DigiLL
Digital Liquid Level Sensor (RS485)
Digital Liquid Level Sensor (SDI-12)
User Manual

Index

1 Customer Support 3
2 Introduction 4
2.1 Specifications 5
2.2 Range, resolution and overload limit 6
2.3 Sensor Operating Mode 6
2.4 The decimal place of the liquid level value 7
3 Wiring diagrams 8
3.1 SDI-12 Interface 8
3.2 RS485 Interface 9
4 Dimension and Ordering Information 10
4.1 Dimension 10
4.2 Ordering Infomation 11
5 Installation & Maintenance 12
5.1 Installation considerations 12
5.2 Maintenance 12
6 SDI-12 Communication 13
6.1 Parameters and Symbols 13
6.2 Protocol 17
7 RS485 Communication 26
7.1 Modbus Protocol 26
7.2 Modbus Register 26
7.3 Modbus Register Detail Description 34
8 Common Operations 44
8.1 Liquid Level Unit and Value 44
8.2 Set Sensor Operating Mode 45
8.3 Zero-Trim of Liquid Level 46
8.4 User Calibration 46
Appendix A SDI-12 Sensor Testing and Settings 49
A.1 Testing SDI-12 Sensors with SDI12ELF20 Converter 49
A.2 Testing Example 50
Appendix B RS485 Sensor Testing and Settings 52
B.1 Testing RS485 Sensors with RS485 Converter 52
B.2 Testing Example 53
Copyright and Trademark 56
Version Control 56

1 Customer Support

Thank you very much for your order. Our success comes from the continuous faith in the excellence of our products and services, something we are committed to and would never sacrifice. Our customer service, especially in the after sales phase, guarantees the satisfaction of our clients. In line with this strategy, we appreciate that you can share with us your feedback at any time for our improvement, be it positive or negative, so if we can serve you better in anyway, please do inform us.

Website

http://www.infwin.com

E-Mail

infwin@163.com

Telephone

+86-411-66831953, +86-4000-511-521

2 Introduction

The DigiLL digital liquid level sensor integrates a high-precision liquid level measurement unit. With a compact size, it is widely used in long-term continuous monitoring of sea level measurement, irrigation equipment, deep well and groundwater monitoring, surface water, etc.

The SDI-12 output provides universal compatibility with any SDI-12-enabled data logger and low power applications, or use a RS485 physical interface for applications that require long cable runs or many sensors.

Features:

- Integrated high-precision liquid level measurement unit, with multiple range available
- Fully sealed waterproof design, waterproof rating of IP68
- Small in size, low power consumption
- High precision, fast response, good interchangeability and reliable performance
- Submersible installation or installation in pipelines and tanks
- RS485 or SDI-12 communication interface with surge protection
- Low power consumption
- Reverse power protection and Built-in TVS/ESD protection
- ODM/OEM

Applications:

- Irrigation Equipment
- Sea Water Level Measurement and Control
- Deep Well and Groundwater Monitoring
- Vessel and Storage Monitor Systems
- Control of Lift and Pumping Stations
- Surface Water Monitoring
- Dewatering

2.1 Specifications

Specifications	
Output Interface	Optional: SDI-12, V1.3 Optional: RS485, Modbus-RTU
Power Supply	4.5-28V DC
Power Consumption	**SDI-12 Interface:** User request measurement mode: Standby current <30uA; Measurement current <5mA; **RS485 Interface:** Continuous measurement mode: Average current <5mA; Periodic automatic measurement mode: Standby current <1mA within the cycle, measurement current <5mA;
Level Measurement	Measurement principle: Piezoelectric sensor Medium: Water or liquid compatible with the sensor material Range: 1/2/3/5/10/20/30/50/100/200 mH2O (Vented Gauge), or customized Accuracy: 0.25% F.S.@ 25℃ Temperature compensation: 0-50℃ Overload capacity: 200% F.S.
Temperature	Range: -40~80°C, Resolution: 0.03°C, Accuracy: +/-0.5°C
IP Ratings	IP68
Operating	Temperature: 0-80℃ (non-frozen)
Storage	Temperature: -5~80°C, Humidity: 0~90%
Contact surface material	304/316 stainless steels, polyurethane cable
Dimension	162*28mm (Length*Diameter)

2.2 Range, resolution and overload limit

Range	Accuracy@25℃	Resolution	Overload	Burst	Comment
1 mH2O(vented)	0.25% F.S.	1 mm	200% F.S.	500% F.S.	N/A
2 mH2O(vented)	0.25% F.S.	1 mm	200% F.S.	500% F.S.	N/A
3 mH2O(vented)	0.25% F.S.	1 mm	200% F.S.	500% F.S.	N/A
5 mH2O(vented)	0.25% F.S.	1 mm	200% F.S.	500% F.S.	N/A
10 mH2O(vented)	0.25% F.S.	1 mm	200% F.S.	500% F.S.	N/A
20 mH2O(vented)	0.25% F.S.	5 mm	200% F.S.	500% F.S.	N/A
30 mH2O(vented)	0.25% F.S.	5 mm	200% F.S.	500% F.S.	N/A
50 mH2O(vented)	0.25% F.S.	5 mm	200% F.S.	500% F.S.	N/A
100 mH2O(vented)	0.25% F.S.	10 mm	200% F.S.	500% F.S.	N/A
200 mH2O(vented)	0.25% F.S.	10 mm	200% F.S.	500% F.S.	N/A
Customized	TBD	TBD	TBD	TBD	TBD

2.3 Sensor Operating Mode

The sensor supports the following operating modes, and the features of each mode are as follows:

Interface	Operating Mode	Entry condition	Features
RS485	Continuous measurement mode (Factory default)	(1) Set Modbus register "Acquisition Interval <AcqInterval>" to 0. (2) Power cycle the sensor for the changes to take effect.	The sensor is in continuous measurement mode; it operates constantly with real-time data updates, leading to high power consumption.
	Periodic measurement mode	(1) Write desired value to modbus register "Acquisition Interval <AcqInterval>"; (2) Power cycle the sensor for the changes to take effect.	The sensor operates in a periodic measurement mode, conducting measurements and updating data during each cycle, and then entering a low-power state to wait for the next measurement cycle to arrive. The host can send data reading instructions at any time, and the sensor can immediately return the measurement data from the previous cycle.
SDI-12	User request measurement mode (Factory default)	(1) Automatically enters this mode upon power-on.	The sensor is in the user requested measurement mode. After receiving the measurement instruction, it conducts the measurement and returns the data, then enters a low-power state.

2.4 The decimal place of the liquid level value

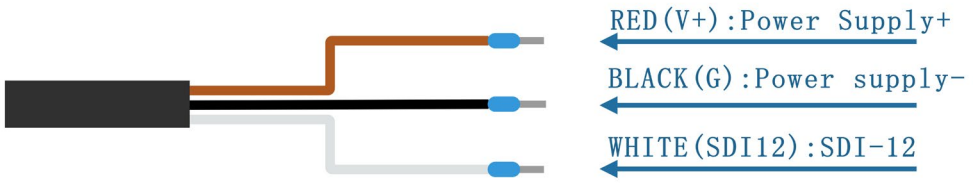
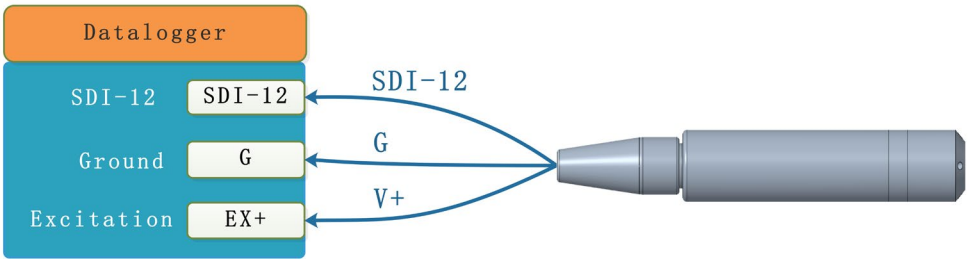
Interface	Liquid Level Register	Comments
SDI-12	<±LiquidLevel>	For floating-point registers, no conversion is required; the readout value is the actual liquid level.
RS485	<±LiquidLevel_FLOAT>	For floating-point registers, no conversion is required; the readout value is the actual liquid level.
	<±LiquidLevel_INT32>	32-bit signed integer, converted to the actual liquid level value according to the number of decimal places specified in the table below.

The number of decimal places for the liquid level value read by the user is determined by the liquid level unit set by the user, as shown in the table below:

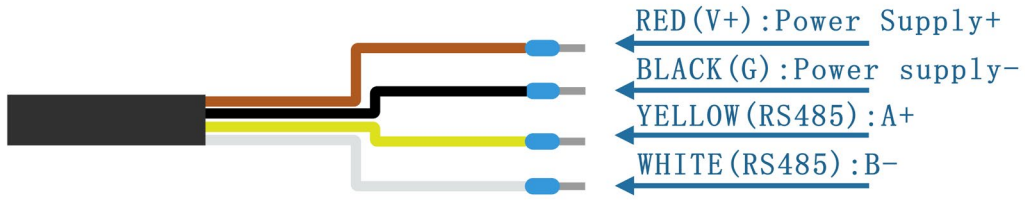
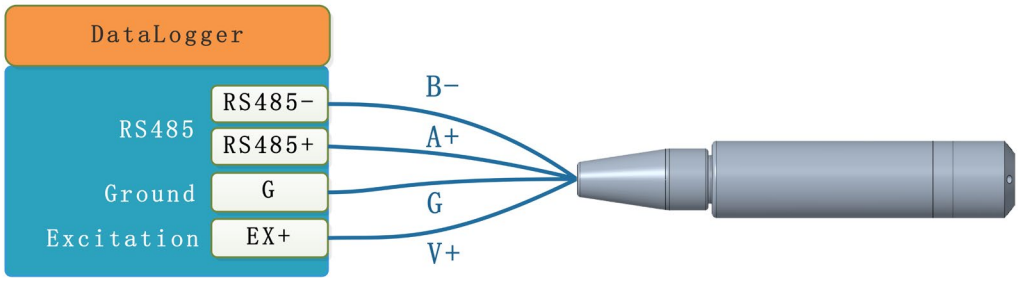
Liquid Level Unit <LiquidLevelUnit>	Decimal places <LiquidLevelDecimalPointNum>	Comments
0: Meter (m)	3	N/A
1: Centimeter (cm)	1	N/A
2: Millimeter (mm)	0	N/A
3: Inch (inch)	3	N/A
4: Feet (feet)	4	N/A
5: Pa (Pa)	0	N/A
6: hPa (hPa)	1	N/A
7: Kpa (KPa)	2	N/A
8: MPa (MPa)	5	N/A
9: bar (bar)	4	N/A
10: mbar (mbar)	1	N/A
11: psi (psi)	4	N/A

3 Wiring diagrams

3.1 SDI-12 Interface

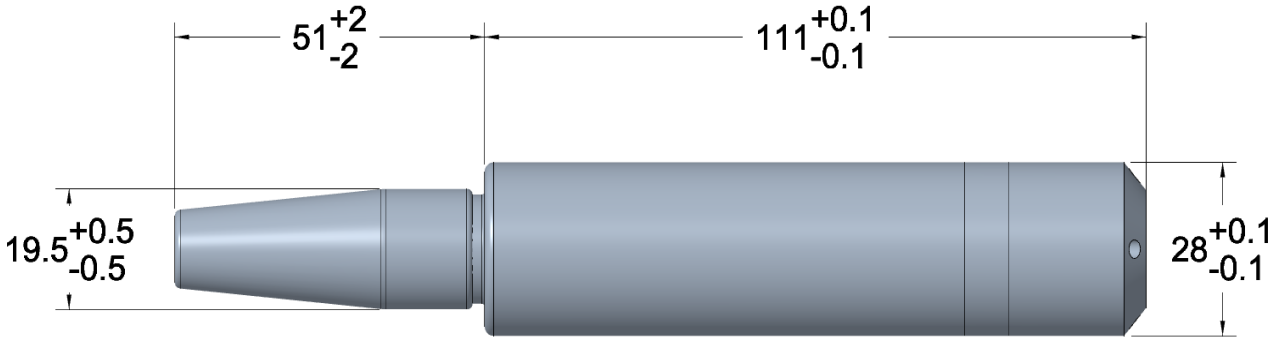
Type	Wiring diagram
SDI-12 Interface	Cold pressed terminal  Tinned lead wires  Wiring Diagram 

3.2 RS485 Interface

Type	Wiring diagram
RS485 Interface	Cold pressed terminal  Tinned lead wires  Wiring Diagram 

4 Dimension and Ordering Information

4.1 Dimension



Unit: mm

4.2 Ordering Infomation

Ordering Information		
Parameters	**Code**	**Comments**
Code 1: Product Series	DigiLL	DigiLL liquid level sensor
Code 2: Level Range	A B C D E F G H I J X	1 mH2O (Vented Gauge) 2 mH2O (Vented Gauge) 3 mH2O (Vented Gauge) 5 mH2O (Vented Gauge) 10 mH2O (Vented Gauge) 20 mH2O (Vented Gauge) 30 mH2O (Vented Gauge) 50 mH2O (Vented Gauge) 100 mH2O (Vented Gauge) 200 mH2O (Vented Gauge) Customized
Code 3: Power Supply	C	4.5-28V DC
Code 4: Output Interface	C F	RS485, Modbus-RTU SDI-12
Code 5: Housing Material	A X	304 Stainless steels Customized
Code 6: Cable Material	A X	PUR (Vented) Customized
Code 7: Cable length	XXX	xxx meters
Ordering Code Example: DigiLL - B C C A A 005 Product Series: DigiLL liquid level sensor Level Range: 2 mH2O (Vented Gauge) Power Supply: 4.5-28V DC Output Interface: RS485, Modbus-RTU Housing Material: 304 Stainless steels Cable Material: PUR (Vented) Additional cable length: 5 meters		

5 Installation & Maintenance

5.1 Installation considerations

The sensor is designed for water level measurements. Typical applications include agricultural water level/flow, water wells, lakes, streams, and tanks. If the device is to be installed in a liquid other than water or in contaminated water, check the compatibility of the wetted material. Please check with us if the sensor will be used in harsh water applications, including salt water.

