SZ521-A

Four-electrode

Conductivity (Salinity) Sensor

User Manual

SIBO.X INDUSTRIAL CO.,LTD.
Add: No. Building 1, No. 1, Jingshi Road, Cicheng Town Industrial Park, Jiangbei District, Ningbo City, Zhejiang, China
https://www.sbxsun.com
Email: info@sbxsun.com
Tel: +86-15958288207

Catalog

Preface

Dear customer

Thank you for using our product. Reading the entire manual before use is highly recommended for operation and maintenance the instrument and out of unnecessary trouble.

Please observe the operating procedures and precautions in this manual.

To make sure the effective after-sales protection provided by the instrument, please do not use any operation or maintenance other than which mentioned in the manual.

Due to non-compliance with the precautions specified in this manual, any fault and loss caused shall not be covered by the warranty, and the manufacturer shall not bear any relevant responsibility. If you have any questions, please contact our after-sales service department or representative.

Carefully unpack the instrument and accessories from the shipping container, and inspect for possible damage during shipping. Check received parts with items on the packing list. If any parts or materials are damaged or missing, please contact our customer service or the authorized distributor immediately.

Save all packing materials until you are sure that the instrument functions properly. Any damaged or defective items must be returned in their original packaging materials.

1 Overview

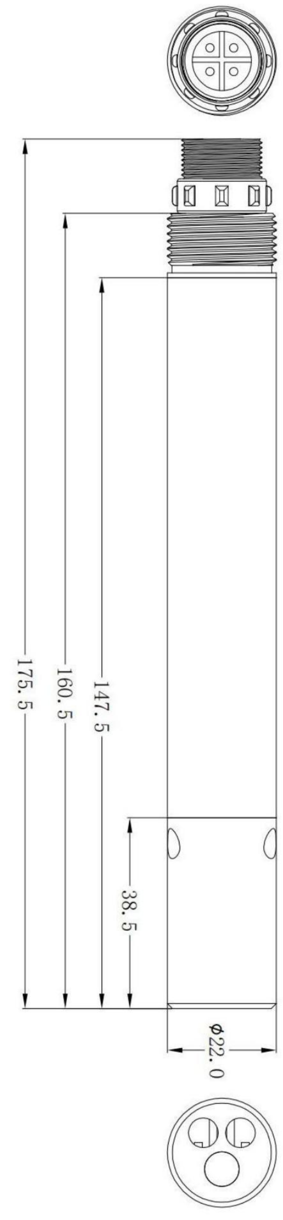
Our company's 4-electrode conductivity (salinity) sensor provides more accurate, stable measurement and wider measurement range than traditional two electrode design. It has no polarization effect and can deploy in long periods of time without frequent maintenance. It supports RS485, MODBUS protocols.

Features：

The RS-485 output , Modbus protocol compatible;
No polarization effect, low maintenance;
Automatic temperature compensation;
Wide range and excellent interference rejection.

1.1 Introduction

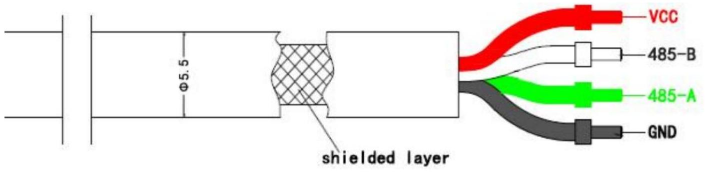
▲ **Conductivity（Salinity）Sensor**



▲ **Conductivity (Salinity) Sensor size**

1.2 Cable definition

4 wire AWG-24 or AWG-26 shielding wire. OD=5.5mm



1, Red—Power (VCC)

2, White—485 Date_B (485_B)

3, Green—485 Date_A (485_A)

4, Black—Ground (GND)

