DigiTS

Digital Temperature String (SDI-12 Interface)

Digital Temperature String (RS485 Interface)

User Manual

Index

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

2.1 Brief

The DigiTS Temperature String consists of an arrangement of temperature sensors mounted in rugged reinforced cable. Temperature points are able to provide long-term connection in all mediums. Each DigiTS is manufactured to the client's specific requirements.

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.

Features

- Accurate and stable measurements
- Each sensor is individually addressed and referenced to its depth
- Digital Output SDI-12 or RS485 with built-in surge protection
- Low power consumption
- Robust waterproof construction and reinforced cable
- Reverse power protection and Built-in TVS/ESD protection
- Excellent price-performance ratio

Applications

- Permafrost profile temperature
- Ocean and lake profile temperature
- Temperature field measurement
- Scientific Research

Specifications	
Output Interface	Optional: SDI-12, V1.3 Optional: RS485, Modbus-RTU
Power Supply	9-28V DC
Power Consumption	SDI-12 Interface: Quiescent current: < Number of temperature nodes * 20uA @12V DC Current during measuring: < Number of temperature nodes * 1mA @12V DC Current during data reading:< 20mA + Number of temperature nodes * 1mA @12V DC RS485 Interface: Quiescent current: < Number of temperature nodes * 400uA @12V DC Current during measuring: < Number of temperature nodes * 1.2mA @12V DC Current during data reading:< 20mA + Number of temperature nodes * 1.2mA @12V DC
Temperature Accuracy	Operating Range: -55~85°C Accuracy (Typical): ±0.2°C over -40~85°C Accuracy (Worst Case): ±0.4°C over -40~85°C; ±0.5°C over -55~-40°C Resolution: 0.004°C
Measurement Interval	Configurable, It can be configured to measure while reading data or to measure periodically automatically
Warm-up time	<10 Seconds
Sensor nodes	Max. 36 nodes in one temperature string
Dimension	Sensor Node: 110*23mm (Length * Diameter)
Cable Length	Max. 152m (500 feet)
Minimum sensor spacing	15cm (6 inch)
Material of the contact surface	Sensor Node: Nylon, Cable: PUR
Cable	Rugged kevlar reinforced cable, 20AWG, Cable Diameter<7.5mm
IP Ratings	IP68

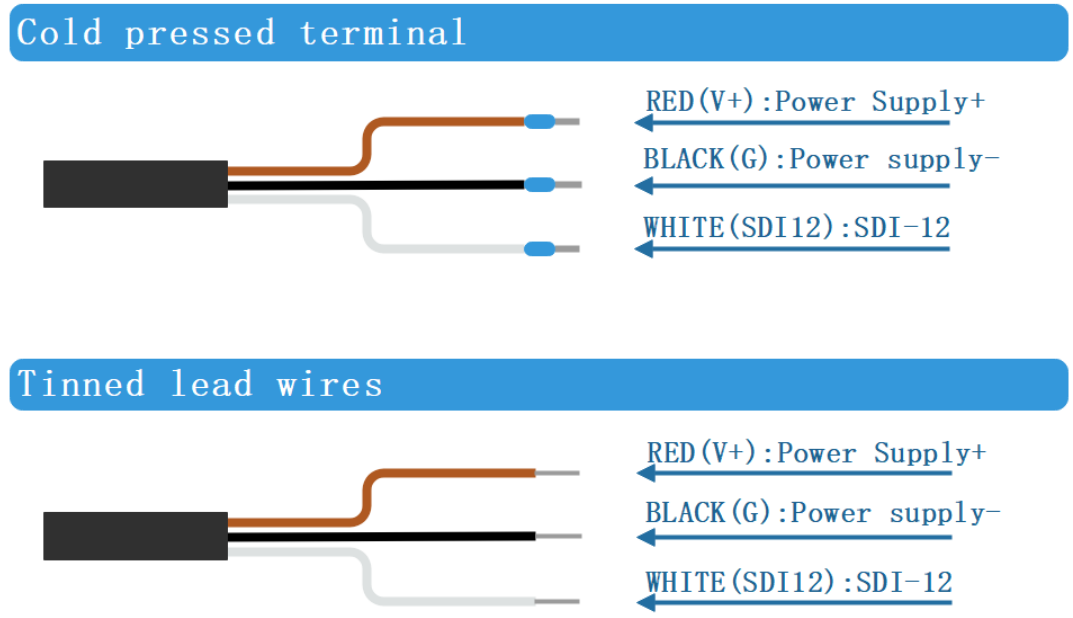
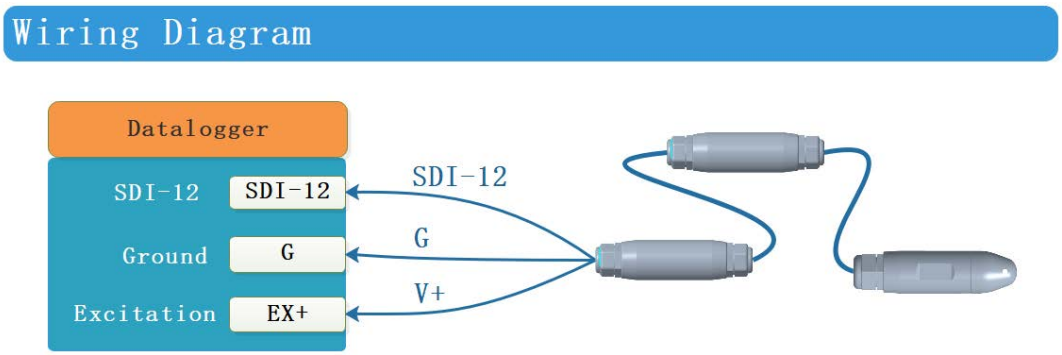
2.2 Sensor node sampling interval

The sampling interval of sensor node is configurable as shown below:

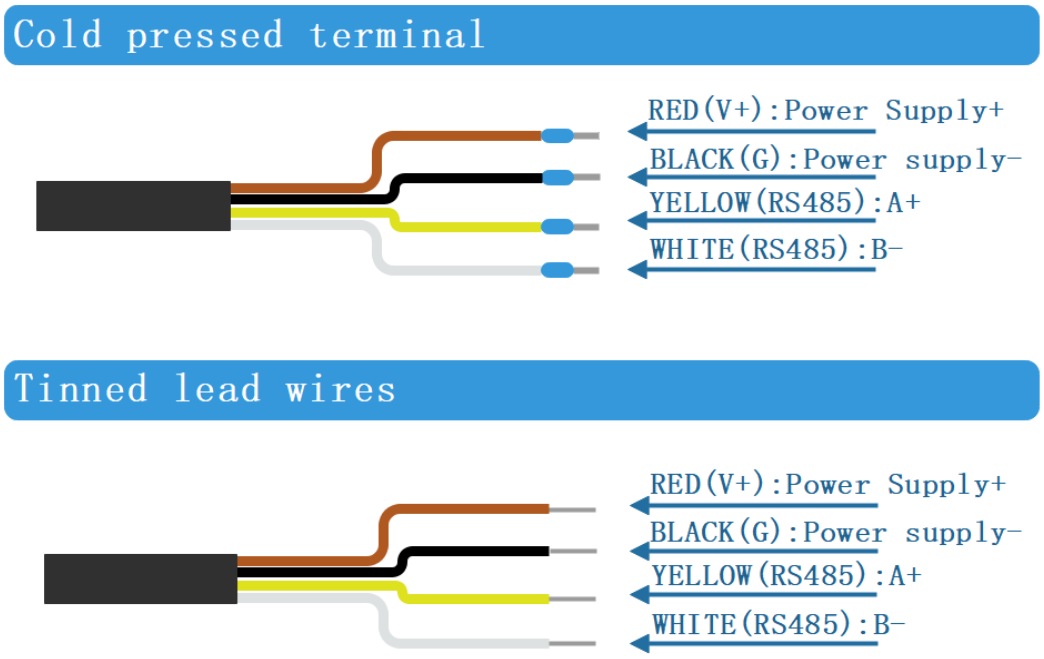
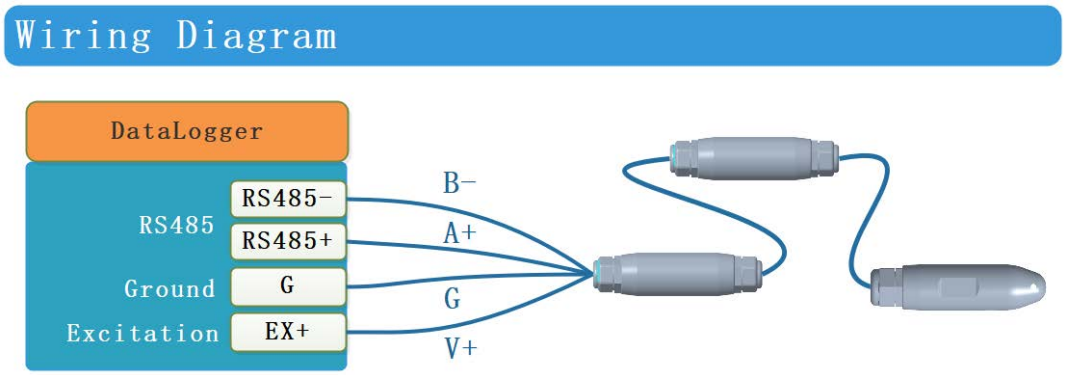
Sampling Interval	Setting Method	Features
Measure while reading data	**SDI-12:** Use SDI-12 command aXW_AAI_<+AsyncAcqInterval>! Set <+AsyncAcqInterval> to 0. **RS485:** Write modbus register ASYNCACQINTERVAL with value 0. Please refer to details in communication chapter	The sensor is usually in sleep mode, and measures after receiving the data reading instruction, and then returns the measurement data. This mode reduces temperature chain power consumption. In this mode, the temperature node determines the maximum and minimum value based on the value when the user reads the data.
Measure periodically automatically (Can be configured to 1-43200s)	**SDI-12:** Use SDI-12 command aXW_AAI_<+AsyncAcqInterval>! Set <+AsyncAcqInterval> to value 1-43200. **RS485:** Write modbus register ASYNCACQINTERVAL with value 1-43200. Please refer to details in communication chapter	The sensor automatically measures at the set time interval, and immediately returns the latest measurement data after receiving the data reading instruction. In this mode, the sensor samples the data at the set time interval and uses it as the basis for judging the maximum and minimum value.