■ **Vent and desiccant tubes**

A vent tube incorporated in the cable vents the sensor diaphragm to the atmosphere. This eliminates the need to compensate for changes in barometric pressure. To prevent water vapor from entering the inner cavity of the sensor, the vent tube opening terminates inside a desiccant tube.

Before installing the sensor, check the desiccant and replace it if the desiccant is failure. The desiccant tube must always be attached to the sensor, and the desiccant tube must be housed in an environmental enclosure.

■ **Appropriate water depth**

The sensor must be installed below the water at a fixed depth. This depth should be chosen so that the water pressure will never exceed the transducer's pressure range.

■ **Dislodge bubbles**

While submersing the sensor, air bubbles may become trapped inside the nose cone, causing small, offset errors until the bubbles dissolve. Dislodge these bubbles by gently shaking the sensor while under water.

If the bubbles are not removed by rotation and shaking underwater (or bleeding out the air in a closed system), the sensor reading will drift lower by the distance of the gap as the air is slowly dissolved into the water over time.

5.2 Maintenance

■ Clean Regularly.

■ When cleaning, first open the screw cap of the input transmitter, carefully take out the level sensor, and then remove the impurities attached to it.

■ When cleaning the pressure sensor hole, you can use alcohol to inject the pressure transmitter hole to its height 1 /2 soaked for about 5 minutes, and then slightly shake and repeat multiple times until it is cleaned. Do not use any tools to clean the pressure-inducing hole to avoid damaging the sensitive core pressure-sensitive diaphragm.

■ When cleaning the pressure interface and pressure hole of the transmitter, do not use excessively hard brushes or metal tools to avoid damaging the sensitive core and pressure interface threads.

■ After cleaning, install the screw cap back to the original place.

6 SDI-12 Communication

6.1 Parameters and Symbols

The sensor has SDI-12 interface and protocol. The description and terms used within this chapter are listed in table below:

Parameters	Unit	Description
±	-	Sign of the value
a	-	SDI-12 address
n	-	Number of measurements (fixed width of 1)
nn	-	Number of measurements with leading zero if necessary (fixed width of 2)
ttt	Seconds	Maximum measurement time (fixed width of 3)
tttt	Seconds	Maximum measurement time (fixed width of 4)
<TAB>	-	Tab character
<SAPCE>	-	Space character
<CR>	-	Carriage return character
<LF>	-	Line feed character
<Checksum>	-	SUM Checksum
<CRC>	-	SDI-12 protocol CRC Checksum
<VERIFY_STATUS>	-	Sensor Verification status
<TemperatureUnit>	-	Temperature Unit, Range: C: ℃ (Default) F: ℉
<±TOffset>	℃ ℉	Temperature Offset. The value unit is determined by the <TemperatureUnit> parameter. This value will be applied to the parameter "<±Temperature>" for temperature correction. Range: -10.00~+10.00 Default: 0.00
<±Temperature>	℃ ℉	Temperature (offset corrected). The value unit is determined by the <TemperatureUnit> parameter.
<LiquidLevelType>	-	Indicates the pressure type of liquid level sensor. This value is determined at the time of ordering. 0: Gauge pressure (default) 1: Sealed pressure 2: Absolute pressure
<LiquidLevelRange>	-	Indicates the range of the liquid level sensor. This value is determined at the time of ordering. 0: 1 mH2O (Vented Gauge) 1: 2 mH2O (Vented Gauge) 2: 3 mH2O (Vented Gauge) 3: 5 mH2O (Vented Gauge)

		4: 10 mH2O (Vented Gauge) 5: 20 mH2O (Vented Gauge) 6: 30 mH2O (Vented Gauge) 7: 50 mH2O (Vented Gauge) 8: 100 mH2O (Vented Gauge) 9: 200 mH2O (Vented Gauge)
<LiquidLevelUnit>	-	Liquid Level Unit. 0: Meter(m)(default) 1: Centimeter(cm) 2: Millimeter(mm) 3: Inch(inch) 4: Feet(feet) 5: Pa(Pa) 6: hPa(hPa) 7: KPa(KPa) 8: MPa(MPa) 9: bar(bar) 10: mbar(mbar) 11: psi(psi)
<LiquidLevelDecimalPointNum>	-	Used to indicate the decimal place of the liquid level value. The decimal places of the liquid level value is determined based on the "Liquid Level Unit <LiquidLevelUnit>" set by the user. Refer to the "Decimal Places of Liquid Level Value" section. 0: No decimal 1: 1 decimal place 2: 2 decimal places 3: 3 decimal places 4: 4 decimal places 5: 5 decimal places 6: 6 decimal places
<±LiquidLevel>	-	Liquid level value. The unit of the liquid level value is determined by "Liquid Level Unit <LiquidLevelUnit>". The number of decimal places for the liquid level value is determined by the "Liquid Level Unit <LiquidLevelUnit>" setting set by the user. Refer to the "Decimal Places for Liquid Level Value" section.
<LocalGravityAcceleration>	m/S2	Local Gravity Acceleration. Range: 9.50000~10.00000 Default: 9.80665
<LiquidDensity>	g/cm3	Liquid Density. Range: 0.50000~2.00000 Default: 1.00000
<ZeroPointSuppressRange>	-	Zero-point suppression range.

		When the value of <±LiquidLevel> is within the following range, the sensor will automatically set the value of <±LiquidLevel> to 0. 0: Disabled 1: 0.1% FS (full scale) 2: 0.2% FS (full scale) 3: 0.5% FS (full scale) (default) 4: 1.0% FS (full scale) 5: 2.0% FS (full scale) 6: 5.0% FS (full scale)
<Filter>	-	Pressure/level value filtering strength. Higher values result in greater stability of the data, but the measurement speed decreases. Range: 0~16 (0 is disabled) Default: 16
<LiquidLevelUnitForCalib>	-	Pressure/Level Calibration Unit. The pressure/level unit used for entering calibration point values during user calibration. 0: Meter(m)(default) 1: Centimeter(cm) 2: Millimeter(mm) 3: Inch(inch) 4: Feet(feet) 5: Pa(Pa) 6: hPa(hPa) 7: KPa(KPa) 8: MPa(MPa) 9: bar(bar) 10: mbar(mbar) 11: psi(psi)
<LiquidLevelZeroTrimValue>	-	Pressure/level zero trim value (always 0).
<LiquidLevelZeroTrimInternal Value>	-	Internal value for the pressure/level zero trim.
<LiquidLevelZeroTrimResult>	-	Pressure/level zero trim operation result. 0: Operation successful 1: Internal error; 2: Exceeding the zero trimming range (10% F.S.); 3: Sensor error;
<LiquidLevelCalLowerValue>	-	Pressure/Level lower calibration point value. The unit for calibration point data entered during calibration is determined by the parameter "<LiquidLevelUnitForCalib>".
<LiquidLevelCalLowerInternal Value>	-	Internal value for the pressure/level lower calibration point.
<LiquidLevelCalLowerResult>	-	Pressure/Level lower calibration operation result.

		0: Operation successful; 1: Incorrect input of the low calibration point value (exceeding the range or greater than the high calibration point value); 2: Sensor error;
<LiquidLevelCalUpperValue>	-	Pressure/Level upper calibration point value. The unit for calibration point data entered during calibration is determined by the parameter "<LiquidLevelUnitForCalib>".
<LiquidLevelCalUpperInternal Value>	-	Internal value for the pressure/level upper calibration point.
<LiquidLevelCalUpperResult>	-	Pressure/Level upper calibration operation result. 0: Operation successful; 1: Incorrect input of the low calibration point value (exceeding the range or greater than the high calibration point value); 2: Sensor error;

The measurement value will be set as following when there is sensor error:

Error Value	Modbus Register	Comments
+9999999	<±LiquidLevel > <±Temperature >	sensor error

6.2 Protocol

Request	Response	Comment
a!	a<CR><LF>	**Acknowledge Active** a: Sensor address **Example:** Request: 0! Response: 0<CR><LF>
aI!	allccccccccmmmmmmvvvxxxxxxxxxxxxxxxx<CR><LF>	**Send Identification** a: Sensor address ll: SDI-12 Version Number cccccccc: 8 characters' vendor identification mmmmmm: 6 characters specifying the sensor model number vvv: 3 characters specifying the sensor version xxxxxxxxxxxxx: 13 characters' serial number <CR><LF>: terminates the response **Example:** Request: 0I! Response: 013INFWIN DigiLL3.02506080001000<CR><LF>
?!	a<CR><LF>	**Sensor Address Query** a: Sensor address **Example:** Request: ?! Response: 0<CR><LF>
aAb!	b<CR><LF>	**Change Sensor address** a: Current Sensor address b: New Sensor address **Example:** Request: 0A1! Response: 1<CR><LF>
aM!, aMC!	attt2<CR><LF> a: Sensor address ttt: Measurement data will be ready in ttt seconds, the specific value is affected by the parameter "<Filter>". 2: Number of measurement data returned by aD0! <CR><LF>: terminates the response	Measure the liquid level value and the temperature. Note: The unit of the liquid level value is determined by "Liquid Level Unit <LiquidLevelUnit>". Note: The number of decimal places for the liquid level value is determined by the "Liquid Level Unit <LiquidLevelUnit>" setting set by the user. Refer to the "Decimal Places for Liquid Level Value" section.

