KWS-300 4-Electrode Digital Conductivity Sensor

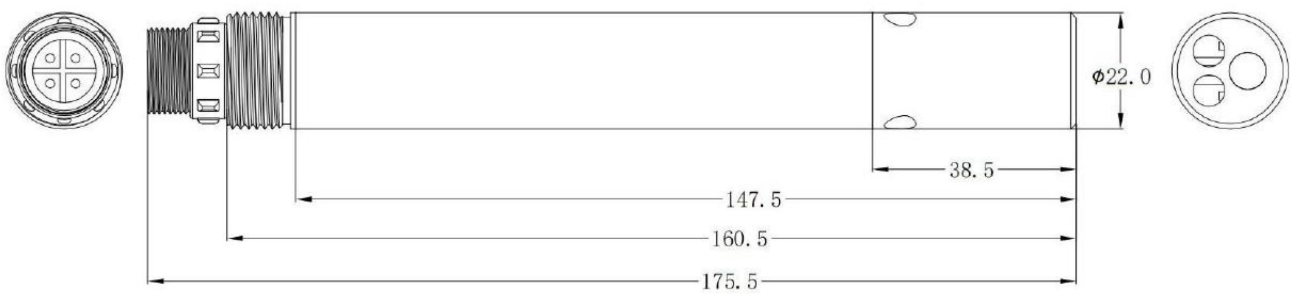
Introduction

Kacise introduces a new generation of four-electrode conductivity sensor, adopts international leading four-electrode technology, RS485 digital interface, supports MODBUS protocol, and environmentally friendly design. Compared with the traditional two-electrode conductivity sensor, it has higher accuracy, wider measurement range and excellent stability. The four-electrode conductivity sensor also has a unique advantage in quantity: one is to completely solve the polarization problem in high conductivity test; the other is to solve the problem of inaccurate reading caused by electrode contamination.

Technical Specifications

Item	**Conductivity Sensor**	**Salinity Sensor**
Model	KWS-300	KWS-310
Range	0.01-5mS/cm or 0.01-100 mS/cm	0-2.5ppt or 0-80ppt
Accuracy	< ±1% or 0.01mS/cm	±0.05ppt or ±1ppt
Response Time	< 10s	
Ip Grade	IP68	
Max. Pressure	6 bar	
Electrode Material	Nickel+PEEK	
Sheath Material	Titanium	
Operating Temperature	0~50℃(no freezing)	
Output	RS485, Modbus	
Power Supply	DC9 -24V, ≥1A	
Sensor Dimension	φ22mm*175.5mm	
Cable Length	10 meters (default), customizable	
Calibration	support one point or two-points calibration	

Dimensions



Tel: 0086-029-62929722 **Email:support@kacise.com**

Maintenance Schedule and Method

1. Maintenance Schedule

Unlike traditional 2-electrode conductivity sensors, 4-electrode conductivity sensors require low maintenance. (except when applied to viscous liquids).

Maintenance Task	Recommended Maintenance Frequency
Clean the sensor	Recommended every 3 - 4 weeks (shorter if the water body is dirty); The inside of the electrode cavity needs to be gently brushed clean with the matching brush.
Calibrate the sensor (if required by the competent authority)	Recommended every 3 - 4 weeks (depending on the working conditions)

Note:

The maintenance frequency in the above table is only a recommendation.

Please maintain the sensor according to the actual use of the sensor.

2. Maintenance Methods

Sensor maintenance

1) Sensor outer surface: Clean the outer surface of the sensor with tap water. If there is still debris left, wipe it with a soft cloth. For some stubborn dirt, add some household washing liquid to the tap water to clean it.

2) Sensor inlet and outlet holes: Wipe with a cotton swab or a soft cloth. For some stubborn dirt, add household washing liquid in tap water to clean

3) Check the cable of the sensor: the cable should not be tightened during normal operation, otherwise the internal wire of the cable may be broken, causing the sensor to not work properly;

4) Check the sensor's outer casing for damage due to corrosion or other causes.

Attention

The probe contains sensitive optical and electronic components. Make sure the probe is not subject to severe mechanical shock. There are no parts inside the probe that require user maintenance.

Wiring

Red——power supply (VCC)

White——485 data _B (485_B)

Green——485 data _A (485_A)

Black——GND

Calibration

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

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

Composition of solution

Modbus Protocol

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

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, low first, high last	

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 (low first, high last DCBA)	See Register Table For Details
XX XX	CRC Checksum, low first, high last	

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	Read The Number of Registers
02	Number of Bytes	Read The Number of Registers x2
01 00	Data (low first, high last DCBA)	
0C C1	CRC Checksum, low first, high last	

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 Written Registers	
7D 2D	CRC Checksum, low first, high last	

Calculating CRC Checksum

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

(2) XOR the first 8-bit binary data (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 eight bits unchanged;

(3) Shift the contents of the CRC register one bit to the right (toward the low bit) 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 one bit to the right again); if the shifted-out bit is 1, XOR the CRC register with the polynomial A001 (1010 0000 0000 0001);

(5) Repeat steps 3 and 4 until the right shift is made 8 times, so that the entire 8-bit data is processed;

(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 content of the CRC register is CRC code.

Register Table

Register Starting Address	Function Definition	Number of Registers	Data Format (hexadecimal)
0x3000H	Device Address (Read / Write)	1	2 bytes in total 00~01: Device address The settable range is 1~247 For example, the data obtained is 02 00 (low first, which means the address is 2) Taking address 15 as an example, write 0F 00 (low first) to the corresponding address. When the current device address is unknown, FF can be used as the general device address to query the current device address.
0x1100H	User Calibration K/B (Read / Write)	4	8 bytes in total 00~03: K 04~07: B To read K for example, read out is 4 bytes (low first, DCBA format, this data needs to be converted into a floating point number) To write K for example, K needs to be converted into a 32-bit floating point number first and written in according to (DCBA format). Note: K and B need to be read and written together.
0x2600H	Temperature, Conductivity and Salinity Value Acquisition	8	16 bytes in total 00~03: Temperature value 04~07: Conductivity value 08~11: TDS 12~15: Salinity value Continuously read temperature value/ conductivity value/ TDS value / salinity value, each is 4 bytes of data. (The low first, DCBA format, this data needs to be converted to floating point number)
0x2000	Temperature Calibration TK/TB (Read/Write)	4	8 bytes in total 00~03: TK 04~07: TB To read TK for example, readout is 4 bytes of data (the low first, DCBA format, this data needs to be converted to floating point number) To write TK for example, TK needs to be converted to 32-bit floating point first and written in according to (DCBA format) Note: TK and TB need to be read and written together.