3 Wiring diagrams and Installation

3.1 SDI-12 Interface

Type	Wiring diagram
SDI-12 Interface	
Connections	

3.2 RS485 Interface

Type	Wiring diagram
RS485 Interface	
Connections	

3.3 Installation

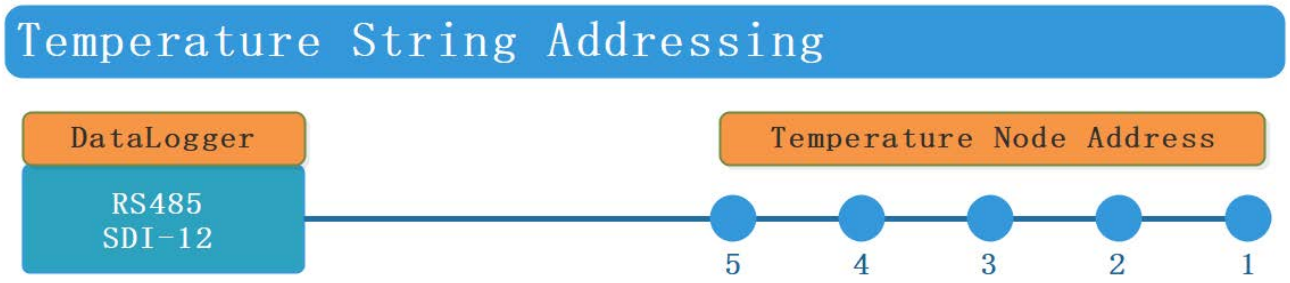
Install the DigiTS directly in the medium that is to be measured. The types of medium that can be measured are varied, including soils, roadbeds, and water. To make the most representative measurement, it is important that consistent contact be made between the temperature profiler and the medium.

The DigiTS consists of the lead cable and the sensor array. The lead cable length is the length between the data logger and sensor array. The sensor array length is the length of cable that

incorporates all temperature sensors in their specified configuration. The installation position of the string and its measurement points are referenced from the first sensor position at the end of the sensor array. This information is needed for the sensor configuration process.

Orient and secure the DigiTS in the measurement medium. Keep materials removed during installation and use that material as backfill. If a burial depth is required for each temperature sensor, you need to know the length of the sensor array, the “measurement from end” metadata of each sensor in the array, and any offset used during installation of the string. You will also need to calculate the depth of each temperature sensor.

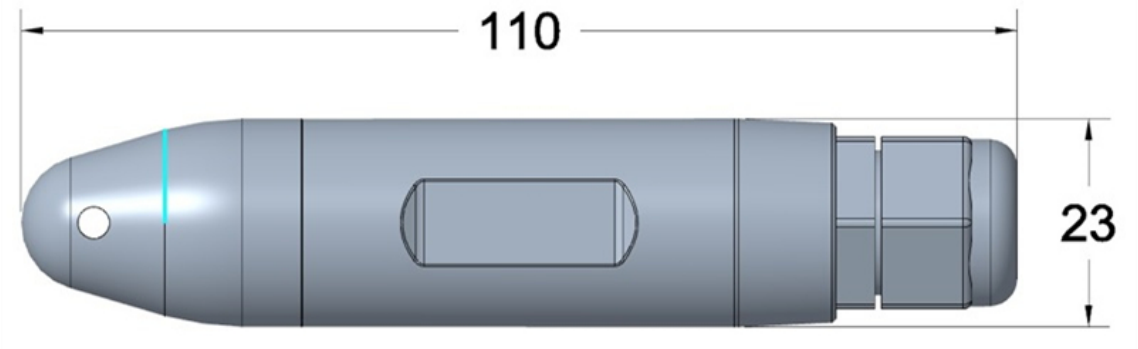
The furthest temperature point from the data logger (or the temperature point at the end of the string) is addressed starting at ‘1’. Each temperature point going up the string towards the data logger changes by one address number/letter.



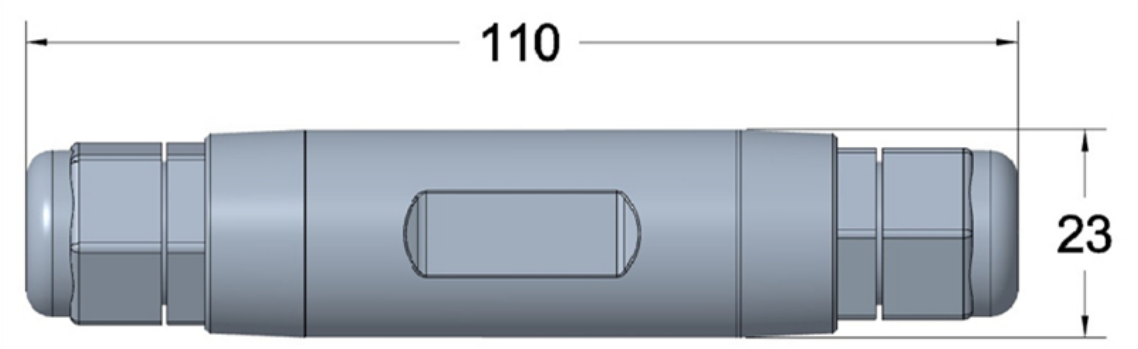
Orient the lead cable of the DigiTS towards the data logger to avoid loops or strain on the cable. Also use a suitable trench or conduit to protect the signal and power cable from damage.

