 95% RH	Register address: 0x2006
Read command: 10 03 20 06 00 01 6C 8A Return value: 10 03 02 <b>18 BD</b> 8E 36		
The returned value is 18BD (HEX). Since the data type is 16-bit unsigned integer, (byte[0]<<8 byte[1]), then 18BD (HEX)=6333 (DEC), the final value needs to be divided Take 100, that is, the temperature value=6333/100=63.33%RH		

**Tip:** Command resolution

Read the ammonia value command **10 03 20 00 00 02 CC 8A**

**10**: Modbus address; **03**: Function code; **20 00**: Register address;

**00 02**: Number of registers; **CC E5**: Modbus CRC16 check;

Return value: **10 03 04 3C BB 08 2F 2F 9A**

**10**: Modbus address; **03**: Functioncode; **04**: Number of data bytes returned;

**3C BB 08 2F**: H<sub>2</sub>S value; **2F 9A**: Modbus CRC16 check;

#### 3.3.1 CRC validation algorithms and examples

```
static const unsigned char aucCRCHi[] = {
```

```
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,
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,
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,
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
```

```
};
```

```
static const unsigned char aucCRCLo[] = {
```

```
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
};

unsigned short usCRC16( unsigned char * pucFrame, unsigned short usLen )
{
    unsigned char    ucCRCHi = 0xFF;
    unsigned char    ucCRCLo = 0xFF;
    int              iIndex;

    while( usLen-- )
    {
        iIndex = ucCRCLo ^ *( pucFrame++ );
        ucCRCLo = ( UCHAR )( ucCRCHi ^ ucCRCHi[iIndex] );
        ucCRCHi = ucCRCLo[iIndex];
    }
    return ( unsigned short )( ucCRCHi << 8 | ucCRCLo );
}

```

The CRC generated by this function has exchanged the high and low bytes and can be directly placed in the message for transmission.

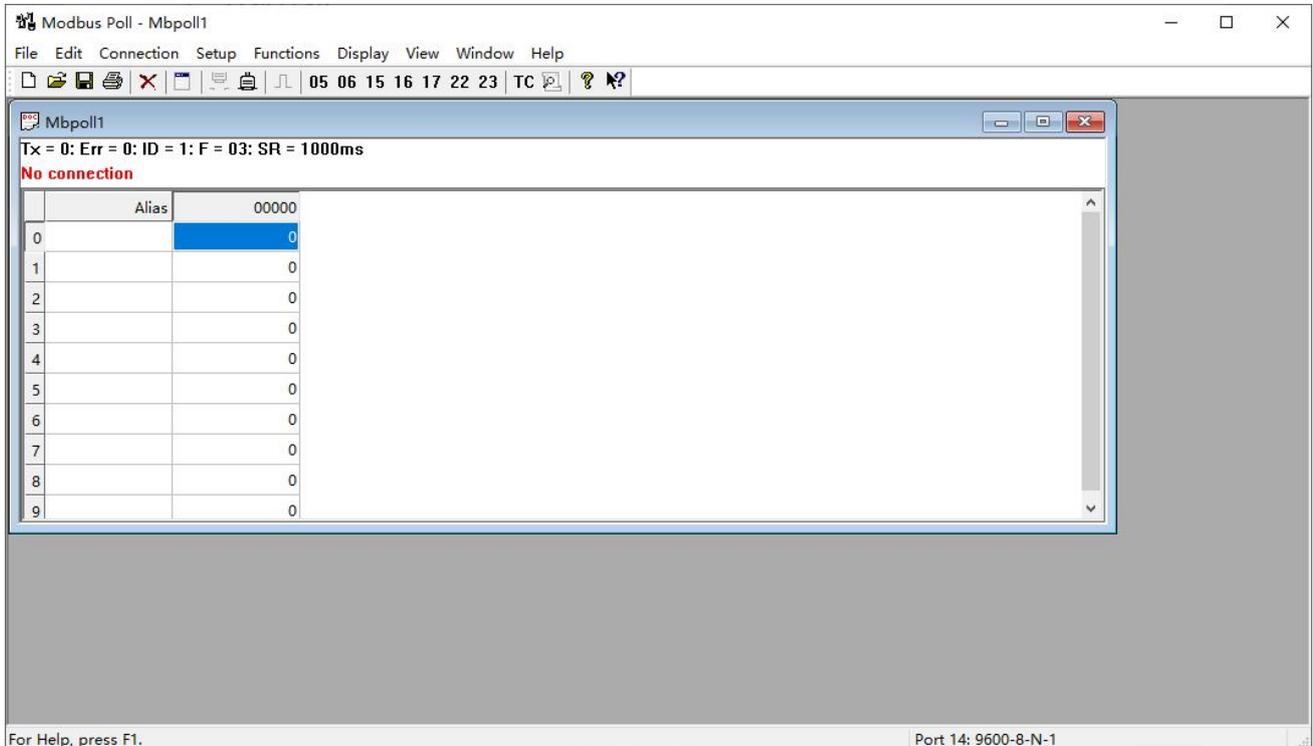
**Example:** The CRC16 of a frame is calculated by this function to be equal to 0x4112, then the message is placed as follows:

Address	Function Code	Data Count	Data	Data	Data	Data	CRC low	CRC high
							0x41	0x12

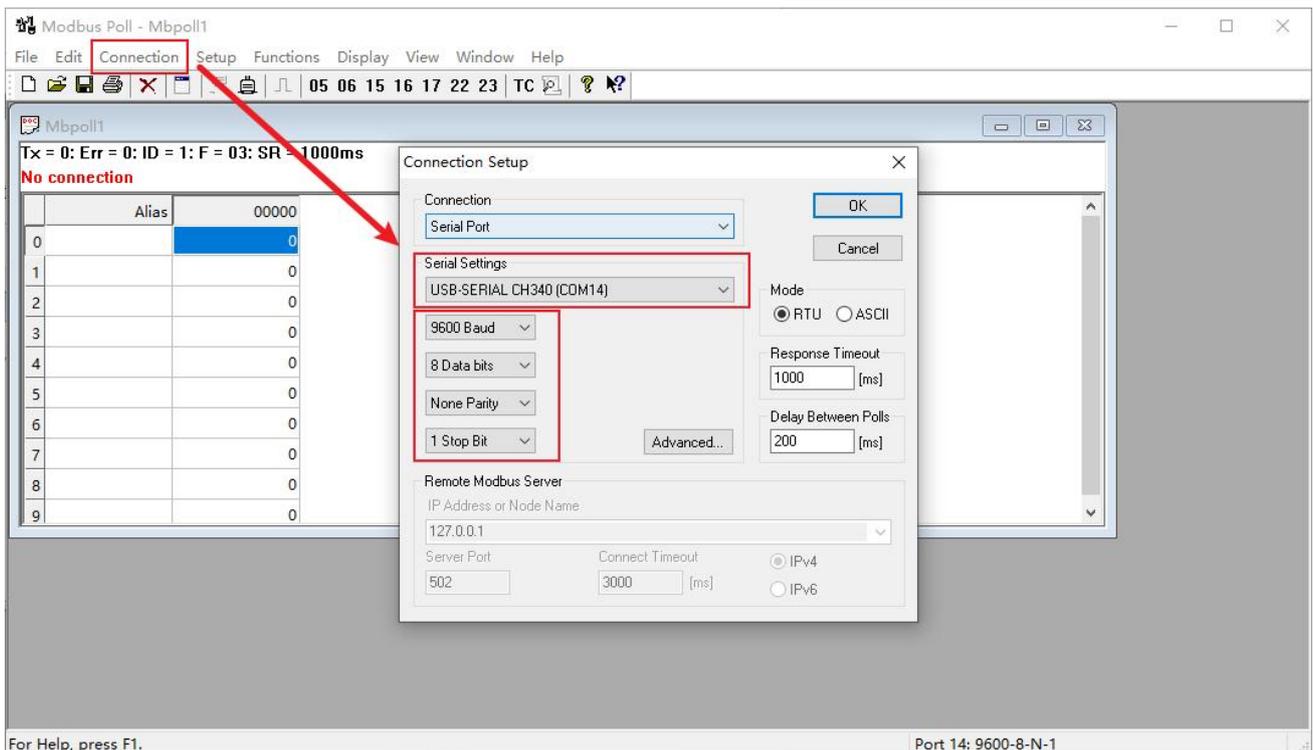
## 3.4 Use Modbus to debug software communications

