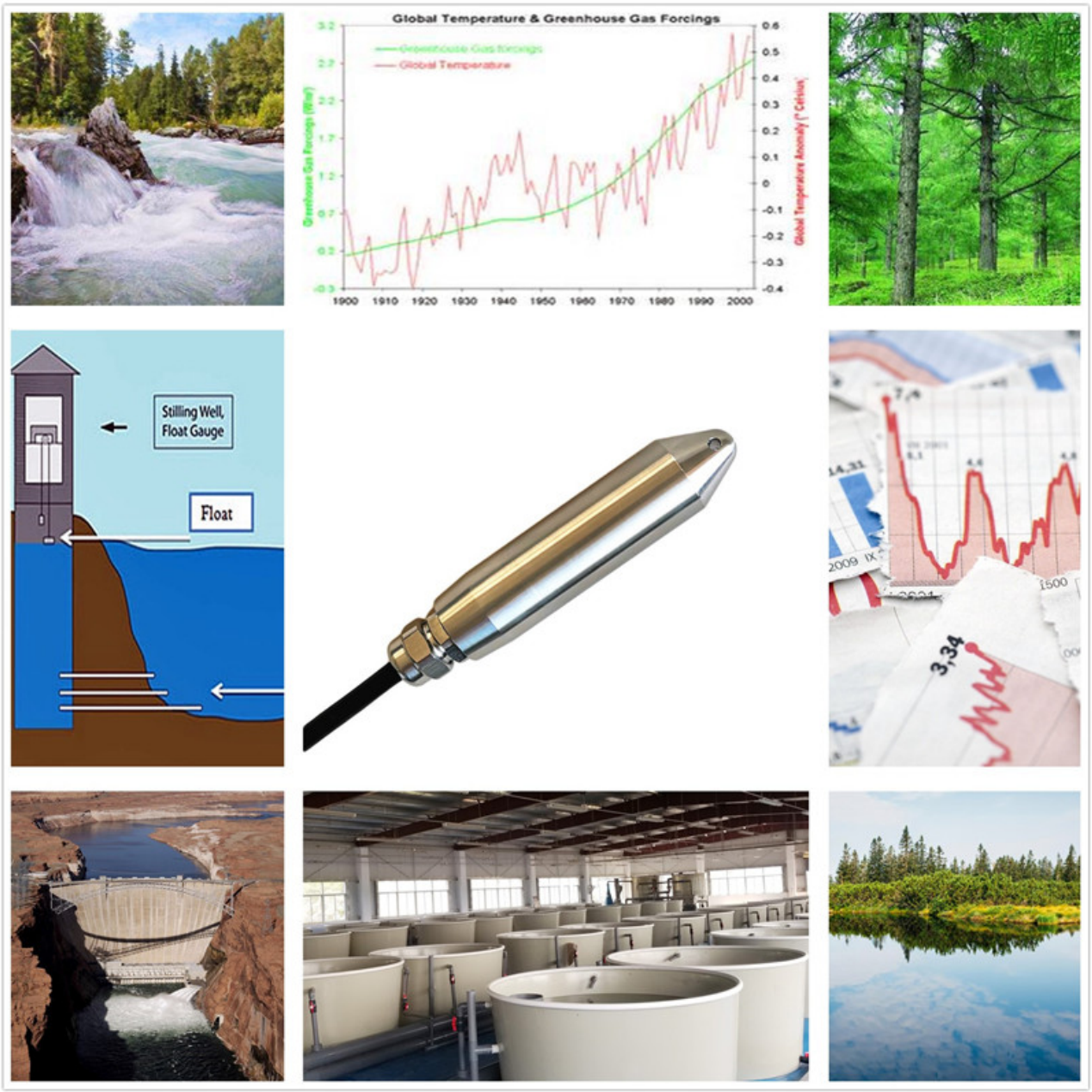
DigiTEMP

Rugged Digital Temperature Sensor (SDI-12 Interface)
Rugged Digital Temperature Sensor (RS485 Interface)
User Manual