1.3 Technical Specifications

Name	Conductivity (Salinity) Sensor
Principle	4-Eclectrode
Conductivity Range	0-200mS/cm （0-2mS/cm、2-20mS/cm、20-200mS/cm）
Conductivity Accuracy	±1% or 0.01mS/cm （Max）
Salinity Range	0-175ppt
Salinity Accuracy	±1ppt
TDS Range	0-128000mg/L
Response Time	<10 sec
Housing IP Rating	IP68
Maximum Pressure	6bar
Temperature Range	0-50℃
Sensor Interface	Supports RS-485，MODBUS protocol
Power	0.2W （Suggested power supply： DC 9-24V, > 500mA）
Size	Φ22mm*175.5mm
Probe Cable Length	10m (default), customizable
Calibration	One-point or two-points calibration
Electrode Material	Ni +PEEK
Sheath Material	Titanium alloy

Note:

The above technical parameters are all data under laboratory standard liquid environment.

Sensor life and maintenance calibration frequency are related to actual field conditions.

2 Installation

2.1 Configuration table

Standard configuration	Number	Remarks
Conductivity (Salinity) Sensor	1	
Wires and Cables	1	10m

2.2 Sensor Installation

(1) Wiring and power supply

① Do not use the sensor cable to pull the sensor! It is required to install sensor in a secure and stable mounting bracket.

② The female and male connector of sensor cable should be screwed tightly to avoid moisture incursio.

③ Make sure power supply voltage is correct before power on.

(2) Sensor installation

① It is recommended to install the sensor vertically with electrodes facing down.

② In consideration of the fluctuation of water level, install the sensor below water level of 30cm, and try to install it in the position where there are no bubbles in the water;

③ Considering the basic principles of optics, Please keep the sensor end of the light window is not less than 10cm from the bottom of the container/related device!

3 Calibration

3.1 Brief description

Conductivity（salinity） sensor supports one-point or two-points calibration. The calibration tool can be used with our SmartPC software. Customers can also develop by themselves according to the communication protocol.

SmartPC User Manual:

（1） Open the SmartPC and select "Language" in the title bar: English.

（2）Select the correct port and click "Connect".

（3） Check calibration to perform calibration operation. Press F1 to get the help document. Find the calibration instructions for the CT/SAL probe.

（4）Conductivity（salinity）sensor user calibration is divided into three calibrations: 0~2mS/cm, 2~20mS/cm, 20~200mS/cm. The three-stage calibration corresponds to the following calibration standards: 1.4083mS/cm, 12.852mS/cm, and 111.31mS/cm. Depending on the application, select the appropriate standard solution for sensor calibration.

（5）You can also use the tool to measure and record data, referring to the help documentation.

3.2 Standard solution configuration

(1) Conductivity standard solution configuration:

Reagents required to prepare calibration solutions:

Potassium chloride: GR, dry at (220-240)℃ for 2h, and then put into a dryer to cool to room temperature;

Water: laboratory grade water or distilled or deionized water with a conductivity of not more than 0.2* 10^{-6} S/cm (at 25℃).

Solution code	Approximate concentration mol/L	Conductivity values S/cm				
		15℃	18℃	20℃	25℃	35℃
A	1	0.09212	0.09780	0.10170	0.11131	0.13110
B	0.1	0.010455	0.011163	0.011644	0.012852	0.015353
C	0.01	0.0011414	0.0012200	0.0012737	0.0014083	0.0016876
D	0.001	0.0001185	0.0001267	0.0001322	0.0001465	0.0001765

▲Table 1 Potassium chloride concentration corresponds to conductivity values

Solution number	Approximate molar concentration Mol/L	Potassium chloride required to prepare 1 L of solution g
A	1	74.2457
B	0.1	7.4365
C	0.01	0.7440
D	0.001	dilute 100 mL of C solution 10-fold

▲Table 2 Composition of solution

4 Maintenance schedule and methods

4.1Maintenance cycle

Unlike traditional 2-electrode conductivity sensors, 4-electrode conductivity (salinity) sensors require low maintenance.