Here's an example of the Modbus Poll tool

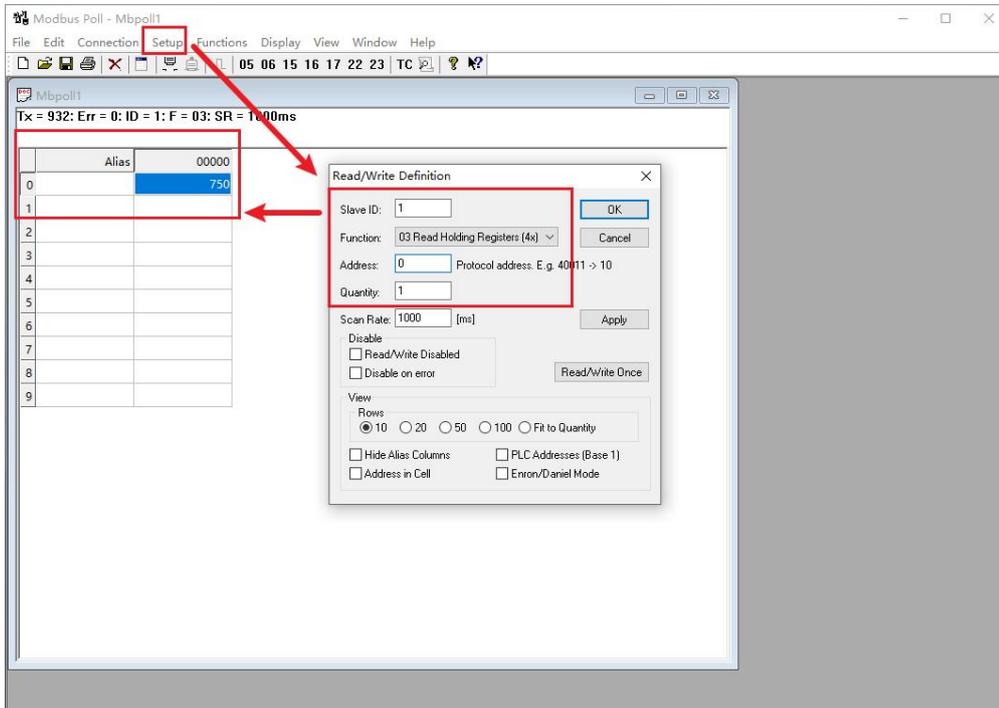
download address: <https://www.modbustools.com/download.html>.



Configure connection parameters: Baud rate 9600bps, one start bit, 8 data bits, no verification, one stop bit.



Configure the parameters of the read register 0x00: the default slave address is 1, the function code is 03, the starting address is 0, and the number is 1