Index

1 Customer Support 3

2 Introduction 4

3 Wiring diagrams 6

3.1 SDI-12 Interface 6
3.2 RS485 Interface 7

4 Dimension and Ordering Infomation 8

4.1 Dimension 8
4.2 Ordering Information 8

5 Safty ,Care and Installation 9

5.1 Care and Safty 9
5.2 Installation 9

6 SDI-12 Communication 10

6.1 SDI-12 Interface and Protocol 10
6.2.1 SDI-12 Interface 10
6.2.2 Protocol 10

7 RS485 Communication 14

7.1 Modbus Protocol 14
7.2 Modbus Register 14
7.3 Modbus Register Detail Descripton 15
7.4 Modbus Function Code 17
7.4.1 Function Code 3 Protocol Example 17
7.4.2 Function Code 4 Protocol Example 19
7.4.3 Function Code 6 Protocol Example 20
7.4.4 Function Code 16 Protocol Example 21
7.5 Software Configuration Utility 22
7.5.1 Universal Modbus Comm Utility 22
7.5.2 SensorOneSet Configuration Utility 22

Appendix 24

Copyright and Trademark 24
Version Control 24

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

Fax

+86-411-66831953

2 Introduction

The DigiTEMP is a high precision digital temperature sensor ideal for high-accuracy readings in water, soil, and air. It features fully-potted components, robust stainless steel housing, making the sensor ideal for harsh environments. The sensor is small enough to easily deploy through standard 1” (2.5cm) PVC conduit with 8” (20.3cm) factory bend corners. It also offers a loop hole which can be used to mount weights or pull the sensor through pipes or other small areas.

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 with high precision temperature sensor
- Digital Output SDI-12 or RS485 with built-in surge protection
- Tough UV resistant polyurethane cable with waterblock
- Stainless steel body with eyelet for weight or pull strings
- Rugged housing and fully potted electronics - no risk of leaking
- Low power consumption
- Reverse power protection and Built-in TVS/ESD protection
- ODM/OEM Service

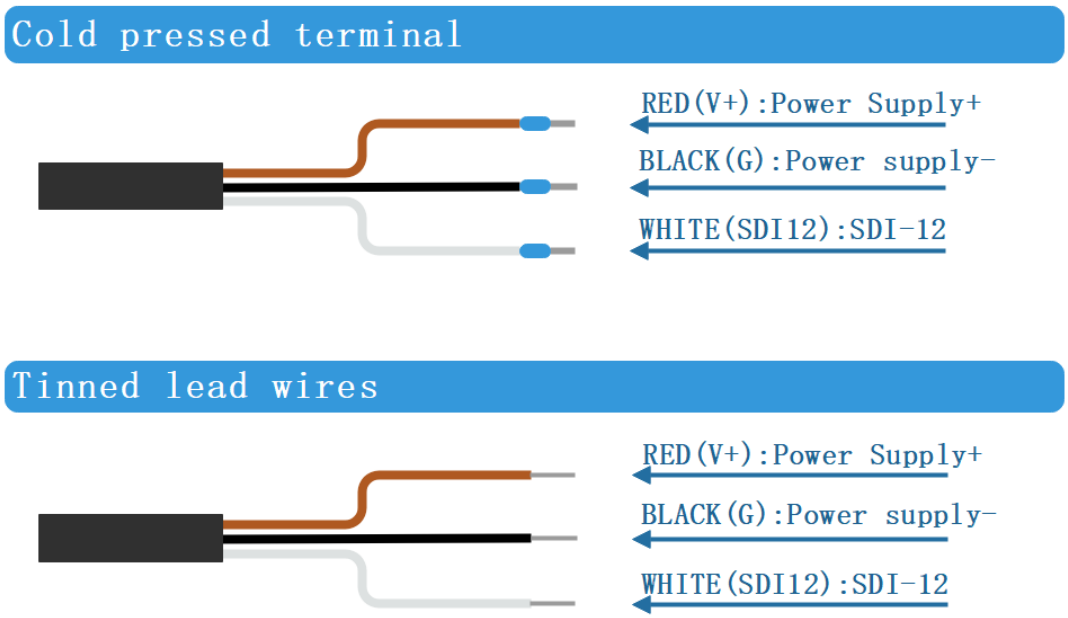
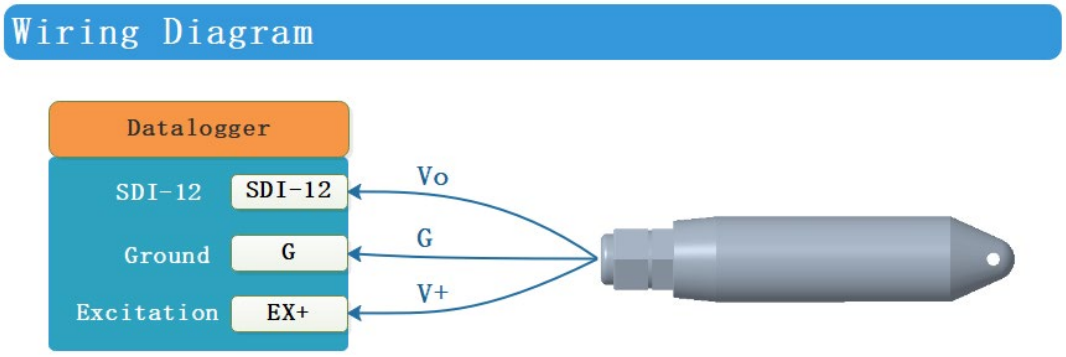
Applications