4 Dimension and Ordering Infomation

4.1 Dimension



End Node, Unit: mm

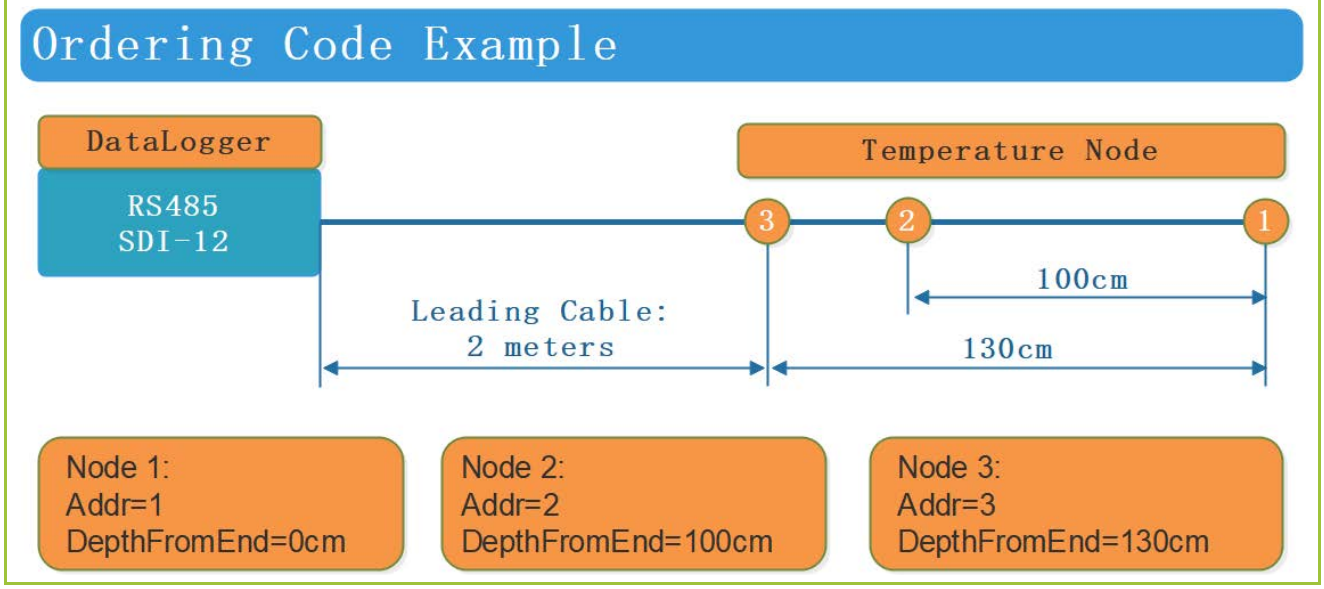


Intermediate node, Unit: mm

4.2 Ordering Information

Ordering Information		
Parameters	**Code**	**Comments**
Code 1:Product Series	DigiTS	DigiTS, Digital Temperature String
Code 2: Power Supply	A X	9-28V DC Customize
Code 3: Output Interface	A B	RS485(Modbus-RTU) SDI-12
Code 4: Connector	B C	Cold pressed terminal Stripped & tinned lead wires
Code 5: Leading Cable Length	002 XXX	2 Meters Customize, XXX is required cable length (Unit: meter)
Code 6: Temperature String Serial Number	[TempStringSerialNum]	Specifies a serial number for the temperature chain, [TempStringSerialNum] ranging from 0-65534. Nodes in the same temperature chain have the same serial number of the temperature chain. If there are multiple temperature chains in the system, ensure that the serial number of each temperature chain is different from that of other temperature chains.
Node 1 Info (end node) :	{Addr/DepthFromEnd}	The end node is the node furthest away from the collector. Addr indicates the slave address of the temperature node. DepthFromEnd is the distance (cm) between the local node and the end node. The DepthFromEnd of the end node is generally set to 0cm.
Node 2 Info: Temperature node next to the Node 1	{Addr/DepthFromEnd}	Addr indicates the slave address of the temperature node. DepthFromEnd is the distance (cm) between the local node and the end node.
……	……	……
Node 36 Info: Temperature node next to the Node 35	{Addr/DepthFromEnd}	Addr indicates the slave address of the temperature node. DepthFromEnd is the distance (cm) between the local node and the end node.

Ordering Code Example
Example：DigiTS-AAB002[0]{1/0}{2/100}{3/130} Product Series: DigiTS, Digital Temperature String; Power Supply: 9-28V DC; Output Interface: RS485(Modbus-RTU)； Connector: Cold pressed terminal; Leading Cable Length: 2 meters; Temperature String Serial Number: 0; Node 1 Info (end node) : Sensor node's Modbus is 1, The distance to the end node is 0cm; Node 2 Info: Sensor node's Modbus is 2, The distance to the end node is 0cm; Node 3 Info: Sensor node's Modbus is 3, The distance to the end node is 0cm;



5 SDI-12 Communication

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_ADI>		ADI protocol CRC Checksum
<CRC>	-	SDI-12 protocol CRC Checksum
<VERIFY_STATUS>	-	Sensor Verification status
<±TempOrig>	℃ ℉	Temperature(Original), The value is output based on the temperature unit setting
<±Temp>	℃ ℉	Temperature(Offseted), The value is output based on the temperature unit setting
<±TOffset>	℃ ℉	Temperature offset, The value is output based on the temperature unit setting Range: -10.00~10.00 Default: 0.00 <±Temp > = <±Temp > + <±TOffset>
<±TempMaxResettable>	℃ ℉	Max temperature value (user resettable). This data is the maximum temperature since last read; it is reset by data reading command.
<±TempMinResettable>	℃ ℉	Min temperature value (user resettable). This data is the maximum temperature since last read; it is reset by data reading command.
<±TempMaxPowerOn>	℃ ℉	Max temperature value (since power on). This data is the maximum temperature since power on.
<±TempMinPowerOn>	℃ ℉	Min temperature value (since power on). This data is the maximum temperature since power on.
<+TempStringSerialNum>	N/A	Temperature chain serial numbers, different temperature chains have different temperature chain

		serial numbers. Value range: 0 to 65534 This data cannot be modified after delivery.
<+TempLocationNum>	N/A	Temperature node location number, different temperature nodes on the same temperature chain have different location numbers, and the location number is related to the factory node address of the temperature node. Value range: 1 to 255 This data cannot be modified after delivery.
<+TempDepthFromEnd>	cm	Distance between the temperature node and the end temperature node. The end temperature node is the temperature node furthest away from the collector. Value range: 0-65535cm This data cannot be modified after delivery.
<TemperatureUnit>	℃ ℉	Temperature unit C: Degree Celsius(Default) F: Degree Fahrenheit
<+AsyncAcqInterval>	Seconds	Auto measurement interval. The sensor automatically measures at the set time interval, and immediately returns the latest measurement data after receiving the data reading instruction. Value range: 0-43200 (Seconds) Default: 1 Note: When set to 0, the sensor will only take measurement samples when the user reads the data. When set to a non-zero value, the sensor automatically measures the sampling at the set time.

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

Error Value	Description
-9999	Sensor Broken

5.1 SDI-12 addressing of temperature node

Each temperature node on the temperature chain has a different default SDI-12 address. The address number starts from 1, corresponding to the temperature node at the end of the temperature chain; The last address corresponds to the topmost sensor (that is, the temperature node closest to the data collector).

Use the default SDI-12 address whenever possible. If multiple sensors are accidentally

assigned the same address, sensor communication will be interrupted. In this case, use aA{! Command (where “a” represents the sensor address to be reset) to restore the corresponding node on the temperature chain to the factory default address value; To restore all nodes on the temperature chain to factory default address values, use SDI-12 broadcast address “?A{!” . If you need to change the SDI-12 address, you are advised to re-address the temperature sensor from the largest address number to avoid duplicate addresses. The relationship between addresses and locations is shown in the following table:

SDI-12 addresses / Temperature Node Location Number（<+TempLocationNum>）		
1 / 1	A / 10	a / 36
2 / 2	B / 11	b / 37
3 / 3	C / 12	c / 38
4 / 4	D / 13	d / 39
5 / 5	E / 14	e / 40
6 / 6	F / 15	f / 41
7 / 7	G / 16	g / 42
8 / 8	H / 17	h / 43
9 / 9	I / 18	i / 44
	J / 19	j / 45
	K / 20	k / 46
	L / 21	l / 47
	M / 22	m / 48
	N / 23	n / 49
	O / 24	o / 50
	P / 25	p / 51
	Q / 26	q / 52
	R / 27	r / 53
	S / 28	s / 54
	T / 29	t / 55
	U / 30	u / 56
	V / 31	v / 57
	W / 32	w / 58
	X / 33	x / 59
	Y / 34	y / 60
	Z / 35	z / 61
		{ (reset to factory address)

5.2 Metadata of temperature node

Each temperature node in the temperature chain contains the following metadata (available

through the SDI-12 command aM1! or aR1! command). This data can be used to identify the metadata of the temperature chain and its individual temperature nodes.

Metadata of the temperature node		
Name	Range	Description
Temperature chain serial number <+TempStringSerialNum>	0~65534	Temperature chain serial numbers, different temperature chains have different temperature chain serial numbers. This data cannot be modified after delivery.
Temperature node location number <+TempLocationNum>	1-255	Temperature node location number, different temperature nodes on the same temperature chain have different location numbers, and the location number is related to the factory node address of the temperature node. This data cannot be modified after delivery.
Distance between the temperature node and the end temperature node <+TempDepthFromEnd>	0-65535cm	Distance between the temperature node and the end temperature node, The distance values in centimeters (cm). These are intended to reflect the profile distances. Typically, the top most temperature point would be designated as 0 cm and if the next temperature point below was 50 cm away, then its distance value would be 50 cm. Users may designate other starting values. These can be configured at the time of ordering.

5.3 SDI-12 Protocol

Request	**Response**	**Comment**
a!	a<CR><LF> a: Sensor address	**Acknowledge Active** **Example:** Request: 1! Response: 1<CR><LF>
aI!	allccccccccmmmmmmvvvxxxxxxxxxxxxx<CR><LF> a: Sensor address ll:SDI-12 Version Number cccccccc: 8 characters vendor identification mmmmmm: 6 characters specifying	**Send Identification** **Example:** Request: 1I! Response: 113INFWIN DigiTS1.02504010006000<CR><LF>

	the sensor model number vvv: 3 characters specifying the sensor version xxxxxxxxxxxxx: 13 characters serial number <CR><LF>: terminates the response	
?!	a<CR><LF> a: Sensor address	**Sensor Address Query** **Example:** Request: ?! Response: 1<CR><LF>
aAb!	b<CR><LF> a:Current Sensor address b:New Sensor address Address range: 1-9 / A-Z/A-Z (No address 0) Use "?" Sending a broadcast message as an address and changing the address to "{" can restore the sensor on the temperature chain to the factory address to correct the address conflict of the temperature node.	**Change Sensor address** **Example:** Request: 1A2! Response: 2<CR><LF> **Example: Restore all nodes on the temperature chain to their factory addresses** Request: ?A{!
aM!, aMC!	a0011<CR><LF> a: Sensor address 001: Measurement data will be ready in 001 seconds, 1: Number of measurement data returned by aD0! <CR><LF>: terminates the response aD0! Response data format: a<±Temp>[<CRC>]<CR><LF>	**Temperature(Offseted) Measurement** **Example:** Request: 1M! Response: 10011<CR><LF> Response: 1<CR><LF> Request: 1D0! Response: 1+23.6602<CR><LF>
aM1!, aMC1!	a0013<CR><LF> a: Sensor address 001: Measurement data will be ready in 001 seconds 3: Number of measurement data returned by aD0!	**Temperature chain serial numbers, Temperature node location number, Distance between the temperature node and the end temperature node Measurement** **Example:** Request: 1M1!