	aD0! Response data format: a<±LiquidLevel><±Temperature>[<CRC>]<CR><LF>	**Example:** Request: 0M! Response: 00012<CR><LF> Response: 0<CR><LF> Request: 0D0! Response: 0+0.000+26.16<CR><LF>
aM1!, aMC1!	attt6<CR><LF> a: Sensor address ttt: Measurement data will be ready in ttt seconds, , the specific value is affected by the parameter "<Filter>". 6: Number of measurement data returned by aD0! <CR><LF>: terminates the response aD0! Response data format: a<LiquidLevelType><LiquidLevelRange><LiquidLevelUnit><LiquidLevelDecimalPointNum><±LiquidLevel><±Temperature>[<CRC>]<CR><LF>	Liquid Level Type, Range, Unit, Decimal Point Number, Liquid Level, Temperature (offset corrected) Measurement Note: The unit of the liquid level value is determined by “Liquid Level Unit <LiquidLevelUnit>“. Note: The number of decimal places for the liquid level value is determined by the "Liquid Level Unit <LiquidLevelUnit>" setting set by the user. Refer to the "Decimal Places for Liquid Level Value" section. **Example:** Request: 0M1! Response: 00016<CR><LF> Response: 0<CR><LF> Request: 0D0! Response: 0+0+2+0+3+0.000+26.16<CR><LF>
aC!,aCC! aC1!,aCC1! aC2!,aCC2! aC3!,aCC3! aC4!,aCC4! aC5!,aCC5! aC6!,aCC6! aC7!,aCC7! aC8!,aCC8! aC9!,aCC9!	This command set has the same data format as aM!, aMC!, aMx!, aMCx! (x=1-9), refer to the corresponding command.	This command set has the same data format as aM!, aMC!, aMx!, aMCx! (x=1-9), refer to the corresponding command. The only difference is that data response has no SDI-12 service request output compared to aM!, aMC!, aMx!, aMCx! Command.
aV!	attt1<CR><LF> a: Sensor address ttt: Measurement data will be ready in ttt seconds, , the specific value is affected by the parameter "<Filter>". 1:Number of measurement data <CR><LF>: terminates the response aD0! Response data format: a<VERIFY_STATUS><CR><LF>	**Sensor Verification Command** **Example:** Request: 0V! Response: 00011<CR><LF> Response: 0<CR><LF> Request: 0D0! Response: 0+0<CR><LF> “+0” indicates sensor normal. “Non-zero” indicates sensor error.
aD0!	[<svvvv><svvvv><svvvv>…][<CRC	Send Data since the last aM, aMC, aC, aCC, aV

aD1! aD2!	>]<CR><LF> [<svvvv>]: data value [<CRC>]: Optional 3 characters CRC checksum, <CR><LF>:terminates the response	command, The data returned depends on the command sent most recently.
aR0!, aRC0!	Response data format: a<±LiquidLevel><±Temperature>[<CRC>]<CR><LF>	Measure the liquid level value and the temperature. The unit of the liquid level value is determined by "Liquid Level Unit <LiquidLevelUnit>". The number of decimal places for the liquid level value is determined by the "Liquid Level Unit <LiquidLevelUnit>" setting set by the user. Refer to the "Decimal Places for Liquid Level Value" section. **Example:** Request: 0R0! Response: 0+0.000+26.16<CR><LF>
aR1!,aRC1!	attt6<CR><LF> a: Sensor address ttt: Measurement data will be ready in ttt seconds, , the specific value is affected by the parameter "<Filter>". 6: Number of measurement data returned by aD0! <CR><LF>: terminates the response aD0! Response data format: a<LiquidLevelType><LiquidLevelRange><LiquidLevelUnit><LiquidLevelDecimalPointNum><±LiquidLevel><±Temperature>[<CRC>]<CR><LF>	Liquid Level Type, Range, Unit, Decimal Point Number, Liquid Level, Temperature (offset corrected) Measurement The unit of the liquid level value is determined by "Liquid Level Unit <LiquidLevelUnit>". The number of decimal places for the liquid level value is determined by the "Liquid Level Unit <LiquidLevelUnit>" setting set by the user. Refer to the "Decimal Places for Liquid Level Value" section. **Example:** Request: 0M1! Response: 00016<CR><LF> Response: 0<CR><LF> Request: 0D0! Response: 0+0+2+0+3+0.000+26.16<CR><LF>
aXR_TUNIT!	aTUNIT=<TemperatureUnit> <TemperatureUnit> is temperature unit: C: Degrees Centigrade (default) F: Degrees Fahrenheit	**Query temperature unit** **Example:** Request: 0XR_TUNIT! Response: 0TUNIT=C<CR><LF>
aXW_TUNIT_<TemperatureUnit>!	aTUNIT=<TemperatureUnit>	**Configure temperature unit** **Example:** Request: 0XW_TUNIT_C! Response: 0TUNIT=C<CR><LF>
aXR_TOFFSE	aTOFFSET=<±TOffset>	**Query temperature offset value**

T!	<±TOffset>: temperature offset value between -10.00~10.00 (default 0.00), it will be effective when issuing a new measurement command. This value will be applied to the parameter "<±Temperature>" for temperature correction.	**Example:** Request: 0XR_TOFFSET! Response: 0TOFFSET=+1.00<CR><LF>
aXW_TOFFSET_<±TOffset>!	aTOFFSET=<±TOffset>	**Configure temperature offset value** **Example:** Request: 0XW_TOFFSET_+1.00! Response: 0TOFFSET=+1.00<CR><LF>
aXR_PLUNIT!	aPLUNIT=<LiquidLevelUnit><CR><LF> <LiquidLevelUnit> is Liquid Level Unit: 0: Meter (m) (default) 1: Centimeter (cm) 2: Millimeter (mm) 3: Inch (inch) 4: Feet (feet) 5: Pa (Pa) 6: hPa (hPa) 7: Kpa (KPa) 8: MPa (MPa) 9: bar (bar) 10: mbar (mbar) 11: psi (psi)	**Query Liquid Level Unit** **Example:** Request: 0XR_PLUNIT! Response: 0PLUNIT=+0<CR><LF>
aXW_PLUNIT_<LiquidLevelUnit>!	aPLUNIT=<LiquidLevelUnit><CR><LF>	**Configure Liquid Level Unit** **Example:** Request: 0XW_PLUNIT_0! Response: 0PLUNIT=+0<CR><LF>
aXR_CALPLUNIT!	aCALPLUNIT=<LiquidLevelUnitForCalib><CR><LF> <LiquidLevelUnitForCalib>: Pressure/Level Calibration Unit. The pressure/level unit used for entering calibration point values during user calibration. 0: Meter (m) (default) 1: Centimeter (cm) 2: Millimeter (mm) 3: Inch (inch) 4: Feet (feet) 5: Pa (Pa) 6: hPa (hPa) 7: Kpa (KPa)	**Query Pressure/Level Calibration Unit** **Example:** Request: 0XR_PLUNIT! Response: 0PLUNIT=+0<CR><LF>

	8: MPa (MPa) 9: bar (bar) 10: mbar (mbar) 11: psi (psi)	
aXW_CALPLUNIT_<LiquidLevelUnitForCalib>!	aCALPLUNIT=<LiquidLevelUnitForCalib><CR><LF>	**Configure Pressure/Level Calibration Unit** **Example:** Request: 0XW_PLUNIT_0! Response: 0PLUNIT=+0<CR><LF>
aXR_LGA!	aLGA=<LocalGravityAcceleration><CR><LF> <LocalGravityAcceleration> is Local Gravity Acceleration. Range: 9.50000~10.00000 Default: 9.80665 Accurately calculate the liquid level value under local gravity acceleration conditions based on the hydrostatic pressure formula: P=ρ*g*h	**Query Local Gravity Acceleration.** **Example:** Request: 0XR_LGA! Response: 0LGA=+9.80665<CR><LF>
aXW_LGA_<LocalGravityAcceleration>!	aLGA=<LocalGravityAcceleration><CR><LF>	**Configure Local Gravity Acceleration.** **Example:** Request: 0XW_LGA_9.80665! Response: 0LGA=9.80665<CR><LF>
aXR_LD!	aLD=<LiquidDensity><CR><LF> <LiquidDensity> is Liquid Density. Range: 0.50000~2.00000 Default: 1.00000 Accurately calculate the liquid level value under different liquid density conditions based on the hydrostatic pressure formula: P=ρ*g*h.	**Query Liquid Density.** **Example:** Request: 0XR_LD! Response: 0LD=+1.00000<CR><LF>
aXW_LD_<LiquidDensity>!	aLD=<LiquidDensity><CR><LF>	**Configure Liquid Density.** **Example:** Request: 0XW_LD_1.00000! Response: 0LD=1.00000<CR><LF>
aXR_FILTER!	aFILTER=<Filter><CR><LF> <Filter>: Pressure/level value filtering strength. Higher values result in greater stability of the data, but the measurement speed decreases. Range: 0~16 (0 is disabled) Default: 16	**Query the filtering strength of the pressure/level values.** **Example:** Request: 0XR_FILTER! Response: 0FILTER=16<CR><LF>
aXW_FILTER_<Filter>!	aFILTER=<Filter><CR><LF>	**Configure the filtering strength of the pressure/level values.**

		Example: Request: 0XW_FILTER_16! Response: 0FILTER=16<CR><LF>
aXR_ZAS!	aZAS=<ZeroPointSuppressRange><CR><LF> <ZeroPointSuppressRange>: Zero-point suppression range. When the value of <±LiquidLevel> is within the following range, the sensor will automatically set the value of <±LiquidLevel> to 0. Range: 0: Disabled (default) 1: 0.1% F.S. 2: 0.2% F.S. 3: 0.5% F.S. 4: 1.0% F.S. 5: 2.0% F.S. 6: 5.0% F.S.	**Query zero-point suppression range.** **Example:** Request: 0XR_ZAS! Response: 0ZAS=+0<CR><LF>
aXR_ZAS!	aZAS=<ZeroPointSuppressRange><CR><LF> <ZeroPointSuppressRange>: Zero Point Suppression Range. When the value of <±LiquidLevel> falls within the following range, the sensor will automatically set the value of <±LiquidLevel> to 0. 0: Disabled 1: 0.1% FS (full scale) 2: 0.2% FS (full scale) 3: 0.5% FS (full scale) (default) 4: 1.0% FS (full scale) 5: 2.0% FS (full scale) 6: 5.0% FS (full scale)	**Query zero-point suppression range.** **Example:** Request: 0XR_ZAS! Response: 0ZAS=+4<CR><LF>
aXW_ZAS_<ZeroPointSuppressRange>!	aZAS=<ZeroPointSuppressRange><CR><LF>	**Configure Zero Point Suppress Range.** **Example:** Request: 0XW_ZAS_4! Response: 0ZAS=+0<CR><LF>
aXR_ZTRIM!	aZTRIM=<LiquidLevelZeroTrimResult>,<LiquidLevelZeroTrimValue>,<LiquidLevelZeroTrimInternalValue><CR><LF>	**Read the zero-trim result.** **Example:** Request: 0XW_ZTRIM! Response: 0ZTRIM=0,0,0.0300000<CR><LF>
aXW_ZTRIM!	aZTRIM=<LiquidLevelZeroTrimResu	**Perform zero-trim operation.**

	lt>,<LiquidLevelZeroTrimValue>,<LiquidLevelZeroTrimInternalValue><CR><LF>	After installing the sensor or before immersion, if the <±LiquidLevel> reading is non-zero while the sensor is suspended in the air, send this command to zero the reading. **Example:** Request: 0XW_ZTRIM! Response: 0ZTRIM=0,0,0.0300000<CR><LF>
aXW_RESETZTRIM!	aRESETZTRIM=<LiquidLevelZeroTrimResult>,<LiquidLevelZeroTrimValue>,<LiquidLevelZeroTrimInternalValue><CR><LF>	**Reset zero-trim to factory default.** **Example:** Request: 0XW_RESETZTRIM! Response: 0RESETZTRIM=0,0,0.0000000<CR><LF>
aXW_CALLOWER_<LiquidLevelCalLowerValue>!	aCALLOWER=<LiquidLevelCalLowerResult>,<LiquidLevelCalLowerValue>,<LiquidLevelCalLowerInternalValue><CR><LF> <LiquidLevelCalLowerValue>: Pressure/Level lower calibration point value. The unit for calibration point data entered during calibration is determined by the parameter "<LiquidLevelUnitForCalib>".	**Calibrate Pressure/Level lower point.** **Example:** **Use 0 KPa as the lower calibration point to perform user calibration.** (1)Set the pressure/level calibration unit "<LiquidLevelUnitForCalib>" to 7, which is "KPa". Request: 0XW_CALPLUNIT_7! Response: 0CALPLUNIT=+0<CR><LF> (2)Connect the sensor to the 0 KPa calibration source. Once the reading stabilizes, send the following command. Request: 0XW_CALLOWER_0.00! Response: 0CALLOWER=0,0.00,0.12<CR><LF> (3)Read the measured values for verification. Request: 0R0! Response: 0+0.00+26.16<CR><LF>
aXR_CALLOWER!	aCALLOWER=<LiquidLevelCalLowerResult>,<LiquidLevelCalLowerValue>,<LiquidLevelCalLowerInternalValue><CR><LF>	Query Pressure/Level lower calibration operation result. **Example:** Request: 0XR_CALLOWER! Response: 0CALLOWER=0,0.00,0.12<CR><LF>
aXW_CALUPPER_<LiquidLevelCalUpperValue>!	aCALUPPER=<LiquidLevelCalUpperResult>,<LiquidLevelCalUpperValue>,<LiquidLevelCalUpperInternalValue><CR><LF> <LiquidLevelCalUppererValue>: Pressure/Level upper calibration point value. The unit for calibration point	**Calibrate Pressure/Level upper point.** **Example:** **Use 10 KPa as the upper calibration point to perform user calibration.** (1)Set the pressure/level calibration unit "<LiquidLevelUnitForCalib>" to 7, which is "KPa".