- Soil Temperature
- Streams
- Surface Water
- Stilling Wells
- Dams
- Aquaculture Tanks

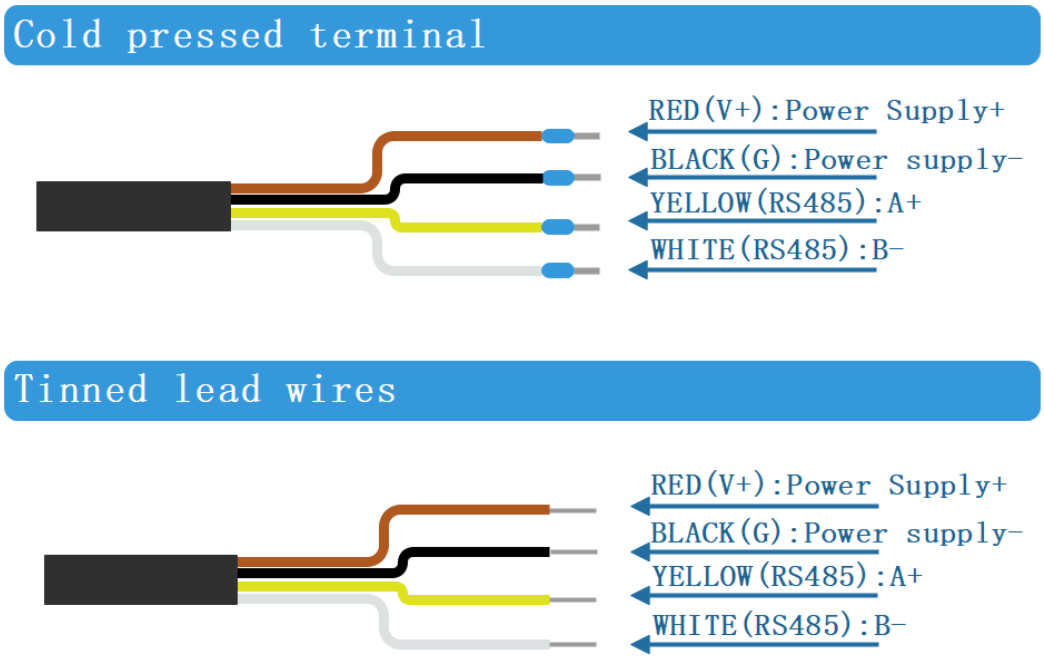
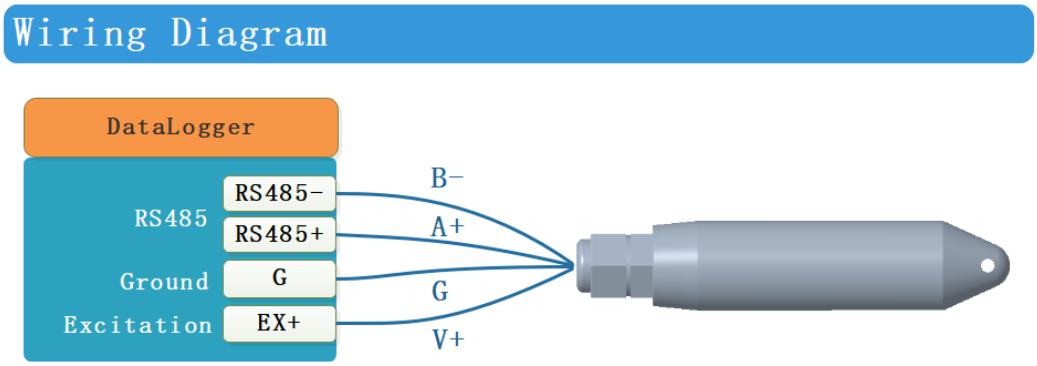
Specifications	
Output Interface	Optional: SDI-12, V1.3 Optional: RS485, Modbus-RTU
Power Supply	4.5-18V/DC
Power Consumption	SDI-12 Interface: Quiescent Current : <10uA RS485 Interface: Quiescent Current : <300uA Measuring Current : 10mA during 50ms measurement
Temperature Measurement	Range: 0-70°C, Resolution:0.01°C, Accuracy:+/-0.2°C Range: -40-80°C, Resolution:0.01°C, Accuracy:+/-0.3°C Range: -40-125°C, Resolution:0.01°C, Accuracy:+/-0.5°C **Attention: The long term operating temperature is -40~80°C**
Operating Temperature	-40~80°C
IP Ratings	IP68
Sensor Sealed	Epoxy resin
Installation	Immersed, Surface or buried installation
Cable Length	5 meters or Customize
Dimension	95*20mm (Length*Diameter)