	<CR><LF>: terminates the response aD0! Response data format: a<+TempStringSerialNum><+TempLocationNum><+TempDepthFromEnd>[<CRC>]<CR><LF>	Response: 10013<CR><LF> Response: 1<CR><LF> Request: 1D0! Response: 1+65534+61+65535<CR><LF>
aM2!, aMC2!	a0011<CR><LF> a: Sensor address 001: Measurement data will be ready in 001 seconds, 1: Number of measurement data returned by aD0! <CR><LF>: terminates the response aD0! Response data format: a<±TempMinResettable>[<CRC>]<CR><LF>	**Min temperature value (user resettable) Measurement** **Example:** Request: 1M2! Response: 10011<CR><LF> Response: 1<CR><LF> Request: 1D0! Response: 1+23.6602<CR><LF>
aM3!, aMC3!	a0011<CR><LF> a: Sensor address 001: Measurement data will be ready in 001 seconds, 1: Number of measurement data returned by aD0! <CR><LF>: terminates the response aD0! Response data format: a<±TempMaxResettable>[<CRC>]<CR><LF>	**Max temperature value (user resettable) Measurement** **Example:** Request: 1M3! Response: 10011<CR><LF> Response: 1<CR><LF> Request: 1D0! Response: 1+23.6602<CR><LF>
aM4!, aMC4!	a0011<CR><LF> a: Sensor address 001: Measurement data will be ready in 001 seconds, 1: Number of measurement data returned by aD0! <CR><LF>: terminates the response aD0! Response data format: a<±TempMinPowerOn>[<CRC>]<CR><LF>	**Min temperature value (since power on) Measurement** **Example:** Request: 1M4! Response: 10011<CR><LF> Response: 1<CR><LF> Request: 1D0! Response: 1+23.6602<CR><LF>
aM5!, aMC5!	a0011<CR><LF> a: Sensor address	**Max temperature value (since power on) Measurement**

	001: Measurement data will be ready in 001 seconds, 1: Number of measurement data returned by aD0! <CR><LF>: terminates the response aD0! Response data format: a<±TempMaxPowerOn>[<CRC>]<CR><LF>	**Example:** Request: 1M5! Response: 10011<CR><LF> Response: 1<CR><LF> Request: 1D0! Response: 1+23.6602<CR><LF>
aM6!, aMC6!	a0011<CR><LF> a: Sensor address 001: Measurement data will be ready in 001 seconds, 1: Number of measurement data returned by aD0! <CR><LF>: terminates the response aD0! Response data format: a<±TempMinResettable>[<CRC>]<CR><LF>	**Min temperature value (user resettable) Measurement and reset** **Example:** Request: 1M6! Response: 10011<CR><LF> Response: 1<CR><LF> Request: 1D0! Response: 1+23.6602<CR><LF>
aM7!, aMC7!	a0011<CR><LF> a: Sensor address 001: Measurement data will be ready in 001 seconds, 1: Number of measurement data returned by aD0! <CR><LF>: terminates the response aD0! Response data format: a<±TempMaxResettable>[<CRC>]<CR><LF>	**Max temperature value (user resettable) Measurement and reset** **Example:** Request: 1M7! Response: 10011<CR><LF> Response: 1<CR><LF> Request: 1D0! Response: 1+23.6602<CR><LF>
aM8!, aMC8!	a0018<CR><LF> a: Sensor address 001: Measurement data will be ready in 001 seconds, 8: Number of measurement data returned by aD0! <CR><LF>: terminates the response aD0! Response data format: a<±Temp><+TempStringSerialNum><+TempLocationNum><+TempDepth	**Temperature(Offseted), Temperature chain serial numbers, Temperature node location number, Distance between the temperature node and the end temperature node, Min temperature value (user resettable), Max temperature value (user resettable), Min temperature value (since power on), Max temperature value (since power on) Measurement** **Example:** Request: 1M8! Response: 10018<CR><LF>

	FromEnd>[<CRC>]<CR><LF> aD1! Response data format: a<±TempMinResettable><±TempMax Resettable><±TempMinPowerOn><± TempMaxPowerOn>[<CRC>]<CR>< LF>	Response: 1<CR><LF> Request: 1D0! Response: 1-19.6602+65534+61+65535<CR><LF> Request: 1D1! Response: 1-19.6758-19.5508-19.6758-11.9727<CR><LF>
aM9!, aMC9!	a0018<CR><LF> a: Sensor address 001: Measurement data will be ready in 001 seconds, 8: Number of measurement data returned by aD0! <CR><LF>: terminates the response aD0! Response data format: a<±Temp><+TempStringSerialNum><+TempLocationNum><+TempDepth FromEnd>[<CRC>]<CR><LF> aD1! Response data format: a<±TempMinResettable><±TempMax Resettable><±TempMinPowerOn><± TempMaxPowerOn>[<CRC>]<CR>< LF>	**Temperature(Offseted), Temperature chain serial numbers, Temperature node location number, Distance between the temperature node and the end temperature node, Min temperature value (user resettable), Max temperature value (user resettable), Min temperature value (since power on), Max temperature value (since power on) Measurement and reset** **Example:** Request: 1M9! Response: 10018<CR><LF> Response: 1<CR><LF> Request: 1D0! Response: 1-19.6602+65534+61+65535<CR><LF> Request: 1D1! Response: 1-19.6758-19.5508-19.6758-11.9727<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 data response has no SDI-12 service request output compared to aM!, aMC!, aMx!, aMCx! Command.
aV!	a0012<CR><LF> a: Sensor address 001: Measurement data will be ready in 001 seconds 2: Number of measurement data <CR><LF>: terminates the response aD0! Response data format: a<VERIFY_STATUS><VERIFY_STATUS><CR><LF>	**Sensor Verification Command** **Example:** Request: 1V! Response: 10012<CR><LF> Response: 1<CR><LF> Request: 1D0! Response: 0+0+0<CR><LF>, “+0” indicate sensor normal, “+1” means sensor error.

aD0! aD1! aD2!	[<svvvv><svvvv><svvvv>…][<CRC>]<CR><LF> [<svvvv>]: data value [<CRC>]: Optional 3 characters CRC checksum, <CR><LF>:terminates the response	Send Data since the last aM, aMC, aC, aCC, aV command, The data returned depends on the command sent most recently.
aR0!, aRC0!	Response data format: a<±Temp>[<CRC>]<CR><LF>	**Temperature(Offseted) Measurement** **Example:** Request: 1R0! Response: 1+23.6602<CR><LF>
aR1!, aRC1!	Response data format: a<+TempStringSerialNum><+TempLocationNum><+TempDepthFromEnd>[<CRC>]<CR><LF>	**Temperature chain serial numbers, Temperature node location number, Distance between the temperature node and the end temperature node Measurement** **Example:** Request: 1R1! Response: 1+65534+61+65535<CR><LF>
aR2!, aRC2!	Response data format: a<±TempMinResettable>[<CRC>]<CR><LF>	**Min temperature value (user resettable) Measurement** **Example:** Request: 1R2! Response: 1+23.6602<CR><LF>
aR3!, aRC3!	Response data format: a<±TempMaxResettable>[<CRC>]<CR><LF>	**Max temperature value (user resettable) Measurement** **Example:** Request: 1R3! Response: 1+23.6602<CR><LF>
aR4!, aRC4!	Response data format: a<±TempMinPowerOn>[<CRC>]<CR><LF>	**Min temperature value (since power on) Measurement** **Example:** Request: 1R4! Response: 1+23.6602<CR><LF>
aR5!, aRC5!	Response data format: a<±TempMaxPowerOn>[<CRC>]<CR><LF>	**Max temperature value (since power on) Measurement** **Example:** Request: 1R5! Response: 1+23.6602<CR><LF>
aR6!, aRC6!	Response data format:	**Min temperature value (user resettable)**