	data entered during calibration is determined by the parameter "<LiquidLevelUnitForCalib>".	Request: 0XW_CALPLUNIT_7! Response: 0CALPLUNIT=+0<CR><LF> (2)Connect the sensor to the 10 KPa calibration source. Once the reading stabilizes, send the following command. Request: 0XW_CALUPPER_10.00! Response: 0CALUPPER=0,10.00,10.12<CR><LF> (3)Read the measured values for verification. Request: 0R0! Response: 0+10.00+26.16<CR><LF>
aXR_CALUPPER!	aCALUPPER=<LiquidLevelCalUpperResult>,<LiquidLevelCalUpperValue>,<LiquidLevelCalUpperInternalValue><CR><LF>	Query Pressure/Level upper calibration operation result. **Example:** Request: 0XR_CALUPPER! Response: 0CALUPPER=0,10.00,10.12<CR><LF>
aXR_SENSORCFG!	aSENSORCFG=<LiquidLevelType><LiquidLevelRange><LiquidLevelUnit><LiquidLevelDecimalPointNum><CR><LF>	**Query the liquid level sensor type, range, unit and decimal point number.** **Example:** Request: 0XR_SENSORCFG! Response: 0SENSORCFG=0,4<CR><LF>
aXR_SN!	aSN=<ssssssss> <ssssssss> is 8-digits serial number	**Query serial number** **Example:** Request: 0XR_SN! Response: 0SN=12345678<CR><LF>
aXW_SN_<ssssssss>!	aSN=<ssssssss>	**Configure serial number** **Example:** Request: 0XW_SN_ABCDEFGH! Response: 0SN=ABCDEFGH <CR><LF>
aXW_RESETSYSTEM!	aRESETSYSTEM=0<CR><LF>	**Restart the sensor (equivalent to powering on)** **Example:** Request: 0XW_RESETSYSTEM! Response: 0RESETSYSTEM=0<CR><LF>
aXW_RESETCALIB!	aRESETCALIB=0<CR><LF> Restore calibration data to factory settings, including: <LiquidLevelUnitForCalib> <LiquidLevelZeroTrimInternalValue> <LiquidLevelCalLowerValue> <LiquidLevelCalLowerInternalValue> <LiquidLevelCalUpperValue> <LiquidLevelCalUpperInternalValue>	Restore calibration data to factory settings. **Example:** Request: 0XW_RESETCALIB! Response: 0RESETCALIB=0<CR><LF>
aXW_RESETFACTORY!	aRESETFACTORY=0<CR><LF> Restore all parameters to factory	Restore all parameters to factory settings.

	settings, including: <LiquidLevelUnit> <TemperatureUnit> <Filter> <AcqInterval> <ZeroPointSupressRange> <±TOffset> <LocalGravityAcceleration> <LiquidDensity> <LiquidLevelUnitForCalib> <LiquidLevelZeroTrimInternalValue> <LiquidLevelCalLowerValue> <LiquidLevelCalLowerInternalValue> <LiquidLevelCalUpperValue> <LiquidLevelCalUpperInternalValue>	**Example:** Request: 0XW_RESETFACTORY! Response: 0RESETFACTORY=0<CR><LF>

7 RS485 Communication

7.1 Modbus Protocol

Modbus Protocol is widely used to establish master-slave communication between intelligent devices or sensors. A MODBUS message sent from a master to a slave contains the address of the slave, the function code (e.g. 'read register' or 'write register'), the data, and a check sum (LRC or CRC).

The sensor is RS485 interface with Modbus protocol. The default serial communication settings is slave address 1, modbus rtu, 9600bps, 8 databits and 1 stop bit. All communication settings can be changed with modbus command, and take effect after the sensor is powered on again.

Following modbus function code are supported by sensor.

Modbus Function Code 0x03 : used for reading holding register.

Modbus Function Code 0x04 : used for reading input register.

Modbus Function Code 0x06 : used for writing single holding register.

Modbus Function Code 0x10: used for writing multiple holding register.

7.2 Modbus Register

Parameters	Register (HEX/DEC)	Data Type	Modbus Function	Range and Comments	Default Value
Liquid Level Type Indicator <LiquidLevelType>	0x0000 /0	UINT16 RO	3/4	Refer to the description of the <LiquidLevelType> register.	0
Liquid Level Range Indicator <LiquidLevelRange>	0x0001 /1	UINT16 RO	3/4	Refer to the description of the <LiquidLevelRange> register.	N/A
Liquid Level Unit Indicator <LiquidLevelUnit>	0x0002 /2	UINT16 RO	3/4	Refer to the description of the <LiquidLevelUnit> register.	N/A
Liquid Level Decimal Point Number Indicator <LiquidLevelDecimal PointNum>	0x0003 /3	UINT16 RO	3/4	Refer to the description of the <LiquidLevelDecimalPointNum> register.	N/A
Liquid Level Value <±LiquidLevel_INT32>	0x0004 /4 (High 16 bits) 0x0005 /5 (Low 16 bits)	INT32 RO	3/4	Liquid Level Value Note: The unit of the liquid level value is determined by “Liquid Level Unit <LiquidLevelUnit>“. Note: The number of decimal	N/A

				places for the liquid level value is determined by the “Liquid Level Unit <LiquidLevelUnit>“ setting set by the user. Refer to the “Decimal Places for Liquid Level Value” section.	
Temperature (Offset Corrected) <±Temperature_INT32>	0x0006 /6 (High 16 bits) 0x0007 /7 (Low 16 bits)	INT32 RO	3/4	-4000 - 8000 correspond to -40.00 - 80.00 (℃); -4000 - 17600 correspond to -40.00 - 176.00 (℉) Note: The value unit is determined by the <TemperatureUnit> parameter.	N/A
Reserved <RESERVED>	0x0008 /8	INT16 RO	3/4	Reserved	0
Reserved <RESERVED>	0x0009 /9	INT16 RO	3/4	Reserved	0
Reserved <RESERVED>	0x000A /10	INT16 RO	3/4	Reserved	0
Reserved <RESERVED>	0x000B /11	INT16 RO	3/4	Reserved	0
Reserved <RESERVED>	0x000C /12	INT16 RO	3/4	Reserved	0
Reserved <RESERVED>	0x000D /13	INT16 RO	3/4	Reserved	0
Reserved <RESERVED>	0x000E /14	INT16 RO	3/4	Reserved	0
Reserved <RESERVED>	0x000F /15	INT16 RO	3/4	Reserved	0
Liquid Level Value Unit <LiquidLevelUnit>	0x0020 /32	UINT16 RW	3/4/6/16	Liquid Level Unit. 0: Meter (m) (default) 1: Centimeter (cm) 2: Millimeter (mm) 3: Inch (inch) 4: Feet (feet) 5: Pa (Pa) 6: hPa (hPa) 7: Kpa (KPa) 8: MPa (MPa) 9: bar (bar) 10: mbar (mbar) 11: psi (psi)	0

Temperature Unit <TemperatureUnit>	0x0021 /33	UINT16 RW	3/4/6/16	0: Degree Celsius 1: Degree Fahrenheit	0
Modbus Byte Order for Floating-Point Registers <FloatByteOrder>	0x0022 /34	UINT16 RW	3/4/6/16	Modbus Byte Order for Floating-Point Registers. 0: Big-endian [ABCD] 1: Little-endian[DCBA] 2: Big-endian byte swap [BADC] 3: Little-endian byte swap[CDAB]	3
Pressure/Liquid Level Value Filter Strength <Filter>	0x0023 /35	UINT16 RW	3/4/6/16	Pressure/level value filtering strength. Higher values result in greater stability of the data, but the measurement speed decreases. Range: 0~16 (0 is disabled) Default: 16	16
Acquisition Interval <AcqInterval>	0x0024 /36	UINT16 RW	3/4/6/16	0: Disabled (Continuous Measurement) 10~43200: 10~43200 (seconds).	0
Zero Suppress Range <ZeroPointSupressRange>	0x0025 /37	UINT16 RW	3/4/6/16	Zero-point suppression range. When the value of <±LiquidLevel> is within the following range, the sensor will automatically set the value of <±LiquidLevel> to 0. 0: Disabled 1: 0.1% FS (full scale) 2: 0.2% FS (full scale) 3: 0.5% FS (full scale) (default) 4: 1.0% FS (full scale) 5: 2.0% FS (full scale) 6: 5.0% FS (full scale)	3
Temperature Offset <TOffset>	0x0040 /64	FLOAT R/W	3/6/16	Temperature Offset. -10.00~+10.00 Note: The value unit is determined by the <TemperatureUnit> parameter.	0.00
Local Gravity Acceleration <LocalGravityAcceleration>	0x0042 /66	FLOAT R/W	3/6/16	9.50000~10.00000 m/S2	9.80665
Liquid Density <LiquidDensity>	0x0044 /68	FLOAT R/W	3/6/16	0.50000~2.00000 g/cm3	1.00000

Restore all parameters to factory settings <ResetFactory>	0x0050 /80	UINT16 RW	3/4/6/16	Reading value is 0; Writing 0xFFFF will restore all parameters to factory settings. Please refer to description of <ResetFactory> parameter.	N/A
Restart the sensor <ResetSystem>	0x0051 /81	UINT16 RW	3/4/6/16	Reading value is 0; Writing 0xFFFF will restart the sensor (equivalent to powering on). Please refer to description of <ResetSystem> parameter.	N/A
Restore calibration data to factory settings <ResetCalib>	0x0052 /82	UINT16 RW	3/4/6/16	Reading value is 0; Writing 0xFFFF will restore calibration data to factory settings. Please refer to description of <ResetCalib> parameter.	N/A
Pressure/Level Calibration Unit <LiquidLevelUnitForCalib>	0x0090 /144	UINT16 RW	3/4/6/16	Pressure/Level Calibration Unit. The pressure/level unit used for entering calibration point values during user calibration. 0: Meter(m)(default) 1: Centimeter(cm) 2: Millimeter(mm) 3: Inch(inch) 4: Feet(feet) 5: Pa(Pa) 6: hPa(hPa) 7: KPa(KPa) 8: MPa(MPa) 9: bar(bar) 10: mbar(mbar) 11: psi(psi)	0
Pressure/level zero trim value <LiquidLevelZeroTrimValue>	0x0100 /256	FLOAT RW	3/4/16	Reading value is always 0. Writing: 0x0000: Zero-trim 0xFFFF: Reset zero-trim	N/A
Internal value for the pressure/level zero trim	0x0102 /258	FLOAT RO	3/4	Internal value for the pressure/level zero trim.	N/A

<LiquidLevelZeroTrimInternalValue>					
Pressure/level zero trim operation result <LiquidLevelZeroTrimResult>	0x0104 /260	FLOAT RO	3/4	Pressure/level zero trim operation result. 0: Operation successful 1: Internal error; 2: Exceeding the zero trimming range (10% F.S.); 3: Sensor error;	N/A
Pressure/Level lower calibration point value <LiquidLevelCalLowerValue>	0x0106 /262	FLOAT RW	3/4/16	Pressure/Level lower calibration point value. The unit for calibration point data entered during calibration is determined by the parameter "<LiquidLevelUnitForCalib>".	N/A
Internal value for the pressure/level lower calibration point <LiquidLevelCalLowerInternalValue>	0x0108 /264	FLOAT RO	3/4	Internal value for the pressure/level lower calibration point.	N/A
Pressure/Level lower calibration operation result <LiquidLevelCalLowerResult>	0x010A /266	FLOAT RO	3/4	Pressure/Level lower calibration operation result. 0: Operation successful; 1: Incorrect input of the low calibration point value (exceeding the range or greater than the high calibration point value);	N/A
Pressure/Level upper calibration point value <LiquidLevelCalUpperValue>	0x010C /268	FLOAT RW	3/4/16	Pressure/Level upper calibration point value. The unit for calibration point data entered during calibration is determined by the parameter "<LiquidLevelUnitForCalib>".	N/A
Internal value for the pressure/level upper calibration point<LiquidLevelCalUpperInternalValue>	0x010E /270	FLOAT RO	3/4	Internal value for the pressure/level upper calibration point.	N/A
Pressure/Level upper calibration operation result <LiquidLevelCalUpperResult>	0x0110 /272	FLOAT RO	3/4	Pressure/Level upper calibration operation result. 0: Operation successful; 1: Incorrect input of the low	N/A

				calibration point value (exceeding the range or greater than the high calibration point value);	
Modbus Slave Address <ModbusSlaveAddress>	0x0200 /512	UINT16 R/W	3/6/16	1-255	1
Serial Comm Baudrate <Baudrate>	0x0201 /513	UINT16 R/W	3/6/16	0: 1200bps 1: 2400bps 2: 4800bps 3: 9600bps 4: 19200bps 5: 38400bps	3
Serial Comm Protocol <Protocol>	0x0202 /514	UINT16 R/W	3/6/16	0: Modbus RTU	0
Serial Comm Parity <Parity>	0x0203 /515	UINT16 R/W	3/6/16	0: NONE 1: EVEN 2: ODD	0
Serial Comm DataBits <DataBits>	0x0204 /516	UINT16 R/W	3/6/16	1: 8 bits	1
Serial Comm StopBits <StopBits>	0x0205 /517	UINT16 R/W	3/6/16	0: 1 stopbit 1: 2 stopbits	0
Reserved <RESERVED>	0x0206 /518	UINT16 R/W	3/6/16	Reserved	0
Reserved <RESERVED>	0x0207 /519	UINT16 R/W	3/6/16	Reserved	0
User Defined Serial Number USERSN	0x0220 /544 0x0221 /545 0x0222 /546 0x0223 /547	UINT16 R/W	3/16	0x0000000000000000- 0xFFFFFFFFFFFFFFFF Those four registers should be read/write in one command.	N/A
Liquid Level Type Indicator<LiquidLevel Type >	0x1000 /4096	FLOAT RO	3/4	Refer to the <LiquidLevelType> register description.	0
Liquid Level Range Indicator<LiquidLevel Range >	0x1002 /4098	FLOAT RO	3/4	Refer to the description of the <LiquidLevelRange> register.	N/A
Liquid Level Unit Indicator<LiquidLevel Unit >	0x1004 /4100	FLOAT RO	3/4	Refer to the description of the <LiquidLevelUnit> register.	N/A