Maintenance task	Frequency
	Conductivity（Salinity）Sensor
Sensor cleaning	Every 30 days
Sensor calibration	Every 3 to 4 weeks

4.2 Maintenance methods

（1） **Clean the sensor surface：** Wash the outer surface of sensor with tap water, if there is still residue, using soft brush for some stubborn dirt, household detergent can be added in tap water to clean.

（2） **Inlet and outlet hole of sensor：** Use a cotton swab or a soft cloth to wipe, for some stubborn dirt, can be added to the water in the washing liquid to wash;

（3） **Check the cable：** The cable should not be taut during normal operation， The wires inside the cable may break if it is under long term stressed condition.

（4） **Check whether the sensor shell is damaged due to corrosion or other reasons.**

4.3 Attention

Probe contains sensitive optical components and electronic components. Ensure that the probe far away from severe mechanical impact.

5 Trouble Shooting

Table 5-1 lists the symptoms, possible causes, and recommended solutions for common problems encountered with the CT/SAL sensor. If your symptom is not listed, or if none of the solutions solves your problem, please contact us.

ERROR	POSSIBLE CAUSE	SOLUTION
Communication abnormal	Power supplyor wiring issues	Check whether the power supply and wiring are correct according to the instruction
	Interface orprotocol issues	1. Use our SmartPC upper computer software to check whether the communication is normal. 2. Check according to the supporting communication protocol of the product.
Measured value is too high, Too Low or instability	Sensor's window isdirty and worn	Clean sensor body, special light window table
	Electrode Aging	Specimen validation evaluation
	Calibration is required	Perform user calibration
Other errors	Contact customer service	

Table 5-1 List of frequently asked questions

6 Quality Assurance

(1) The warranty period is 1 year.

(2) This quality assurance does not cover the following cases.

① Due to force majeure, natural disasters, social unrest, war (declared or undeclared), terrorism, the War or damage caused by any governmental compulsion.

②damage caused by misuse, negligence, accident or improper application and installation.

③ Freight charges for shipping the goods back to our company.

④ Freight charges for expedited or express shipping of parts or products covered by warranty.

⑤Travel to perform warranty repairs locally.

(3) This warranty includes the entire contents of the warranty provided by our company with respect to its products.

① This warranty constitutes a final, complete and exclusive statement of the terms of the warranty, and no person or The agent is authorized to establish other warranties in the name of our company.

② The remedies of repair, replacement, or return of payment as described above are exceptional cases that do not violate this warranty, and the remedies of replacement or return of payment are for our products themselves. Based on strict liability or other legal theory, our company shall not be liable for any other damage caused by a defective product or by negligent operation, including any subsequent damage that is causally related to these conditions.

7 Communication protocols

The RS485 communication protocol uses MODBUS communication protocol, and the sensors are used as slaves.

Data byte format.

Baud rate	**9600**
Starting position	1
Data bits	8
Stop bit	1
Check digit	N

Read and write data (standard MODBUS protocol)

The default address is 0x01, the address can be modified by register

7.1 Reading data

Host call (hexadecimal)

01 03 00 00 00 01 84 0A

Code	**Function Definition**	**Remarks**
01	Device Address	
03	Function Code	
00 00	Start Address	See register table for details
00 01	Number of registers	Length of registers (2 bytes for 1 register)
84 0A	CRC checksum, front low and back high	

Slave answer (hexadecimal)

01 03 02 00 xx xx xx xx

Code	**Function Definition**	**Remarks**
01	Device Address	
03	Function Code	
02	Number of bytes read	
XX XX	Data (front low and back high DCBA)	See register table for details
XX XX	CRC checksum, front low and back high	

7.2 Writing data

Host call (hexadecimal)

01 10 1B 00 00 01 02 01 00 0C C1

Code	Function Definition	Remarks
01	Device Address	
10	Function Code	
1B 00	Register Address	See register table for details
00 01	Number of registers	Number of read registers
02	Number of bytes	Number of read registers x2
01 00	Data (front low and back high DCBA)	
0C C1	CRC checksum, front low and back high	