	a<±TempMinResettable>[<CRC>]<CR><LF>	**Measurement and reset** **Example:** Request: 1R6! Response: 1+23.6602<CR><LF>
aR7!, aRC7!	Response data format: a<±TempMaxResettable>[<CRC>]<CR><LF>	**Max temperature value (user resettable)** **Measurement and reset** **Example:** Request: 1R7! Response: 1+23.6602<CR><LF>
aR8!, aRC8!	Response data format: a<±Temp><+TempStringSerialNum><+TempLocationNum><+TempDepthFromEnd><±TempMinResettable><±TempMaxResettable><±TempMinPowerOn><±TempMaxPowerOn>[<CRC>]<CR><LF>	**Temperature(Offseted), Temperature chain serial numbers, Temperature node location number, Distance between the temperature node and the end temperature node, Min temperature value (user resettable), Max temperature value (user resettable), Min temperature value (since power on), Max temperature value (since power on) Measurement** **Example:** Request: 1R8! Response: 1-19.6602+65534+61+65535-19.6758-19.5508-19.6758-11.9727<CR><LF>
aM9!, aMC9!	Response data format: a<±Temp><+TempStringSerialNum><+TempLocationNum><+TempDepthFromEnd><±TempMinResettable><±TempMaxResettable><±TempMinPowerOn><±TempMaxPowerOn>[<CRC>]<CR><LF>	**Temperature(Offseted), Temperature chain serial numbers, Temperature node location number, Distance between the temperature node and the end temperature node, Min temperature value (user resettable), Max temperature value (user resettable), Min temperature value (since power on), Max temperature value (since power on) Measurement and reset** **Example:** Request: 1R9! Response: 1-19.6602+65534+61+65535-19.6758-19.5508-19.6758-11.9727<CR><LF>
aXR_TUNIT!	aTUNIT=<TemperatureUnit><CR><LF> <X> is temperature unit: C: degrees Centigrade F: degrees Fahrenheit	**Query temperature unit** **Example:** Request: 0XR_TUNIT! Response: 0TUNIT=C<CR><LF>
aXW_TUNIT_<TemperatureU	aTUNIT=<TemperatureUnit><CR><LF>	**Configure temperature unit** **Example:**

nit>!		Request: 0XW_TUNIT_C! Response: 0TUNIT=C<CR><LF>
aXR_TOFFSET!	aTOFFSET=<±TOffset><CR><LF> <±TOffset>: Temperature offset value. Range: -10.00~10.00, Default: 0.00, The offset value will be applied from the next measurement command. <±TemperatureCalibed> = <±Temperature> + <±TOffset>	**Query temperature offset value** **Example:** Request: 0XR_TOFFSET! Response: 0TOFFSET=+1.00<CR><LF>
aXW_TOFFSET_<±TOffset>!	aTOFFSET=<±TOffset><CR><LF>	**Configure temperature offset value** **Example:** Request: 0XW_TOFFSET_+1.00! Response: 0TOFFSET=+1.00<CR><LF>
aXR_AAI!	aAAI=<+AsyncAcqInterval><CR><LF> <+AsyncAcqInterval>：The sensor automatically measures at the set time interval, and immediately returns the latest measurement data after receiving the data reading instruction. Value range: 0-43200 (Seconds) Default: 1 Note: When set to 0, the sensor will only take measurement samples when the user reads the data. When set to a non-zero value, the sensor automatically measures the sampling at the set time.	Query Auto measurement interval. **Example:** Request: 1XR_AAI! Response: 1AAI=+1<CR><LF>
aXW_AAI_<+AsyncAcqInterval>!	aAAI=<+AsyncAcqInterval><CR><LF>	Set Auto measurement interval. **Example:** Request: 1XW_AAI_10! Response: 1AAI=+1<CR><LF>
aXR_SN!	aSN=<ssssssss><CR><LF> <ssssssss> is 8-digits serial number	**Query serial number** **Example:** Request: 0XR_SN! Response: 0SN=12345678<CR><LF>
aXW_SN_<ssssssss>!	aSN=<ssssssss><CR><LF>	**Configure serial number** **Example:** Request: 0XW_SN_ABCDEFGH! Response: 0SN=ABCDEFGH <CR><LF>

6 RS485 Communication

6.1 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.

6.2 Modbus addressing of temperature node

Each temperature node on the temperature chain has a different default Modbus slave address. The address number starts from 1, corresponding to the temperature node at the end of the temperature chain; The last address corresponds to the topmost sensor (that is, the temperature node closest to the data collector).

Use the default modbus address whenever possible. If multiple sensors are accidentally assigned the same address, sensor communication will be interrupted. In this case, the SLAVEADDRESSRESET register should be used to restore the corresponding node on the temperature chain to the factory default address value; To restore all nodes on the temperature chain to factory default address values, use Modbus broadcast address 0 to operate the SLAVEADDRESSRESET register. If you need to change the modbus address, you are advised to re-address the temperature sensor from the largest address number to avoid duplicate addresses. The relationship between addresses and locations is shown in the following table:

Modbus slave addresses / Temperature Node Location Number（<+TempLocationNum>）		
1 / 1	100 / 100	200 / 200
2 / 2	101 / 101	201 / 201
3 / 3	102 / 102	202 / 202
4 / 4	103 / 103	203 / 203
5 / 5	104 / 104	204 / 204
6 / 6	105 / 105	205 / 205
7 / 7	106 / 106	206 / 206
8 / 8	107 / 107	207 / 207
…	…	…
99 / 99	199 / 199	255 / 255
…	…	0: Broadcast address

6.3 Metadata of temperature node

Each temperature node in the temperature chain contains the following metadata (available through the corresponding Modbus registers). This data can be used to identify the metadata of the temperature chain and its individual temperature nodes.

Metadata of the temperature node		
Name	Range	Description
Temperature chain serial number TEMPSTRING_SERIALNUMBER	0-65534	Temperature chain serial number, different temperature chains have different temperature chain serial numbers. This data cannot be modified after delivery.
Temperature node location number TEMP_LOCATIONNUM	1-255	Temperature node location number, different temperature nodes on the same temperature chain have different location numbers, and the location number is related to the factory node address of the temperature node. This data cannot be modified after delivery.
Distance between the temperature node and the end temperature node TEMP_DEPTHFROMEND	0-65535cm	Distance between the temperature node and the end temperature node. The distance values in centimeters (cm). These are intended to reflect the profile distances. Typically, the top most temperature point would be designated as 0 cm and if the next temperature point below was 50 cm away, then its distance value would be 50 cm. Users may designate other starting values. These can be configured at the time of ordering.

6.4 Modbus Register

Parameters	Register Addr. (HEX/DEC)	Data Type	Modbus Function Code(DEC)	Range and Comments	Default Value
Temperature(Original) TEMP	0x0000 /0	INT16 RO	3/4	-5500-8500 mapping to -55.00-85.00 (℃); -6700-18500 mapping to -67.00-185.00 (℉)	N/A
Temperature chain serial number TEMPSTRING_SERIALNUMBER	0x0001 /1	UINT16 RO	3/4	0~65534	N/A
Temperature node location number TEMP_LOCATIONNUM	0x0002 /2	UINT16 RO	3/4	1-255	N/A
Distance between the temperature node and the end temperature node TEMP_DEPTHFROMEND	0x0003 /3	UINT16 RO	3/4	0-65535（cm）	N/A
Min temperature value (user resettable) TEMP_MINRESETTABLE	0x0004 /4	INT16 RO	3/4	-5500-8500 mapping to -55.00-85.00 (℃); -6700-18500 mapping to -67.00-185.00 (℉)	N/A
Max temperature value (user resettable) TEMP_MAXRESETTABLE	0x0005 /5	INT16 RO	3/4	-5500-8500 mapping to -55.00-85.00 (℃); -6700-18500 mapping to -67.00-185.00 (℉)	N/A
Min temperature value (since power on) TEMP_MINPOWERON	0x0006 /6	INT16 RO	3/4	-5500-8500 mapping to -55.00-85.00 (℃); -6700-18500 mapping to -67.00-185.00 (℉)	N/A
Max temperature value (since power on) TEMP_MAXPOWERON	0x0007 /7	INT16 RO	3/4	-5500-8500 mapping to -55.00-85.00 (℃); -6700-18500 mapping to -67.00-185.00 (℉)	N/A
RESERVED	0x0008 /8	INT16 RO	3/4	Reserved	0
RESERVED	0x0009 /9	INT16 RO	3/4	Reserved	0

RESERVED	0x000A /10	INT16 RO	3/4	Reserved	0
RESERVED	0x000B /11	INT16 RO	3/4	Reserved	0
RESERVED	0x000C /12	INT16 RO	3/4	Reserved	0
RESERVED	0x000D /13	INT16 RO	3/4	Reserved	0
RESERVED	0x000E /14	INT16 RO	3/4	Reserved	0
RESERVED	0x000F /15	INT16 RO	3/4	Reserved	0
Temperature Unit TEMPUNIT	0x0020 /32	UINT16 R/W	3/6/16	0: ℃ 1: ℉	0
Temperature Offset TOFFSET	0x0021 /33	INT16 R/W	3/6/16	-1000～1000 mapping to -10.00～10.00℃	0
Auto measurement interval ASYNCACQINTERVAL	0x0022 /34	UINT16 R/W	3/6/16	0-43200（Seconds）	1
Floating Point Register Byte Order FLOATBYTEORDER	0x0023 /35	UINT16 R/W	3/6/16	Configure floating point register byte order. 0: Big-endian [ABCD] 1: Little-endian[DCBA] 2: Big-endian byte swap [BADC] 3: Little-endian byte swap[CDAB]	3
Reset method of Max/Min temperature value (user resettable) REGCLEARCONFIG	0x0024 /36	UINT16 R/W	3/6/16	Reset method of Max/Min temperature value (user resettable) 0：Automatically reset after reading the corresponding register; 1：Reset by Modbus command	0
Reset Max/Min temperature value (user resettable)	0x0030 /48	UINT16 R/W	3/6/16	Always 0 when read; After writing 0xFFFF, the user-resettable min/max	N/A