Liquid Level Decimal Point Number Indicator <LiquidLevelDecimal PointNum>	0x1006 /4102	FLOAT RO	3/4	Refer to the description of the <LiquidLevelDecimalPointNum> register.	N/A
Liquid Level Value <±LiquidLevel_FLOAT>	0x1008 /4104	FLOAT RO	3/4	Liquid Level Value Note: The unit of the liquid level value is determined by "Liquid Level Unit <LiquidLevelUnit>".	N/A
Temperature (Offset Corrected) <±Temperature_FLOAT>	0x100A /4106	FLOAT RO	3/4	-40.00-80.00 (℃); -40.00-176.00 (℉). Note: The unit of temperature is determined by the register <TemperatureUnit>.	N/A
Reserved <RESERVED_FLOAT>	0x100C /4108	FLOAT RO	3/4	Reserved	0
Reserved <RESERVED_FLOAT>	0x100E /4110	FLOAT RO	3/4	Reserved	0
Reserved <RESERVED_FLOAT>	0x1010 /4112	FLOAT RO	3/4	Reserved	0
Reserved <RESERVED_FLOAT>	0x1012 /4114	FLOAT RO	3/4	Reserved	0
Reserved <RESERVED_FLOAT>	0x1014 /4116	FLOAT RO	3/4	Reserved	0
Reserved <RESERVED_FLOAT>	0x1016 /4118	FLOAT RO	3/4	Reserved	0
Reserved <RESERVED_FLOAT>	0x1018 /4120	FLOAT RO	3/4	Reserved	0
Reserved <RESERVED_FLOAT>	0x101A /4122	FLOAT RO	3/4	Reserved	0
Reserved <RESERVED_FLOAT>	0x101C /4124	FLOAT RO	3/4	Reserved	0
Reserved <RESERVED_FLOAT>	0x101E /4126	FLOAT RO	3/4	Reserved	0

Liquid Level Type Indicator<LiquidLevel Type >	0x1100 /4352	UINT32 RO	3/4	Refer to the <LiquidLevelType> register description.	0
Liquid Level Range Indicator<LiquidLevel Range >	0x1102 /4354	UINT32 RO	3/4	Refer to the description of the <LiquidLevelRange> register.	N/A
Liquid Level Unit Indicator<LiquidLevel Unit >	0x1104 /4356	UINT32 RO	3/4	Refer to the description of the <LiquidLevelUnit> register.	N/A
Liquid Level Decimal Point Number Indicator <LiquidLevelDecimal PointNum>	0x1106 /4358	UINT32 RO	3/4	Refer to the description of the <LiquidLevelDecimalPointNum> register.	N/A
Liquid Level Value<±LiquidLevel_ INT32>	0x1108 /4360	INT32 RO	3/4	Liquid Level Value Note: The unit of the liquid level value is determined by “Liquid Level Unit <LiquidLevelUnit>“. Note: The number of decimal places for the liquid level value is determined by the “Liquid Level Unit <LiquidLevelUnit>“ setting set by the user. Refer to the “Decimal Places for Liquid Level Value” section.	N/A
Temperature (Offset Corrected) <±Temperature_INT32>	0x110A /4362	INT32 RO	3/4	-4000 - 8000 correspond to -40.00 - 80.00 (℃); -4000 - 17600 correspond to -40.00 - 176.00 (℉) Note: The unit of temperature is determined by the register <TemperatureUnit>.	N/A
Reserved <RESERVED_INT32 >	0x110C /4364	INT32 RO	3/4	Reserved	0
Reserved <RESERVED_INT32 >	0x110E /4366	INT32 RO	3/4	Reserved	0
Reserved <RESERVED_INT32 >	0x1110 /4368	INT32 RO	3/4	Reserved	0

Reserved <RESERVED_INT32>	0x1112 /4370	INT32 RO	3/4	Reserved	0
Reserved <RESERVED_INT32>	0x1114 /4372	INT32 RO	3/4	Reserved	0
Reserved <RESERVED_INT32>	0x1116 /4374	INT32 RO	3/4	Reserved	0
Reserved <RESERVED_INT32>	0x1118 /4376	INT32 RO	3/4	Reserved	0
Reserved <RESERVED_INT32>	0x111A /4378	INT32 RO	3/4	Reserved	0
Reserved <RESERVED_INT32>	0x111C /4380	INT32 RO	3/4	Reserved	0
Reserved <RESERVED_INT32>	0x111E /4382	INT32 RO	3/4	Reserved	0

UINT16:16 bit unsigned integer
INT16:16bit signed integer
RO: Register is Read Only
R/W: Register is Read/Write
HEX: Hexadecimal (data with 0x/0X prefix)
DEC: Decimal
FLOAT: Floating point register. The Modbus transmission byte order is determined by the <FloatByteOrder> register. Please refer to “Modbus Register Detail Description” for more details.

The measurement value register will be set as following when there is sensor error:

Error Value	Modbus Register	Comments
+9999999	<±LiquidLevel_INT32> <±LiquidLevel_FLOAT> <±Temperature_INT32> <±Temperature_FLOAT>	sensor error

7.3 Modbus Register Detail Description

<LiquidLevelType>: Indicates the pressure type of liquid level sensor.		
Description	Indicates the pressure type of liquid level sensor. This value is determined at the time of ordering. 0: Gauge pressure 1: Sealed pressure	Default: 0

	2: Absolute pressure	
Param. Storage	N/A	

<LiquidLevelRange>: Indicates the range of the liquid level sensor		
Description	Indicates the range of the liquid level sensor. This value is determined at the time of ordering. 0: 1 mH2O (Vented Gauge) 1: 2 mH2O (Vented Gauge) 2: 3 mH2O (Vented Gauge) 3: 5 mH2O (Vented Gauge) 4: 10 mH2O (Vented Gauge) 5: 20 mH2O (Vented Gauge) 6: 30 mH2O (Vented Gauge) 7: 50 mH2O (Vented Gauge) 8: 100 mH2O (Vented Gauge) 9: 200 mH2O (Vented Gauge)	Default: N/A
Param. Storage	N/A	

<LiquidLevelUnit>: Liquid Level Value Unit		
Description	Liquid Level Value Unit. 0: Meter(m) 1: Centimeter(cm) 2: Millimeter(mm) 3: Inch(inch) 4: Feet(feet) 5: Pa(Pa) 6: hPa(hPa) 7: KPa(KPa) 8: MPa(MPa) 9: bar(bar) 10: mbar(mbar) 11: psi(psi)	Default: 0
Param. Storage	Non-volatile Memory	

<LiquidLevelDecimalPointNum>: Liquid Level Decimal Point Number Indicator		
Description	Used to indicate the decimal place of the liquid level value. The decimal places of the liquid level value is determined based on the "Liquid Level Unit <LiquidLevelUnit>" set by the user. Refer to the "Decimal Places of Liquid Level Value" section. 0: No decimal 1: 1 decimal place 2: 2 decimal places 3: 3 decimal places	Default: N/A

	4: 4 decimal places 5: 5 decimal places 6: 6 decimal places	
Param. Storage	N/A	

<LiquidLevel_Int32>: Liquid Level Value，INT32		
Description	Liquid Level Value Note: The unit of the liquid level value is determined by "Liquid Level Unit <LiquidLevelUnit>". Note: The decimal places of the liquid level value is determined based on the "Liquid Level Unit <LiquidLevelUnit>" set by the user. Refer to the "Decimal Places of Liquid Level Value" section. **Example:** If the data read from this register is 20000, and the Liquid Level Value Unit set by <LiquidLevelUnit> is "0 (meters)", with the decimal place of this unit being 3 (refer to the "Decimal Places of Liquid Level Value" section), then the actual Liquid Level Value is 20.000 (meters).	Default: N/A
Param. Storage	N/A	

<LiquidLevel_Float>: Liquid Level Value，FLOAT		
Description	Liquid Level Value. Note: The unit of the liquid level value is determined by "Liquid Level Unit <LiquidLevelUnit>". **Example:** If the data read from this register is 20.000, and the Liquid Level Value Unit set by <LiquidLevelUnit> is “0 (meters)”, then the actual Liquid Level Value is 20.000 (meters).	Default: N/A
Param. Storage	N/A	

<Temperature_Int32>: Temperature (Offset Corrected)，INT32		
Description	-4000 - 8000 correspond to -40.00 - 80.00 (℃); -4000 - 17600 correspond to -40.00 - 176.00 (℉) Note: The unit of temperature is determined by the register <TemperatureUnit>.	Default: N/A
Param. Storage	N/A	

<Temperature_Float>: Temperature (Offset Corrected)，FLOAT		
Description	-40.00-80.00 (℃);	Default: N/A

	-40.00-176.00 (°F) Note: The value unit is determined by the <TemperatureUnit> parameter.	
Param. Storage	N/A	

<TemperatureUnit>: Temperature Unit，UINT16		
Description	Temperature Unit. 0: Degree Celsius 1: Degree Fahrenheit	Default: 0
Param. Storage	Non-volatile Memory	

<FloatByteOrder>: Float Byte Order，UINT16		
Description	The Modbus transmission byte order of floating-point register. 0: Big-endian [ABCD] 1: Little-endian[DCBA] 2: Big-endian byte swap [BADC] 3: Little-endian byte swap[CDAB] Example: The 123456.00 in IEC754 is 0x47F12000 (A=47, B=F1, C=20, D=00), then 0: Big-endian [ABCD] 1: Little-endian[DCBA] 2: Big-endian byte swap [BADC] 3: Little-endian byte swap[CDAB]	Default: 3
Param. Storage	Non-volatile Memory	

<Filter>: Pressure/Liquid Level Value Filter Strength，UINT16		
Description	Pressure/level value filtering strength. Higher values result in greater stability of the data, but the measurement speed decreases. Range: 0~16 (0 is disabled)	Default: 16
Param. Storage	Non-volatile Memory	

<AcqInterval>: Acquisition Interval，UINT16		
Description	Acquisition Interval. Refer to the "Sensor Operating Mode" section. 0: Disabled (Continuous Measurement) 10~43200: 10~43200 (seconds).	Default: 0
Param. Storage	Non-volatile Memory	

<±TOffset>: Temperature Offset，FLOAT		
Description	Temperature Offset, This value will be applied to the	Default: 0.00

	parameter "<±Temperature>" for temperature correction. Range: -10.00~+10.00 Note: The value unit is determined by the <TemperatureUnit> parameter.	
Param. Storage	Non-volatile Memory	

<LocalGravityAcceleration>: Local Gravity Acceleration，FLOAT		
Description	Local Gravity Acceleration. Range: 9.50000~10.00000 Accurately calculate the liquid level value under local gravity acceleration conditions based on the hydrostatic pressure formula: P=ρ*g*h	Default: 9.80665
Param. Storage	Non-volatile Memory	

<LiquidDensity>: Liquid Density，FLOAT		
Description	Liquid Density. Range: 0.50000~2.00000 Accurately calculate the liquid level value under different liquid density conditions based on the hydrostatic pressure formula: P=ρ*g*h	Default: 1.00000
Param. Storage	Non-volatile Memory	