Slave answer (hexadecimal)

01 10 1B 00 00 01 07 2D

Code	Function Definition	Remarks
01	Device Address	
10	Function Code	
1B 00	Register Address	See register table for details
00 01	Returns the number of registers written	
7D 2D	CRC checksum (front low and back high)	

7.3 Calculating CRC Checksum

(1) Preset one 16-bit register as hexadecimal FFFF (i.e., all 1s) and call this register the CRC register.

(2) Iso-oring the first 8-bit binary data (both the first byte of the communication information frame) with the lower 8 bits of the 16-bit CRC register and placing the result in the CRC register, leaving the upper 8 bits of data unchanged.

(3) Shift the contents of the CRC register one bit to the right (toward the low side) to fill the

highest bit with a 0, and check the shifted-out bit after the right shift.

(4) If the shifted out bit is 0: repeat step 3 (shift right one bit again); if the shifted out bit is 1, CRC register and polynomial A001 (1010 0000 0000 0001) for the iso-or.

(5) Repeat steps 3 and 4 until the right shift is made 8 times so that the entire 8-bit data is processed in its entirety.

(6) Repeat steps 2 through 5 for the next byte of the communication information frame.

(7) Exchange the high and low bytes of the 16-bit CRC register obtained after all bytes of this communication information frame have been calculated according to the above steps.

(8) The final CRC register content is obtained as follows: CRC code.

7.4 Register Table

Start address	Command Description	Number of registers	Data format (hexadecimal)
0x3000H	Device address (read and write)	1	2 bytes in total 00~01: Device address The range can be set from 1~254 For example, the data obtained is 02 00 (If the low position is in the front, it means that the address is 2) Take address 15 as an example, then 0F 00 Write the corresponding address (low in front) When the current device address is unknown, you can use FF as a common device address to ask for the current
0x0700H	Get Software and Hardware Rev	2	4 bytes in total 00 ~ 01: hardware version 02 ~ 03: software version For example, reading 0101 represents 1.1
0x0900H	Get SN	7	14 bytes in total 00: reserved 01 ~ 12: serial number 13: Reserved The 12 bytes of the serial number are translated according to ASCII code, i.e. the factory serial number

0x2600H	Temperature, conductivity and salinity values acquisition	8	16 bytes in total 00-03： Temperature value 04-07： CT(SAL) value 08-11： TDS 12-15： salinity value The reading temperature /CT(SAL) /TDS/salinity value is 4 bytes of data. (The low position is in the front, DCBA format, and this data needs to be converted to a change floating point number. The conversion method is shown below)
0x2000	Temperature calibration TK/TB (read/write)	4	Total 8 bytes 00~03: TK 04~07: TB To read TK for example, read out as 4 bytes of data (low bit in front, DCBA format, need to convert this data to floating point, see below for conversion method) To write Tk, for example, we need to convert Tk to 32-bit floating point and write it in (DCBA format) **Note: TK and TB should be read and written together**
0x1100H	User calibration K/B (read/write)	4	Total 24 bytes 00~03： K1 04~07： B1 08~11： K2 12~15： B2 16~19： K3 20~24： B3 0~2ms/cm corresponds to K1 and B1, default K1=1, B1=0 2~20ms/cm corresponds to K2 and B2, default K2=1, B2=0 20~200ms/cm corresponds to K3 and B3, default K3=1, B3=0 To read K1 for example, read out as 4 bytes of data (low bit in front, DCBA format, need to convert this data to floating point, see below for conversion method) To write k1, for example, we need to convert k1 to 32-bit floating point and write it in (DCBA format) **Note: The write process requires 24 bytes to be written simultaneously**