REGCLEAR				temperature values are reset.	
Reset to factory configuration FACTORYRESET	0x0031 /49	UINT16 R/W	3/6/16	Always 0 when read; After writing 0xFFFF, the following registers are restored to factory Settings: TEMPUNIT，TOFFSET，ASYNCACQINTERVAL，FLOATBYTEORDER，REGCLEARCONFIG	N/A
Restart the sensor SYSTEMRESET	0x0032 /50	UINT16 R/W	3/6/16	Always 0 when read; After writing 0xFFFF, the sensor will restart.	N/A
Reset the slave Modbus address to the factory default SLAVEADDRESSRESET	0x0033 /51	UINT16 R/W	3/6/16	Always 0 when read; After writing 0xFFFF, the Modbus slave address is reset to the factory value。	N/A
Modbus Slave Address ADDRESS	0x0200 /512	UINT16 R/W	3/6/16	0-255	1
RS485 baudrate BAUDRATE	0x0201 /513	UINT16 R/W	3/6/16	0-5 0: 1200bps 1: 2400bps 2: 4800bps 3: 9600bps 4: 19200bps 5: 38400bps	3: 9600bps
RS485 Protocol PROTOCOL	0x0202 /514	UINT16 R/W	3/6/16	0 0: Modbus RTU	0: Modbus RTU
RS485 parity PARITY	0x0203 /515	UINT16 R/W	3/6/16	0-2 0: NONE 1: EVEN 2: ODD	0: NONE
RS485 data bits DATABITS	0x0204 /516	UINT16 R/W	3/6/16	1 1: 8 bits	1: 8 bits
RS485 Stopbits STOPBITS	0x0205 /517	UINT16 R/W	3/6/16	0-1 0: 1 stopbit 1: 2 stopbits	0: 1 stopbit

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
Temperature (Offseted) TEMP_FLOAT	0x1000 /4096	FLOAT RO	3/4	-55.00-85.00 (℃); -67.00-185.00 (℉)	N/A
Temperature chain serial number TEMPSTRING_SERIALNUMBER_FLOAT	0x1002 /4098	FLOAT RO	3/4	0~65534	N/A
Temperature node location number TEMP_LOCATIONNUM_FLOAT	0x1004 /4100	FLOAT RO	3/4	1-255	N/A
Distance between the temperature node and the end temperature node TEMP_DEPTHFROMEND_FLOAT	0x1006 /4102	FLOAT RO	3/4	0-65535（cm）	N/A
Min temperature value (user resettable) TEMP_MINRESETTABLE_FLOAT	0x1008 /4104	FLOAT RO	3/4	-55.00-85.00 (℃); -67.00-185.00 (℉)	N/A
Max temperature value (user resettable) TEMP_MAXRESETTABLE_FLOAT	0x100A /4106	FLOAT RO	3/4	-55.00-85.00 (℃); -67.00-185.00 (℉)	N/A
Min temperature value (since power on) TEMP_MINPOWERON_FLOAT	0x100C /4108	FLOAT RO	3/4	-55.00-85.00 (℃); -67.00-185.00 (℉)	N/A

RESERVED_FLOAT	0x100E /4110	FLOAT RO	3/4	Reserved	0
RESERVED_FLOAT	0x1010 /4112	FLOAT RO	3/4	Reserved	0
RESERVED_FLOAT	0x1012 /4114	FLOAT RO	3/4	Reserved	0
RESERVED_FLOAT	0x1014 /4116	FLOAT RO	3/4	Reserved	0
RESERVED_FLOAT	0x1016 /4118	FLOAT RO	3/4	Reserved	0
RESERVED_FLOAT	0x1018 /4120	FLOAT RO	3/4	Reserved	0
RESERVED_FLOAT	0x101A /4122	FLOAT RO	3/4	Reserved	0
RESERVED_FLOAT	0x101C /4124	FLOAT RO	3/4	Reserved	0
RESERVED_FLOAT	0x101E /4126	FLOAT 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 byte order is configured by “Floating Point Register Byte Order, FLOATBYTEORDER” register. Please refer to “Modbus Register Detail Descripton” for more details.

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

Error Value	Register Address	Error
+32767	Temperature	sensor broken

6.3 Modbus Register Detail Descripton

TEMP: Temperature(Offseted), INT16		
Range	-5500-8500 mapping to -55.00-85.00 (Temperature Unit set to ℃); -6700-18500 mapping to -67.00-185.00 (Temperature Unit set to ℉)	Default: N/A
Parameter Save	N/A	

Note: Temperature measurement value, It’s calibrated by TOFFSET register.

TEMP_FLOAT: Temperature(Offseted), FLOAT		
Range	-55.00-85.00 (Temperature Unit set to ℃); -67.00-185.00 (Temperature Unit set to ℉)	Default: N/A
Parameter Save	N/A	

Note: Temperature measurement value. It's calibrated by TOFFSET register. The byte order is configured by "Floating Point Register Byte Order, FLOATBYTEORDER" register.

TEMPSTRING_SERIALNUMBER：Temperature chain serial number，UINT16 TEMPSTRING_SERIALNUMBER_FLOAT：Temperature chain serial number，FLOAT		
Range	0~65534	Default: N/A
Parameter Save	N/A	

Note: Temperature chain serial number, different temperature chains have different temperature chain serial numbers. This data cannot be modified after delivery.

TEMP_LOCATIONNUM：Temperature node location number，UINT16 TEMP_LOCATIONNUM_FLOAT：Temperature node location number，FLOAT		
Range	1-255	Default: N/A
Parameter Save	N/A	

Note: Temperature node location number, different temperature nodes on the same temperature chain have different location numbers, and the location number is related to the factory node address of the temperature node. This data cannot be modified after delivery.

TEMP_DEPTHFROMEND： Distance between the temperature node and the end temperature node，UINT16 TEMP_DEPTHFROMEND_FLOAT： Distance between the temperature node and the end temperature node，FLOAT		
Range	0-65535（cm）	Default: N/A
Parameter Save	N/A	

Note: Distance between the temperature node and the end temperature node. The distance values in centimeters (cm). These are intended to reflect the profile distances. Typically, the top most temperature point would be designated as 0 cm and if the next temperature point below was 50 cm away, then its distance value would be 50 cm.

TEMP_MINRESETTABLE：Min temperature value (user resettable)，INT16		
Range	-5500-8500 mapping to -55.00-85.00 (Temperature Unit set to ℃); -6700-18500 mapping to -67.00-185.00 (Temperature Unit set to ℉)	Default: N/A
Parameter Save	N/A	

Note: Min temperature value (user resettable)。

TEMP_MINRESETTABLE _FLOAT：Min temperature value (user resettable)，FLOAT		
Range	-55.00-85.00 (Temperature Unit set to ℃); -67.00-185.00 (Temperature Unit set to ℉)	Default: N/A
Parameter Save	N/A	

Note: Min temperature value (user resettable)。The byte order is configured by “Floating Point Register Byte Order, FLOATBYTEORDER” register.

TEMP_MAXRESETTABLE：Max temperature value (user resettable)，INT16		
Range	-5500-8500 mapping to -55.00-85.00 (Temperature Unit set to ℃); -6700-18500 mapping to -67.00-185.00 (Temperature Unit set to ℉)	Default: N/A
Parameter Save	N/A	

Note: Max temperature value (user resettable)。

TEMP_MAXRESETTABLE _FLOAT：Max temperature value (user resettable)，FLOAT		
Range	-55.00-85.00 (Temperature Unit set to ℃); -67.00-185.00 (Temperature Unit set to ℉)	Default: N/A
Parameter Save	N/A	

Note: Max temperature value (user resettable)。The byte order is configured by “Floating Point Register Byte Order, FLOATBYTEORDER” register.

TEMP_MINPOWERON：Min temperature value (since power on)，INT16		
Range	-5500-8500 mapping to -55.00-85.00 (Temperature Unit set to ℃); -6700-18500 mapping to -67.00-185.00 (Temperature Unit set to ℉)	Default: N/A
Parameter Save	N/A	

Note: Min temperature value (since power on)。

TEMP_MINPOWERON _FLOAT：Min temperature value (since power on)，FLOAT		
Range	-55.00-85.00 (Temperature Unit set to ℃); -67.00-185.00 (Temperature Unit set to ℉)	Default: N/A
Parameter Save	N/A	

Note: Min temperature value (since power on)。The byte order is configured by “Floating Point Register Byte Order, FLOATBYTEORDER” register.