<ZeroPointSupressRange>: Zero Suppress Range，UINT16		
Description	Zero-point suppression range. When the value of <±LiquidLevel> is within the following range, the sensor will automatically set the value of <±LiquidLevel> to 0. 0: Disabled 1: 0.1% FS (full scale) 2: 0.2% FS (full scale) 3: 0.5% FS (full scale) (default) 4: 1.0% FS (full scale) 5: 2.0% FS (full scale) 6: 5.0% FS (full scale) Example: The sensor's range is 20 meters. When <ZeroPointSupressRange> is set to “1: 0.1% FS (full scale)”, then the Liquid Level Value will be displayed as 0 when it is within 20 * 0.1% = 0.02 (meters).	Default: 3
Param. Storage	Non-volatile Memory	

<LiquidLevelUnitForCalib>: Liquid Level Unit For Calibration. UINT16		
Description	Pressure/Level Calibration Unit. The pressure/level unit used for entering calibration point values during user calibration. 0: Meter(m)(default) 1: Centimeter(cm) 2: Millimeter(mm) 3: Inch(inch) 4: Feet(feet) 5: Pa(Pa) 6: hPa(hPa) 7: KPa(KPa) 8: MPa(MPa) 9: bar(bar) 10: mbar(mbar) 11: psi(psi)	Default: 0
Param. Storage	Non-volatile Memory	

<LiquidLevelZeroTrimValue>: Pressure/level zero trim value，FLOAT		
Description	Perform zero-trim operation. Reading value is always 0. Writing: 0x0000: Zero-trim 0xFFFF: Reset zero-trim Example: After installing the sensor or before immersion, if the <±LiquidLevel> reading is non-zero while the sensor is suspended in the air, send this command to zero the reading.	Default: 0
Param. Storage	Non-volatile Memory	

<LiquidLevelZeroTrimInternalValue>: Internal value for the pressure/level zero trim，FLOAT		
Description	Internal value for the pressure/level zero trim. This register stores the internal reference value automatically stored when zero-trim is performed. It is for reference only.	Default: 0
Param. Storage	Non-volatile Memory	

<LiquidLevelZeroTrimResult>: Pressure/level zero trim operation result，FLOAT		
Description	Pressure/level zero trim operation result. 0: Operation successful 1: Internal error; 2: Exceeding the zero trimming range (10% F.S.); 3: Sensor error;	Default: 0

Param. Storage	N/A	

<LiquidLevelCalLowerValue>: Pressure/Level lower calibration point value，FLOAT		
Description	Calibrate Pressure/Level lower point. Pressure/Level lower calibration point value. When user performs calibration, the low calibration point value applied to the sensor is written into this register, and then the sensor will perform low point calibration. The unit for calibration point data entered during calibration is determined by the parameter “<LiquidLevelUnitForCalib>“. **Example:** (1) Set the pressure/level calibration unit “<LiquidLevelUnitForCalib>“ to 7, which is "KPa". This indicates that during calibration, the user will input the calibration point data in "KPa" units. (2) Connect the sensor to the 0 KPa calibration source. Once the reading stabilizes, write 0.00 to the <LiquidLevelCalLowerValue> register to complete the calibration.	Default: N/A
Param. Storage	Non-volatile Memory	

<LiquidLevelCalLowerInternalValue>: Internal value for the pressure/level lower calibration point，FLOAT		
Description	Internal value for the pressure/level lower calibration point. This register stores the internal reference value automatically stored when lower point calibration is performed. It is for reference only.	Default: N/A
Param. Storage	Non-volatile Memory	

<LiquidLevelCalLowerResult>: Pressure/Level lower calibration operation result，FLOAT		
Description	Pressure/Level lower calibration operation result. 0: Operation successful; 1: Incorrect input of the low calibration point value (exceeding the range or greater than the high calibration point value); 2: Sensor error;	Default: 0
Param. Storage	N/A	

<LiquidLevelCalUpperValue>: Pressure/Level upper calibration point value，FLOAT		
Description	Calibrate Pressure/Level upper point. Pressure/Level upper calibration point value.	Default: N/A

	When user performs calibration, the upper calibration point value applied to the sensor is written into this register, and then the sensor will perform upper point calibration. The unit for calibration point data entered during calibration is determined by the parameter "<LiquidLevelUnitForCalib>". **Example:** (1) Set the pressure/level calibration unit "<LiquidLevelUnitForCalib>" to 7, which is "KPa". This indicates that during calibration, the user will input the calibration point data in "KPa" units. (2) Connect the sensor to the 10 KPa calibration source. Once the reading stabilizes, write 10.00 to the <LiquidLevelCalUpperValue> register to complete the calibration.	
Param. Storage	Non-volatile Memory	

<LiquidLevelCalUpperInternalValue>: Internal value for the pressure/level upper calibration point，FLOAT		
Description	Internal value for the pressure/level upper calibration point. This register stores the internal reference value automatically stored when upper point calibration is performed. It is for reference only.	Default: N/A
Param. Storage	Non-volatile Memory	

<LiquidLevelCalUpperResult>: Pressure/Level upper calibration operation result，FLOAT		
Description	Pressure/Level upper calibration operation result. 0: Operation successful; 1: Incorrect input of the low calibration point value (exceeding the range or greater than the high calibration point value); 2: Sensor error;	Default: 0
Param. Storage	N/A	

<ResetFactory>: Restore all parameters to factory settings，UINT16		
Description	Reading value is 0; Writing 0xFFFF will restore all parameters to factory settings, including: <LiquidLevelUnit> <TemperatureUnit> <FloatByteOrder>	Default: N/A

	<Filter> <AcqInterval> <ZeroPointSupressRange> <±TOffset> <LocalGravityAcceleration> <LiquidDensity> <LiquidLevelUnitForCalib> <LiquidLevelZeroTrimInternalValue> <LiquidLevelCalLowerValue> <LiquidLevelCalLowerInternalValue> <LiquidLevelCalUpperValue> <LiquidLevelCalUpperInternalValue>	
Param. Storage	Non-volatile Memory	

<ResetSystem>: Restart the sensor，UINT16		
Description	Reading value is 0; Writing 0xFFFF will restart the sensor (equivalent to powering on). Please refer to description of <ResetSystem> parameter.	Default: N/A
Param. Storage	N/A	

<ResetCalib>: Restore calibration data to factory settings，UINT16		
Description	Reading value is 0; Writing 0xFFFF will restore calibration data to factory settings, including: <LiquidLevelUnitForCalib> <LiquidLevelZeroTrimInternalValue> <LiquidLevelCalLowerValue> <LiquidLevelCalLowerInternalValue> <LiquidLevelCalUpperValue> <LiquidLevelCalUpperInternalValue>	Default: N/A
Param. Storage	Non-volatile Memory	

<ModbusSlaveAddress>: Modbus Slave Address		
Description	1-255 0 is the broadcast address for Modbus. Power cycle the sensor for the changes to take effect.	Default: 1
Param. Storage	Non-volatile Memory	

<Baudrate>: Serial communication baud rate		
Description	0: 1200bps 1: 2400bps 2: 4800bps 3: 9600bps 4: 19200bps 5: 38400bps	Default: 3

	Power cycle the sensor for the changes to take effect.	
Param. Storage	Non-volatile Memory	

<Protocol>: Serial communication protocol		
Description	0: Modbus RTU Power cycle the sensor for the changes to take effect.	Default: 0
Param. Storage	Non-volatile Memory	

<Parity>: Serial communication parity bit		
Description	0: None 1: Even 2: Odd Power cycle the sensor for the changes to take effect.	Default: 0
Param. Storage	Non-volatile Memory	

<DataBits>: Serial communication data bits		
Description	1: 8 data bits Power cycle the sensor for the changes to take effect.	Default: 1
Param. Storage	Non-volatile Memory	

<StopBits>: Serial communication stop bit		
Description	0: 1 stop bit 1: 2 stop bits Power cycle the sensor for the changes to take effect.	Default: 0
Param. Storage	Non-volatile Memory	

8 Common Operations

The examples in this chapter assume that the address of the SDI-12 sensor is "0" and that the address of the RS485 sensor is "1".

8.1 Liquid Level Unit and Value

SDI-12

(1) Set liquid level unit to "meters", and read the temperature and liquid level value.

Step	Operation	Comments
1	Request: 0XW_PLUNIT_0! Response: 0PLUNIT=+0<CR><LF> This parameter is saved to non-volatile memory and only needs to be configured once.	Set the "Liquid Level Value Unit<LiquidLevelUnit>" to "Meters".
2	Request: 0M! Response: 00012<CR><LF> Response: 0<CR><LF> Request: 0D0! Response: 0+1.234+26.16<CR><LF> The Liquid Level Value is 1.234 meters and the temperature value is 26.16℃.	When the "Liquid Level Value Unit <LiquidLevelUnit>" is set to "meter", the liquid level data has three decimal places. Refer to the "Decimal Places of Liquid Level Value" section.

RS485

(1) Set liquid level unit to "meters", and read the temperature and the "Liquid Level Value" (INT32 register).

Step	Operation	Comments
1	Write value "0" to the Modbus register "Liquid Level Value Unit <LiquidLevelUnit>"(register address: 0x0020). This parameter is saved to non-volatile memory and only needs to be configured once.	Set the "Liquid Level Value Unit<LiquidLevelUnit>" to "Meters".
2	Read the Modbus register "Liquid Level Value <±LiquidLevel_INT32>" If the read data is 1234, then the Liquid Level Value is: 1234/1000 = 1.234 (meters)	When the "Liquid Level Value Unit <LiquidLevelUnit>" is set to "meter", the liquid level data has 3 decimal places. Refer to the "Decimal Places of Liquid Level Value" section.

(2) Set liquid level unit to "meters", and read the temperature and the "Liquid Level Value" (FLOAT register).

Step	Operation	Comments
1	Write value "0" to the Modbus register "Liquid Level Value Unit <LiquidLevelUnit>"(register address: 0x0020). This parameter is saved to non-volatile memory and only needs to be configured once.	Set the "Liquid Level Value Unit<LiquidLevelUnit>" to "Meters".
2	Read the Modbus register "Liquid Level Value <±LiquidLevel_FLOAT>" If the read data is 1.234, then the Liquid Level Value is 1.234 (meters).	Floating-point data will directly output the final Liquid Level Value according to the "Liquid Level Unit <LiquidLevelUnit>" set by the user, without any additional conversions.

8.2 Set Sensor Operating Mode

SDI-12

(1) The sensor automatically enters into the user requested measurement mode when power on. After receiving the measurement instruction, it conducts the measurement and returns the data, then enters a low-power state.

RS485

(1)Set to continuous measurement mode

Step	Operation	Comments
1	Write 0 to Modbus register "Acquisition Interval <AcqInterval>". This parameter is saved to non-volatile memory and only needs to be configured once.	The sensor is in continuous measurement mode. The sensor operates continuously and data is updated in real time, and has high power consumption.
2	Power cycle the sensor for the changes to take effect.	The sensor enters the continuous measurement mode.

(2) Set the sensor to the periodic measurement mode, with a acquisition interval of 10 seconds.

Step	Operation	Comments
1	Write 10 to Modbus register "Acquisition Interval <AcqInterval>". (10 seconds) This parameter is saved to non-volatile memory and only needs to be configured once.	Set the measurement period to 10 seconds.
2	Power cycle the sensor for the changes to take effect.	The sensor enters a periodic measurement mode, The sensor conducts measurements and updates the data every10 seconds, then enters a low-power state and waits for the next measurement period to arrive.

8.3 Zero-Trim of Liquid Level

The sensor zero point cancellation should be carried out after installation, as the installation state of the sensor may cause the zero point to shift. After installing the sensor or before putting it into liquid, if the <±LiquidLevel> reading is not 0 when the sensor is suspended in the air, this command can be sent to reset the sensor reading to zero.

SDI-12

Step	Operation	Comments
1	Request: 0XW_ZTRIM! Response: 0ZTRIM=0,0,0.0300000<CR><LF>	Send zero-trim command and set the liquid level value to zero.

RS485

Step	Operation	Comments
1	Write 0 to register "Pressure/level zero trim value <LiquidLevelZeroTrimValue> " (register address: 0x0100)	Send zero-trim command and set the Liquid Level Value to zero.

8.4 User Calibration

The sensor is calibrated using a two-point method by applying two endpoint pressures and linearizing all outputs between them. First, adjust the lower calibration value to determine the correct offset. Then, adjust the upper calibration value based on the lower one to correct the slope of the characteristic curve. These calibration values enable users to optimize performance within the specified measurement range at the calibration temperature.