## 4. Precautions

- (1) The main function of the gas sensor is to detect the gas composition and content, please do not let any part of the sensor come into contact with liquids.
- (2) The sensor wire sequence cannot be reversed. Once reversed, it will cause permanent damage to the sensor and make it unusable.
- (3) Different gas sensors have different concentration measurement ranges. Using the sensor in environments with concentrations above the maximum range for a long period of time can damage the device.
- (4) The white or yellow sheet on the sensor is a waterproof and breathable membrane, please be careful not to scratch or cut it.
- (5) The ventilation surface of the sensor should not be blocked or polluted. Sometimes the clogging of the hole can cause sensitivity reductions and slow response time.
- (6) To measure the gas sensitivity, avoid blowing the gas directly into the equipment, use standard gas caps. One side is the air intake, one side of the air outlet (under normal circumstances the small hole is the intake, and the large hole lets the gas out).
- (7) Avoid excessive shocks or vibration. If the shell is damaged, please ensure that the sensor is not damaged. If the shell is broken and the internal structure is exposed, the output will no longer be reliable (the signal may be too large, too small or no output).
- (8) It takes some time to return to the initial state after long-term exposure to high gas concentrations.
- (9) When using a low-range sensor avoid exposing the sensor to air with high concentrations over a long period of time. Long-term exposure to high concentrations of the gas can damage the sensor and decrease the performance.
- (10) Please do not disassemble the sensor, doing so might damage the sensor.
- (11) Please avoid instantaneous high voltages, which can seriously damage the sensor.
- (12) Calibration and detection, the electrochemical gas sensor's reaction activity with the target gas will change with time, temperature, and humidity changes. Because the sensor outputs the relative changes in current it is necessary to calibrate it to accurately calculate the absolute concentration.

First use a zero gas and a gas with a standard concentration. Calibrate the linear change of the gas to obtain a standard curve. When measuring, compare the electrical signal generated by the concentration of the gas to be measured with the electrical signal of the standard concentration to calculate the accurate gas concentration value. Therefore, zero-calibration and regular recalibration of the equipment are all necessary to ensure the accuracy of the measurement.

- (13) Cross interference of sensors. Each sensor corresponds to a specific detection gas, but any gas sensor cannot be absolutely specific. Therefore, when using a gas sensor, try to avoid interference from other gases with the sensor to ensure its accurate detection of specific gases.
- (14) Measuring range and accuracy: select a gas sensor that matches the range and accuracy for your application scenario, otherwise it may not be able to detect the gas and determine the accurate data or the sensor may get damaged.
- (15) Storage environment: temperature (-5 ~ +30°C), relative humidity 50%RH (non-condensing); the storage environment should have clean air, no pollution, no acetone, no high-concentration

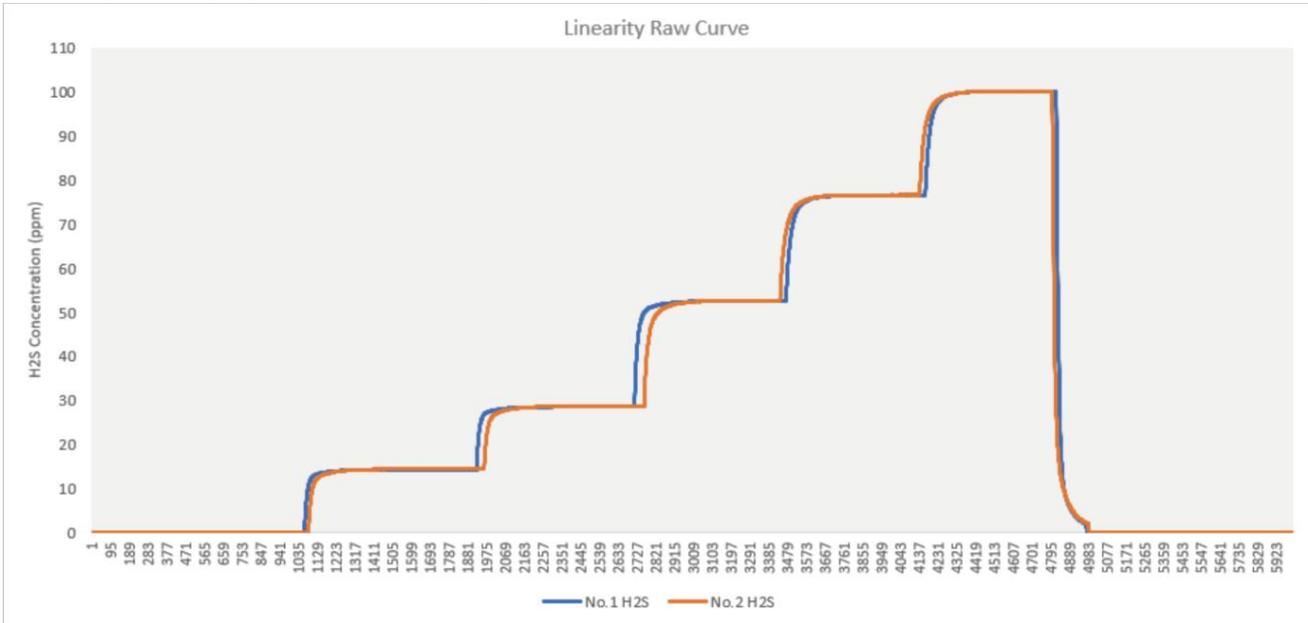
organic gas, no dust, no Smoke; avoid storing with alcohol (ethanol), perfume, sodium silicate and polyurethane liquid or solid; avoid storing in high temperature and low humidity environment.