TEMP_MAXPOWERON：Max temperature value (since power on)，INT16		
Range	-5500-8500 mapping to -55.00-85.00 (Temperature Unit set to ℃); -6700-18500 mapping to -67.00-185.00 (Temperature Unit set to ℉)	Default: N/A
Parameter Save	N/A	

Note: Max temperature value (since power on)。

TEMP_MAXPOWERON _FLOAT：Max temperature value (since power on)，FLOAT		
Range	-55.00-85.00 (Temperature Unit set to ℃); -67.00-185.00 (Temperature Unit set to ℉)	Default: N/A
Parameter Save	N/A	

Note: Max temperature value (since power on)。The byte order is configured by “Floating Point Register Byte Order, FLOATBYTEORDER” register.

TEMPUNIT: Temperature Unit		
Range	0: ℃ 1: ℉	Default: 0
Parameter Save	YES	

Note: Temperature Unit.

TOFFSET: Temperature Offset, INT16		
Range	-1000～1000 mapping to -10.00～10.00℃	Default: 0
Parameter Save	YES	

Note: Temperature Offset.

TEMPRATURE_CALIBED = TEMPRATURE + TOFFSET;

TEMPRATURE_CALIBED_FLOAT = TEMPRATURE_FLOAT + TOFFSET / 100.00;

ASYNCACQINTERVAL：Auto measurement interval，UINT16		
Range	0-43200（Seconds）	Default: 1
Parameter Save	YES	

Note: When set to 0, the sensor will only take measurement samples when the user reads the data. When set to a non-zero value, the sensor automatically measures at the set time interval, and immediately returns the latest measurement data after receiving the data reading instruction.

FLOATBYTEORDER：Floating Point Register Byte Order, INT16		
Range	0: Big-endian [ABCD] 1: Little-endian[DCBA] 2: Big-endian byte swap [BADC] 3: Little-endian byte swap[CDAB]	Default: 3
Parameter Save	YES	

Note: Configure floating point register byte order.

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]

REGCLEARCONFIG：Reset method of Max/Min temperature value (user resettable)， UINT16		
Range	0：Automatically reset after reading the corresponding register; 1：Reset by Modbus command (Refer to REGCLEAR register)	Default: 0
Parameter Save	YES	

Note: Reset method of Max/Min temperature value (user resettable).

REGCLEAR：Reset Max/Min temperature value (user resettable)， UINT16		
Range	Always 0 when read; After writing 0xFFFF, the user-resettable min/max temperature values are reset. Including registers below: TEMP_MINRESETTABLE，TEMP_MAXRESETTABLE，TEMP_MINRESETTABLE_FLOAT，TEMP_MAXRESETTABLE_FLOAT	Default: N/A
Parameter Save	N/A	

Note: Reset Max/Min temperature value (user resettable)

FACTORYRESET：Reset to factory configuration，UINT16		
Range	Always 0 when read; After writing 0xFFFF, the following registers are restored to factory Settings: TEMPUNIT，TOFFSET，ASYNCACQINTERVAL，FLOATBYTEORDER，REGCLEARCONFIG	Default: N/A
Parameter Save	N/A	

Note: Reset to factory configuration

SYSTEMRESET：Restart the sensor，UINT16		
Range	Always 0 when read; After writing 0xFFFF, the sensor will restart.	Default: N/A
Parameter Save	N/A	

Note: Restart the sensor

SLAVEADDRESSRESET：Reset the slave Modbus address to the factory default，UINT16		
Range	Always 0 when read; After writing 0xFFFF, the Modbus slave address is reset to the factory value。	Default: N/A
Parameter Save	N/A	

Note: Reset the slave Modbus address to the factory default

SLAVEADDRESS: Modbus Slave Address, UINT16		
Data Range	0-255	Default: 1
Parameter Storage	YES	

Note: Please power up the sensor to take effective after set.

BAUDRATE: Serial Comm Baudrate, UINT16		
Data Range	0-5 0:1200bps 1:2400bps 2:4800bps 3:9600bps 4:19200bps 5:38400bps	Default: 3
Parameter Storage	YES	

Note: Please power up the sensor to take effective after set.

PROTOCOL: Serial Comm Protocol, UINT16		
Data Range	0 0:Modbus RTU	Default: 0
Parameter Storage	YES	

Note: Please power up the sensor to take effective after set.

PARITY: Serial Comm Parity, UINT16		
Data Range	0-2 0:NONE 1:EVEN 2:ODD	Default: 0
Parameter Storage	YES	

Note: Please power up the sensor to take effective after set.

DATABITS: Serial Comm Databits, UINT16		
Data Range	1 1:8 databits	Default: 1

Parameter Storage	YES	

Note: Please power up the sensor to take effective after set.

STOPBITS: Serial Comm Stopbits, UINT16		
Data Range	0-1 0:1 stopbit 1:2 stopbits	Default: 0
Parameter Storage	YES	

Note: Please power up the sensor to take effective after set.

USER_SN--- User Serial Number		
Data Range	0x0000000000000000-0xFFFFFFFFFFFFFFFF User Serial Number. These four registers should be read and write together from start address 0x0220 with quantity 4.	Default: N/A
Parameter Storage	YES	

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.

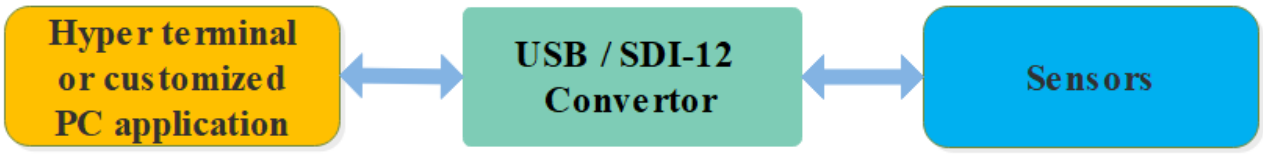
This chapter mainly introduces the communication or parameter setting on a computer for sensor through the SDI-12 converter (SDI12ELF20).

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 debugging tool)	https://www.infwin.com/resource-serial-port-com-development-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.

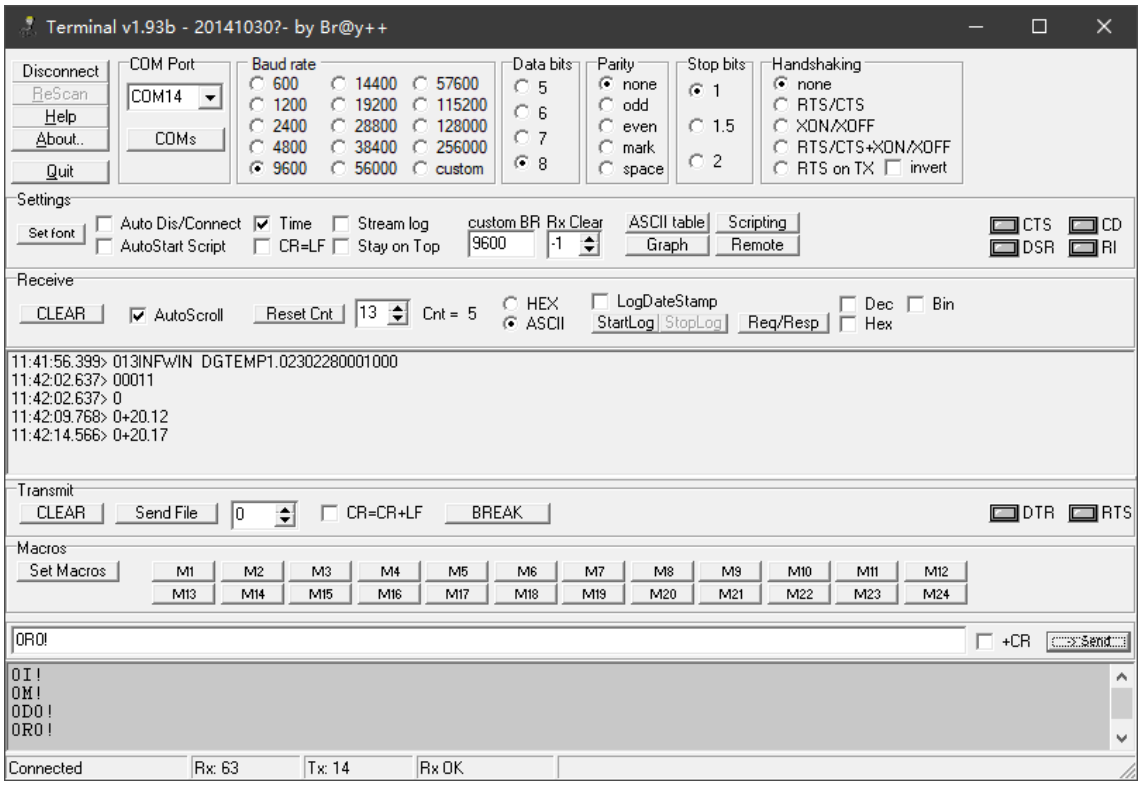
DigiTEMP Rugged Temperature Sensor User Manual
https://www.infwin.com/digitemp-rugged-digital-temperature-sensor-sdi12-rs485-modbus/