The following examples use a sensor with a measurement range of 0~2 meters of water column. Calibration is performed using the liquid at the lower calibration point "0 meters" and the upper calibration point "2 meters"; or using a pressure calibrator at the lower calibration point "0 KPa" and the upper calibration point "20 KPa".

Note:
(1) When using the liquid for calibration, please correctly set "Local Gravity Acceleration" and "Liquid Density" before calibration. After the calibration, do not modify these settings. (2) When using the pressure calibrator for calibration, after calibration, you can set "Local Gravity Acceleration" and "Liquid Density" based on the actual measured liquid and the location where it is used.

SDI-12

(1) Calibrate using the liquid at the low calibration point "0 meters" and the high calibration point "2 meters".

Step	Operation	Comments
1	Request: 0XW_CALPLUNIT_0! Response: 0CALPLUNIT=+0<CR><LF> This parameter is saved to non-volatile memory and only needs to be configured once.	Set the “Pressure/Level Calibration Unit <LiquidLevelUnitForCalib>” to "meters".
2	Request: 0XW_CALLOWER_0.00! Response: 0CALLOWER=0,0.00,0.12<CR><LF>	Use "0 meters" as the lower calibration point to perform user calibration.
3	Request: 0XW_CALUPPER_2.00! Response: 0CALLOWER=0,2.00,2.12<CR><LF>	Use "2 meters" as the upper calibration point to perform user calibration.
4	Verify the output value of the sensor	N/A

(2) Use the pressure calibrator to conduct calibration at the low calibration point "0 KPa" and the high calibration point "20 KPa".

Step	Operation	Comments
1	Request: 0XW_CALPLUNIT_7! Response: 0CALPLUNIT=+7<CR><LF> This parameter is saved to non-volatile memory and only needs to be configured once.	Set the “Pressure/Level Calibration Unit <LiquidLevelUnitForCalib>” to "KPa".
2	Request: 0XW_CALLOWER_0.00! Response: 0CALLOWER=0,0.00,0.12<CR><LF>	Use "0 KPa" as the lower calibration point to perform user calibration.
3	Request: 0XW_CALUPPER_2.00! Response: 0CALLOWER=0,20.00,20.12<CR><LF>	Use "20 KPa" as the upper calibration point to perform user calibration.
4	Verify the output value of the sensor	N/A

RS485

(1) Calibrate using the liquid at the low calibration point "0 meters" and the high calibration point "2 meters".

Step	Operation	Comments
1	Write "0" to register “Pressure/Level Calibration Unit <LiquidLevelUnitForCalib>” (register address: 0x0090). This parameter is saved to non-volatile memory and only needs to be configured once.	Set the “Pressure/Level Calibration Unit <LiquidLevelUnitForCalib>” to "meters".
2	Write "0.00" to the Modbus register "Pressure/Level lower calibration point value <LiquidLevelCalLowerValue>"	Use "0 meters" as the lower calibration point to perform user calibration.
3	Write "2.00" to the Modbus register "Pressure/Level upper calibration point value <LiquidLevelCalUpperValue>"	Use "2 meters" as the upper calibration point to perform user calibration.
4	Verify the output value of the sensor	N/A

(2) Use the pressure calibrator to conduct calibration at the low calibration point "0 KPa" and the high calibration point "20 KPa".

Step	Operation	Comments
1	Write "7" to register “Pressure/Level Calibration Unit <LiquidLevelUnitForCalib>” (register address: 0x0090). This parameter is saved to non-volatile memory and only needs to be configured once.	Set the “Pressure/Level Calibration Unit <LiquidLevelUnitForCalib>” to "KPa".
2	Write "0.00" to the Modbus register "Pressure/Level lower calibration point value <LiquidLevelCalLowerValue>"	Use "0.00 KPa" as the lower calibration point to perform user calibration.
3	Write "20.00" to the Modbus register "Pressure/Level upper calibration point value <LiquidLevelCalUpperValue>"	Use "20.00 KPa" as the upper calibration point to perform user calibration.
4	Verify the output value of the sensor	N/A

Appendix A SDI-12 Sensor Testing and Settings

The user can test the communication or set the parameters with the SDI-12 sensors in the following method.

■ Use any kind of master device that supports the SDI-12 interface (such as data acquisition device, data logger, etc.) to communicate with the sensor or set the parameters.

■ Use a computer to communicate with the sensor through the SDI-12 converter (such as the SDI12ELF20 converter) and to set the parameters.

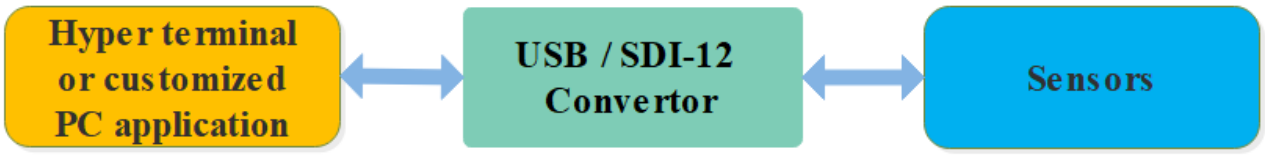
This chapter mainly introduces the communication or parameter setting on a computer for sensor through the SDI-12 converter (SDI12ELF20).

A.1 Testing SDI-12 Sensors with SDI12ELF20 Converter

SDI12ELF20 is a communication converter between USB master device and SDI-12 sensor. It supports bidirectional transparent transmission of SDI-12 communication data and is used to control or test SDI-12 compatible sensors or devices. The USB master device can be a computer, Raspberry PI and other hosts that support USB interface.

SDI12ELF20 Converter User Manual
https://www.infwin.com/sdi12elf20-sdi-12-to-usb-converter/

In this example, a computer is used as a USB host to connect the sensor through the SDI12ELF20 converter for SDI-12 communication test.



Installation steps:

■ Install USB Virtual COM port driver on PC, laptop or other USB master device. The converter uses the CH340C as the USB bridge chip. Download and install the CH340C driver and install it. After the converter is connected to the PC, a COM port is added to the system port. Use this port number in the debugging software to debug the communication with the converter.

Driver Download
https://www.infwin.com/resource-usb-to-serial-port-driver-ch340-series

■ Connect the converter to a PC, laptop or other USB master device through USB port

■ Connect the sensor of the SDI-12 port to the converter

■ The sensor can be powered by the power output that comes with the converter or by an external power supply which has common POWER GROUND with the converter power supply

■ Users can use any serial communication software for SDI-12 communication, such as Terminal or HyperTerminal, The default communication parameters of SDI12ELF20 is 9600bps, none parity, 8 data bits, 1 stop bit. Please use ASCII mode to send and receive data.

Testing Software Download	
Terminal (universal serial port	https://www.infwin.com/resource-serial-port-com-development-tool/

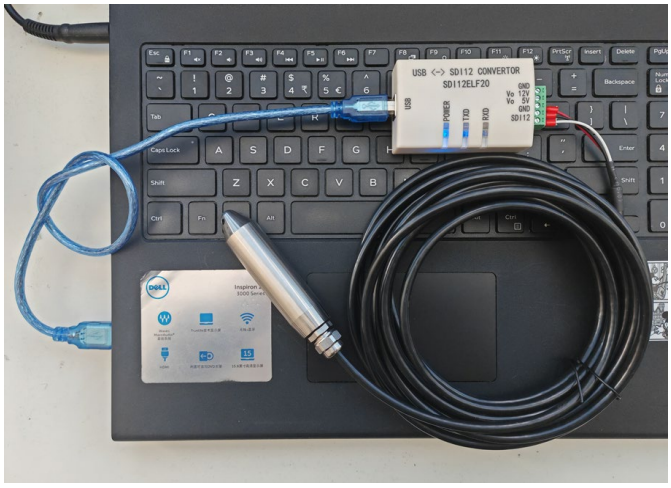
debugging tool)	
HyperTerminal (universal serial port debugging tool)	https://www.infwin.com/resource-serial-port-com-development-tool/
SensorOneSetSDI12 (SDI-12 sensor configuration utility)	https://www.infwin.com/resource-sensoronesetsdi12-configuration-utility-for-sdi-12-sensors/

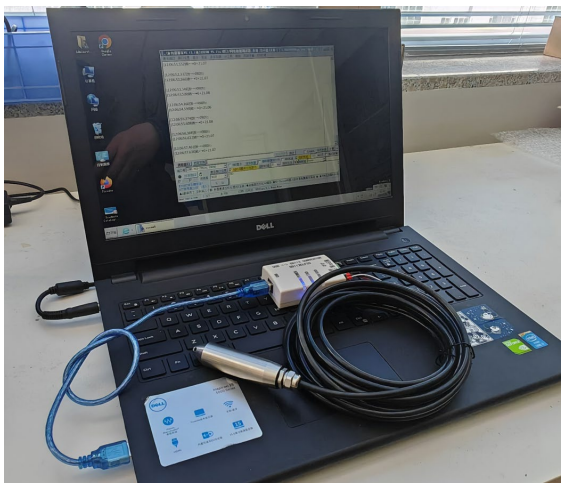
A.2 Testing Example

In this example, we use the SDI12ELF20 converter to communicate between a computer and the rugged temperature sensor DigiTEMP, The power supply of DigiTEMP is also provided by SDI12ELF20.

DigiTEMP Rugged Temperature Sensor User Manual
https://www.infwin.com/digitemp-rugged-digital-temperature-sensor-sdi12-rs485-modbus/