0x1200H	User calibration K1/B1 (read/write)	4	Total 8 bytes 00~03： K1 04~07： B1 User calibration corresponds to 0~2mS/cm To read K1 for example, read out as 4 bytes of data (low bit in front, DCBA format, need to convert this data to floating point, see below for conversion method) To write k1, for example, we need to convert k1 to 32-bit floating point and write it in (DCBA format) **Note: K1, B1 need to read and write together**
0x1300H	User calibration K2/B2 (read/write)	4	Total 8 bytes 00~03： K2 04~07： B2 User calibration corresponds to 2~20mS/cm To read K2 for example, read out as 4 bytes of data (low bit in front, DCBA format, need to convert this data to floating point, see below for conversion method) To write k2, for example, we need to convert k2 to 32-bit floating point and write it in (DCBA format) **Note: K2, B2 need to read and write together**
0x1400H	User calibration K3/B3 (read/write)	4	Total 8 bytes 00~03： K3 04~07： B3 User calibration corresponds to 20~200mS/cm To read K3 for example, read out as 4 bytes of data (low bit in front, DCBA format, need to convert this data to floating point, see below for conversion method) To write k3, for example, we need to convert k3 to 32-bit floating point and write it in (DCBA format) **Note: K3, B3 need to read and write together**

7.5 Conversion algorithms for floating point numbers

7.5.1 Converting floating point numbers to hexadecimal numbers

Step 1: Convert the floating point representation of 17.625 to binary floating point

First find the binary representation of the integer part

$17 = 16 + 1 = 1\times 2^4 + 0\times 2^3 + 0\times 2^2 + 0\times 2^1 + 1\times 2^0$

So the binary representation of the integer part 17 is 10001B

Then find the binary representation of the fractional part

0.625 $= 0.5 + 0.125 = 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$

So the binary representation of the decimal part 0.625 is 0.101B

So the floating point number in binary form for 17.625 expressed in floating point form is 10001.101B

Step 2: Shift to find the exponent.

Shift 10001.101B to the left until there is only one place left before the decimal point to get 1.0001101B, and

10001.101B = 1.0001101 B x 2^4 . So the exponential part is 4, which, when added to 127, becomes 131, whose binary representation is 10000011B

Step 3: Calculate the end number

Removing the 1 before the decimal point of 1.0001101B gives the trailing number 0001101B (because the 1 before the decimal point must be 1, the IEEE specifies that only the one after the decimal point should be recorded). An important note for 23-bit trailing numbers: the first bit (i.e. the hidden bit) is not compiled. The hidden bit is the bit to the left of the separator, which is usually set to 1 and suppressed.

Step 4: Symbol bit definition

A positive number has a sign digit of 0 and a negative number has a sign digit of 1, so 17.625 has a sign digit of 0.

Step 5: Convert to floating point

1 digit sign + 8 digits exponent + 23 digits mantissa

0 10000011 00011010000000000000000B (corresponding to 0x418D0000 in hexadecimal)

7.5.2 Converting hexadecimal numbers to floating point numbers

Step 1: Convert hexadecimal number 0x427B6666 to binary floating point number 0100 0010 0111 1011 0110 0110 0110 0110 0110B into sign, exponent and mantissa bits

0 10000100 11110110110110011001100110b

1 digit sign + 8 digits exponent + 23 digits mantissa

Sign bit S: 0 means positive

Index bit E: 10000100B = $1\times 2^7 + 0\times 2^6 + 0\times 2^5 + 0\times 2^4 + 0 \times 2^3 + 1\times 2^2 + 0\times 2^1 + 0\times 2^0$

=128+0+0+0+0+0+4+0+0=132

Last digit M: 11110110110011001100110B = 8087142

Step 2: Calculating floating point numbers

D = $(-1)^S \times (1.0 + M/2^{23}) \times 2^{E-127}$

= $(-1)^0 \times (1.0 + 8087142/2^{23}) \times 2^{132-127}$

= 1 x 1.964062452316284 x 32

= 62.85