■ **Connections**

■ **Testing with serial port utility "Terminal"**

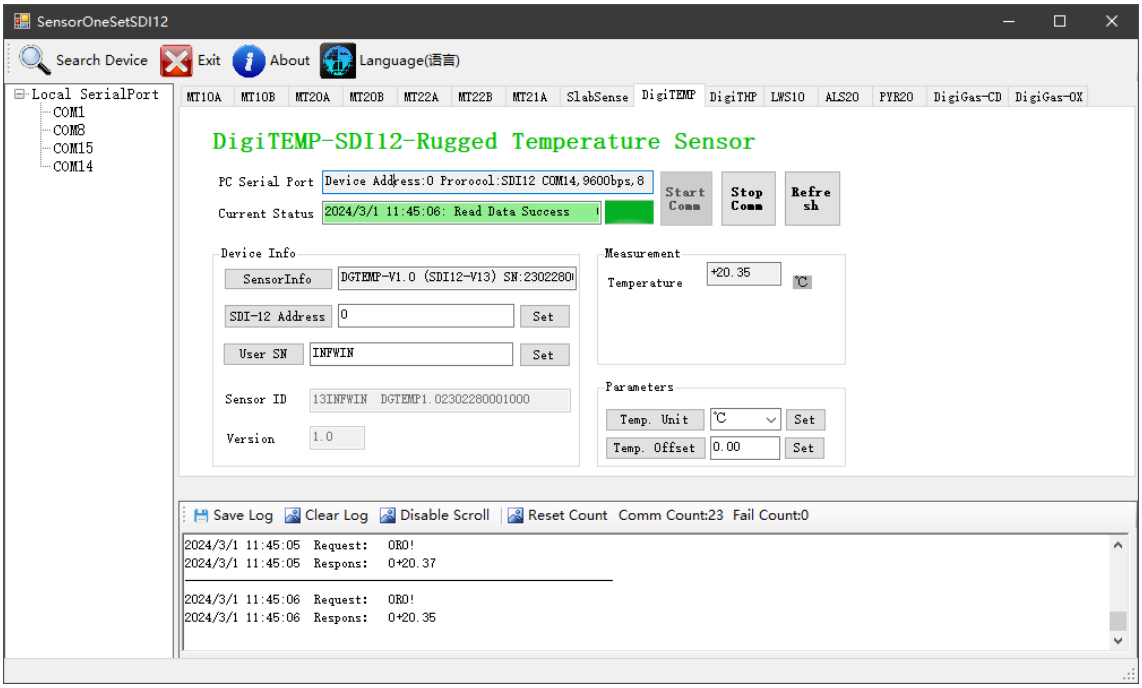
Take "Terminal" as an example, when debugging, please select the corresponding serial port number, baudrate is set to 9600bps, none parity, 8 data bits, 1 stop bit (the default communication Settings of SDI12ELF20), open the serial port and input the SDI-12 command and send. Please note

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

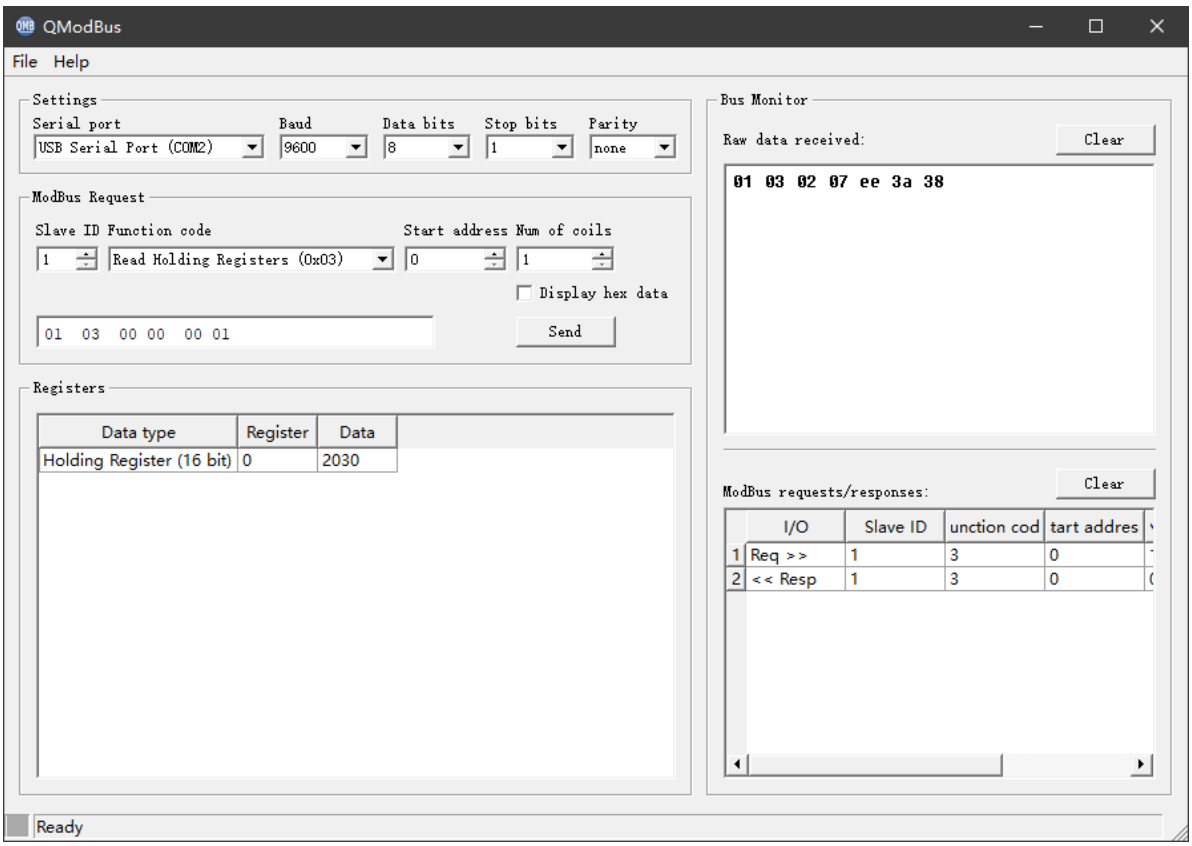
In this example, we use the USB/RS485 converter to communicate between a computer and the rugged temperature sensor DigiTEMP, The power supply of DigiTEMP is also provided by USB/RS485 converter.

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

Connections

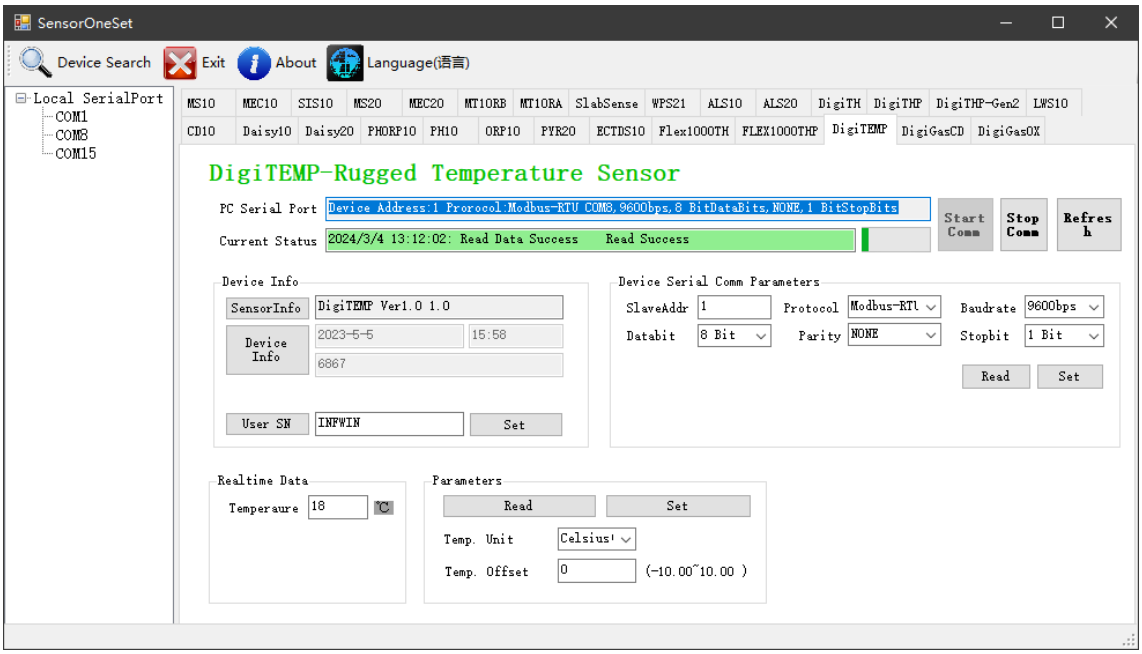
Testing with serial port utility “QModbus”

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.



Copyright and Trademark

Version Control

Date	Version	Comment	Updated by
2024-10-10	V1.0	Initial Creation	sl51930