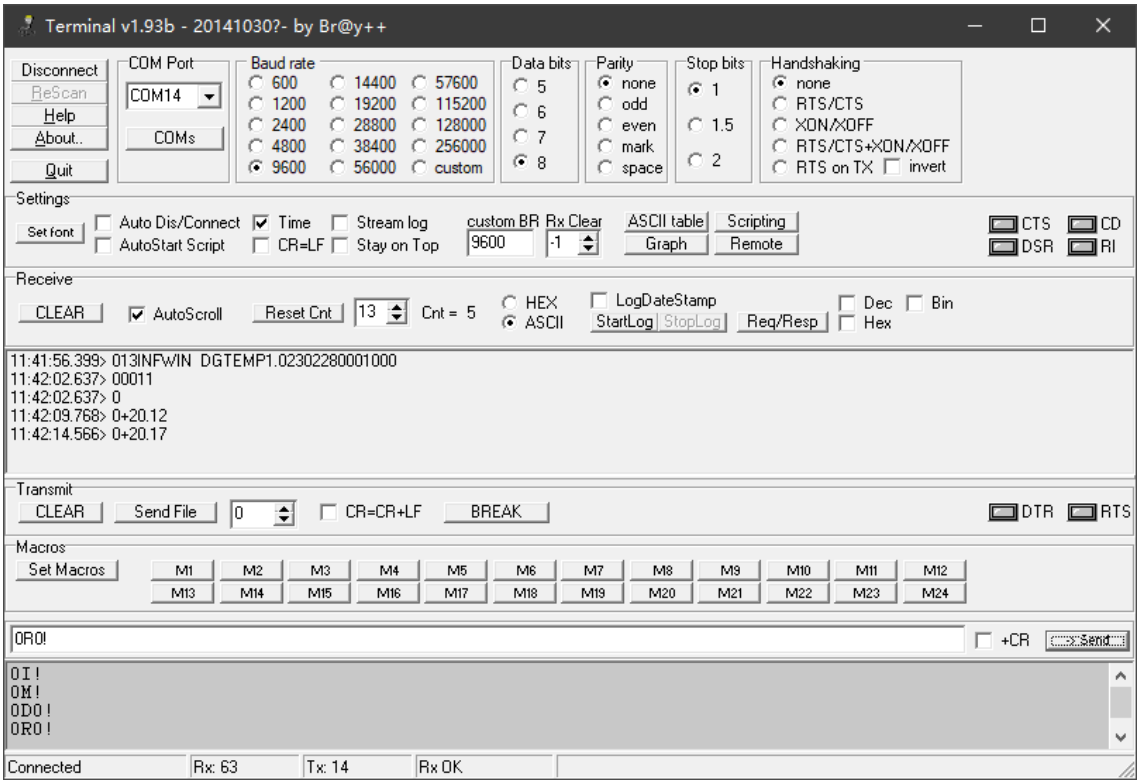
■ Connections





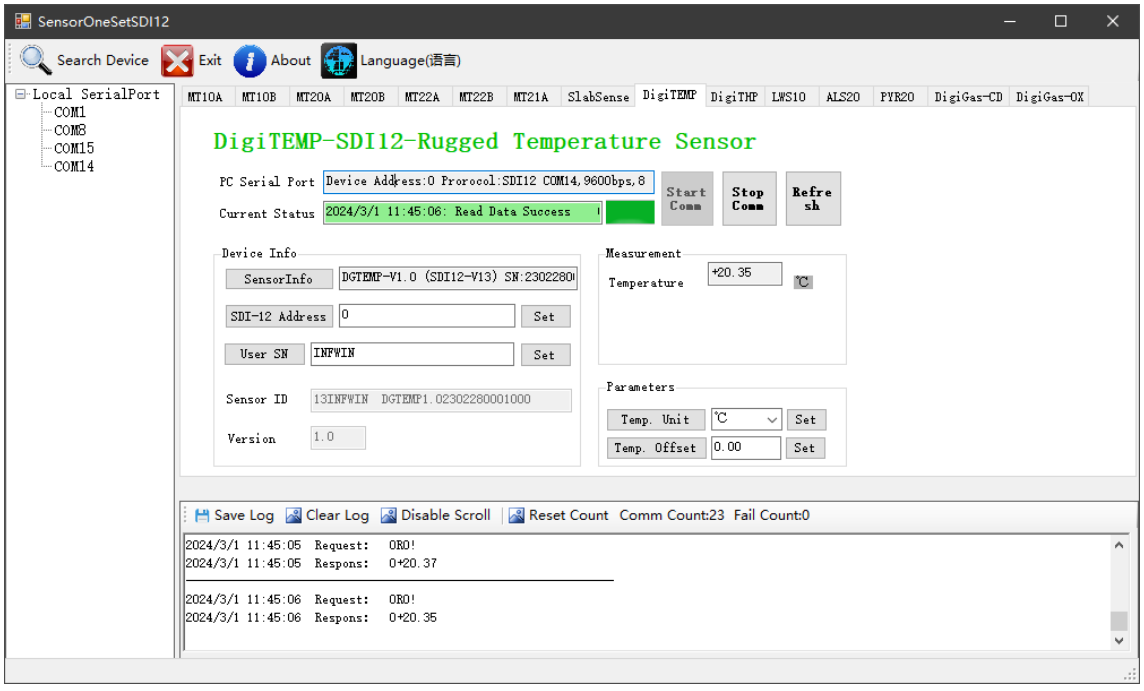
■ Testing with serial port utility “Terminal”

Take “Terminal” as an example, when debugging, please select the corresponding serial port number, baudrate is set to 9600bps, none parity, 8 data bits, 1 stop bit (the default communication Settings of SDI12ELF20), open the serial port and input the SDI-12 command and send. Please note that the ASCII format should be used for data communication.



■ Testing with SDI-12 sensor configuration utility “SensorOneSetSDI12”

Start up the application, select the corresponding product page DigiTEMP, click "start communication" and choose the proper serial port number, 9600bps, none parity, 8 data bits, 1 stop bit (SDI12ELF20 default communication Settings) and start communication.



Appendix B RS485 Sensor Testing and Settings

User can test the communication or set parameters for the sensors with RS485 interface in the following method.

■ Use any kind of master device that supports the RS485 interface (such as data acquisition device, data logger, etc.) to communicate with the sensor or set the parameters.

■ Use a computer to communicate with the sensor through the RS485 converter (such as the USB/RS85 converter) to set the parameters.

This chapter mainly introduces the communication or parameter setting on a computer for sensor through the USB/RS485 converter.

B.1 Testing RS485 Sensors with RS485 Converter

RS485 converter is a communication converter between USB master device and RS485 sensor. It supports bidirectional transparent transmission of RS485 communication data and is used to control or test RS485 compatible sensors or devices. The USB master device can be a computer, Raspberry PI and other hosts that support USB interface.

In this example, a computer is used as a USB host to connect the sensor through the USB/RS485 converter for communication testing.

Installation steps:

■ Install USB Virtual COM port driver on PC, laptop or other USB master device. After the converter is connected to the PC, a COM port is added to the system port. Use this port number in the debugging software to debug the communication with the converter.

■ Connect the converter to a PC, laptop or other USB master device through USB port

■ Connect the sensor to the RS485 port to the converter

■ The sensor can be powered by the power output that comes with the converter or by an external power supply.

■ Users can use any Modbus serial communication software for RS485 communication, such as QModbus.

Testing Software Download	
QModbus (Modbus Debug Tool (For Serial Port))	https://www.infwin.com/modbus-debug-tool-for-serial-port/
SensorOneSet (sensor configuration utility)	https://www.infwin.com/sensoroneset-configuration-utility-for-rs485-sensors/

B.2 Testing Example

In this example, we use the USB/RS485 converter to communicate between a computer and the rugged temperature sensor DigiTEMP, The power supply of DigiTEMP is also provided by USB/RS485 converter.

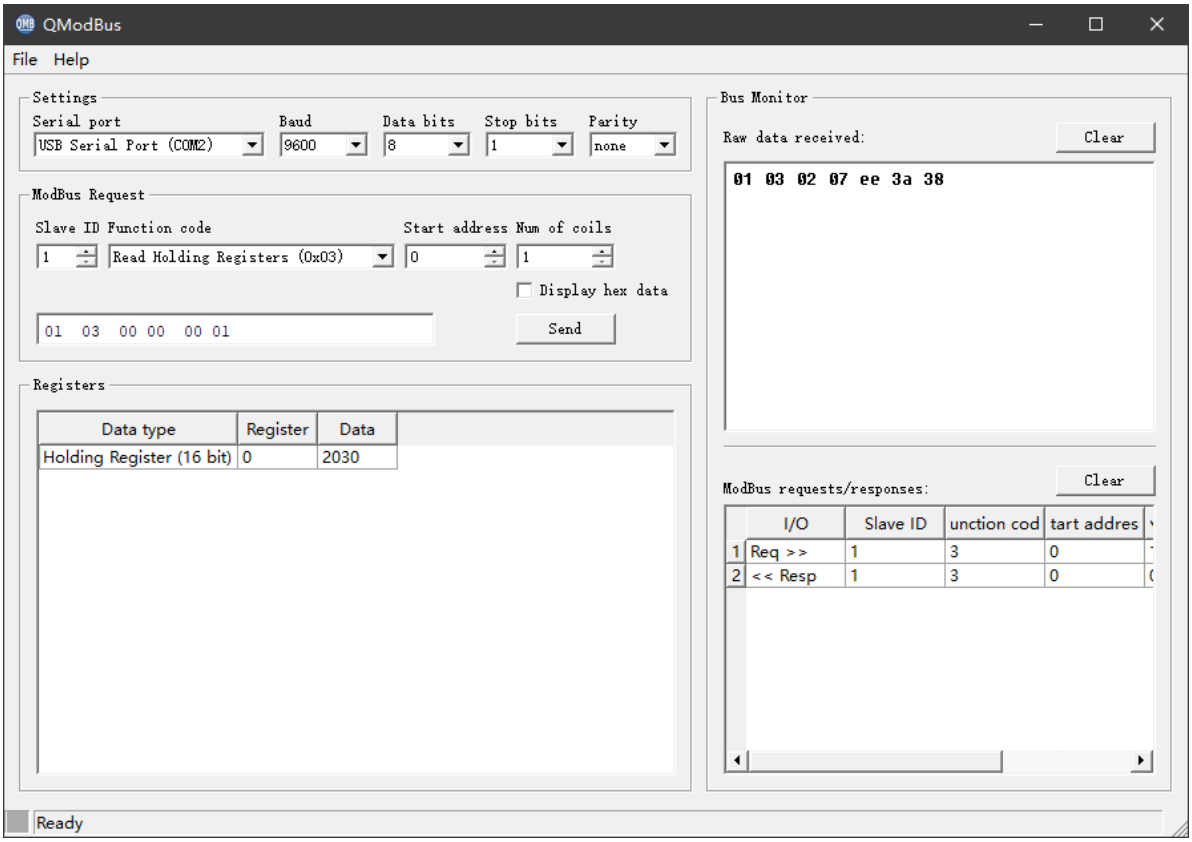
DigiTEMP Rugged Temperature Sensor User Manual
https://www.infwin.com/digitemp-rugged-digital-temperature-sensor-sdi12-rs485-modbus/

Connections

Testing with serial port utility “QModbus”

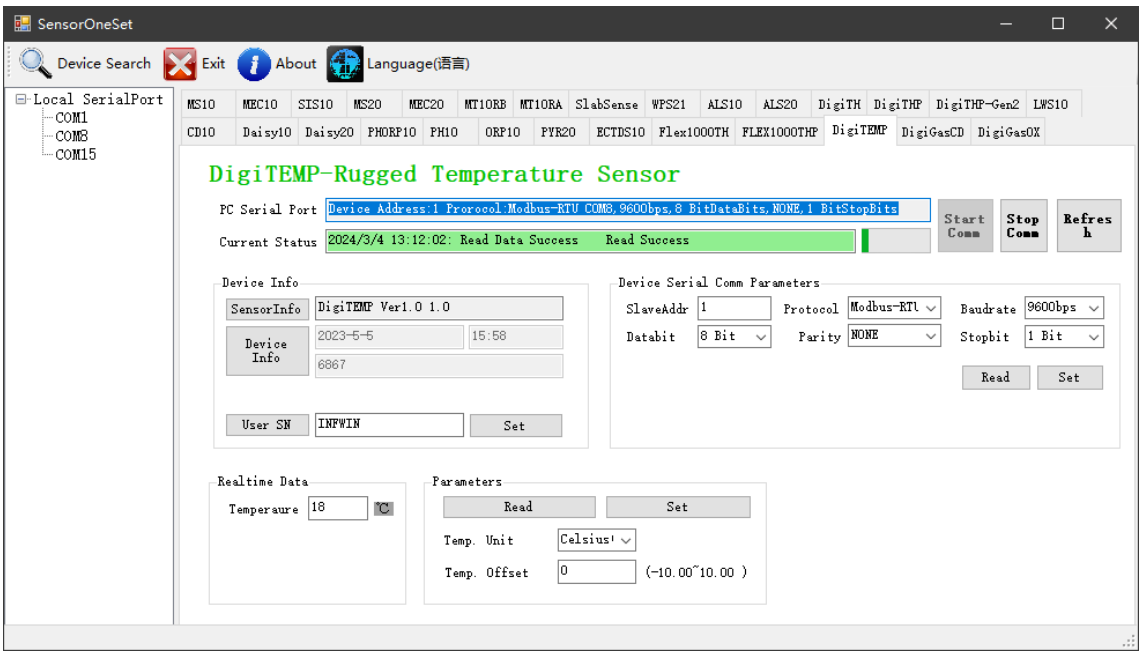
Take “QModbus” as an example, when debugging, please select the corresponding serial port

number, baudrate is set to 9600bps, none parity, 8 data bits, 1 stop bit (the default communication Settings of DigiTEMP), open the serial port and input the Modbus-RTU command and send.



■ Testing with RS485 sensor configuration utility “SensorOneSet”

Start up the application, select the corresponding product page DigiTEMP, click "start communication" and choose the proper serial port number, 9600bps, none parity, 8 data bits, 1 stop bit (DigiTEMP default communication Settings) and start communication.


SensorOneSet
Device Search
Exit
About
Language(语言)
Local SerialPort
COM1
COM3
COM15
MS10 MEC10 SIS10 MS20 MEC20 MT10RB MT10RA SlabSense WPS21 ALS10 ALS20 DigiTH DigiTHP DigiTHP-Gen2 LWS10
CD10 Daisy10 Daisy20 PHORP10 PH10 ORP10 PYR20 ECTDS10 Flex1000TH FLEX1000THP DigiTEMP DigiGasCD DigiGasOX
DigiTEMP-Rugged Temperature Sensor
PC Serial Port
Device Address:1 Prorocol:Modbus-RTU COM3,9600bps,8 BitDataBits,NONE,1 BitStopBits
Current Status
2024/3/4 13:12:02: Read Data Success Read Success
Start Comm
Stop Comm
Refres h
Device Info
SensorInfo
DigiTEMP Ver1.0 1.0
Device Info
2023-5-5
15:58
6867
User SN
INFWIN
Set
Device Serial Comm Parameters
SlaveAddr
1
Protocol
Modbus-RTU
Baudrate
9600bps
Databit
8 Bit
Parity
NONE
Stopbit
1 Bit
Read
Set
Realtime Data
Temperaure
18
℃
Parameters
Read
Set
Temp. Unit
Celsius
Temp. Offset
0
(-10.00~10.00)

Copyright and Trademark

Version Control

Date	Version	Comment	Updated by
2024-04-23	V1.0	Initial Creation	sl51930