- (16) Transportation: For long-distance transportation, the temperature in the sensor package should be kept within 40°C, and the maximum temperature should not exceed 55°C. The humidity should not be lower than 15% RH.
- (17) Maintenance: The maintenance of the sensor mainly involves the calibration. Generally, the solid polymer sensor does not consume chemical electrolyte, but due to the influence of temperature, humidity, dust, and other polluting gases, the sensitivity of the sensor will be affected. When an offset occurs, the hydrogen sulfide sensor needs to be recalibrated. The better the environment, the longer the sensor can go without requiring maintenance.

The user should follow the precautions mentioned above. If improper use caused damage to the sensor, it is not covered under the warranty, and we will not replace the device.

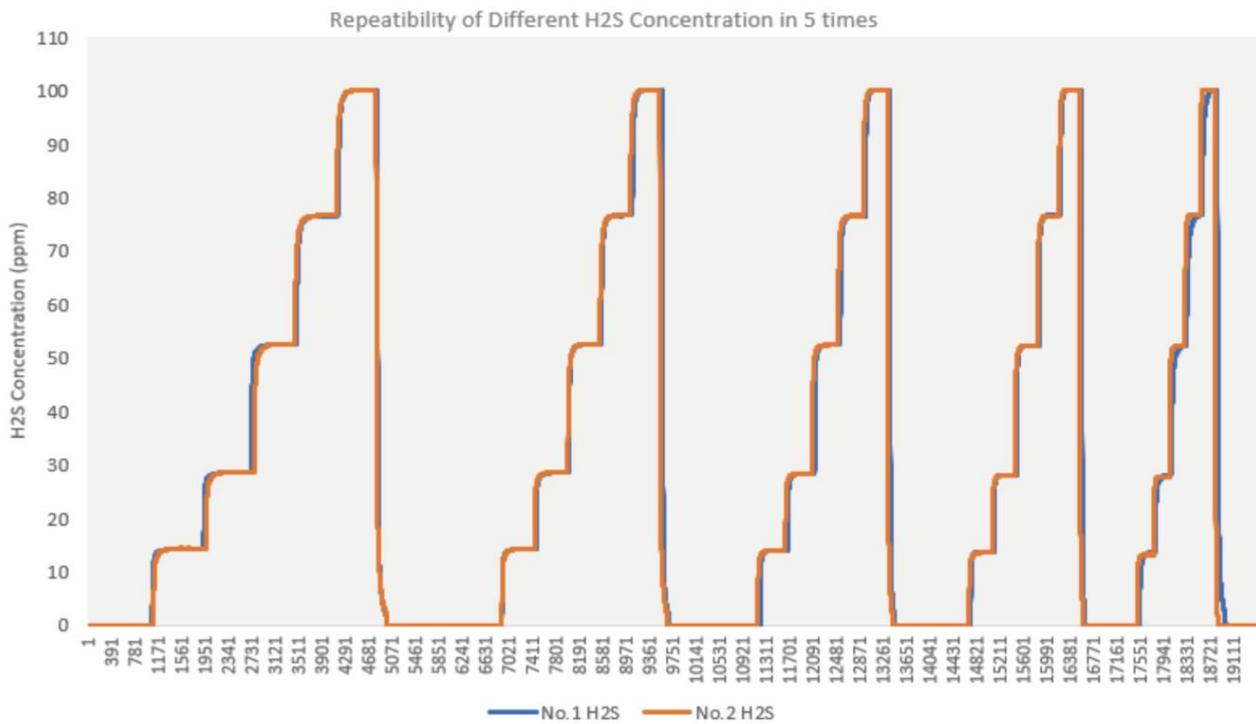
# 5. Test the data

## Linearity testing (typical data testing)



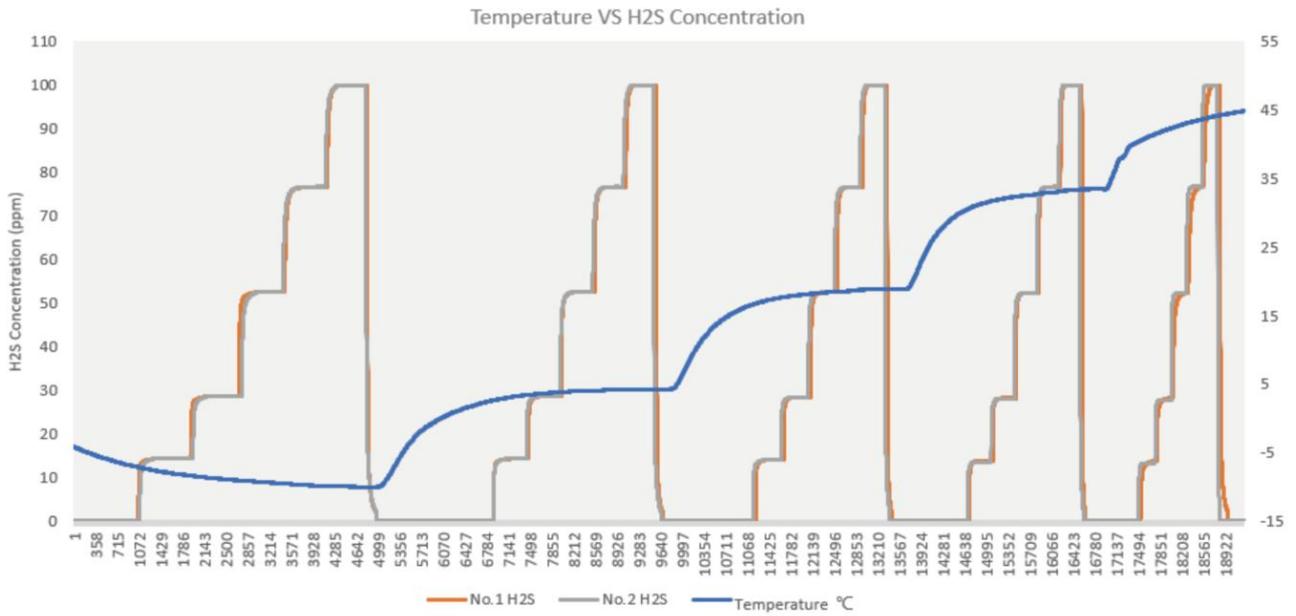
(Temperature:25 °C, Hmidity:55%, Gas chamber space: 0.03m<sup>2</sup>, Ventilation system with ventilation flow 4000sccm)

## Repetitive testing (typical data testing)



(Temperature:26 °C, Hmidity:55%, Gas chamber space: 0.03m<sup>2</sup>, Ventilation system with ventilation flow 3000sccm)

### High and low temperature tests (typical data testing).



(Temperature: -15, -5, 5, 15, 25, 35, 45, 55°C, Gas chamber space 0.03m<sup>2</sup>, ventilation system ventilation flow 3000sccm)