3 Wiring diagrams

3.1 SDI-12 Interface

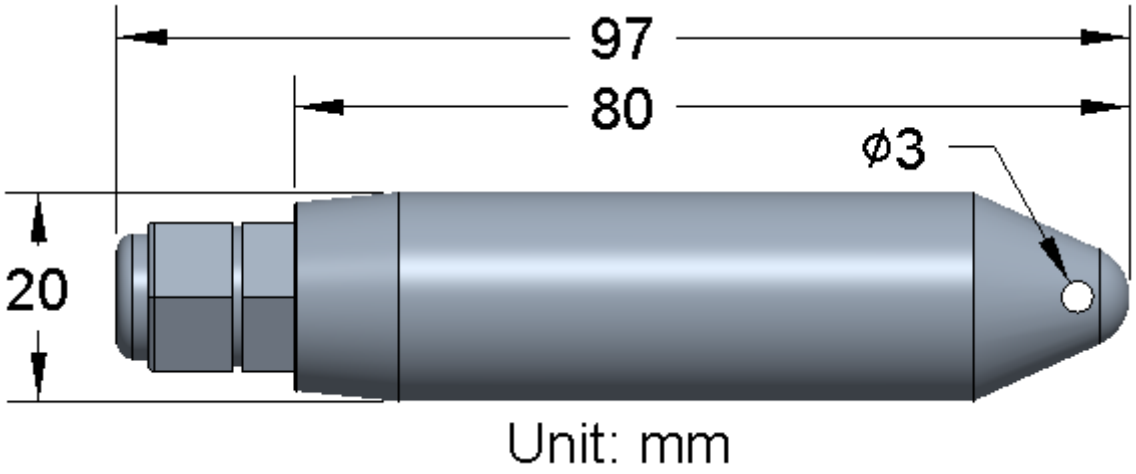
Type	Wiring diagram
SDI-12 Interface	
Connections	

3.2 RS485 Interface

Type	Wiring diagram
RS485 Interface	
Connections	

4 Dimension and Ordering Infomation

4.1 Dimension



Unit: mm

4.2 Ordering Information

Parameters	Code	Comments
Code 1: Product Series	DigiTEMP	Rugged digital temperature sensor
Code 2: Output Interface	A B	SDI-12 RS485 (Modbus-RTU)
Code 3: Power Supply	A B	4.5-18V DC Customize
Code 4: Connector	B C	Cold pressed terminal Stripped & tinned lead wires
Code 5: Cable Length	005 XXX	5 meters Customize, XXX is required cable length(Unit: meter)
Ordering Code Example: DigiTEMP Rugged digital temperature sensor, Output interface SDI-12, Power supply 4.5-18V DC, Cold pressed terminal, Cable length 5 meters. Ordering Code is : DigiTEMP-AAB005		

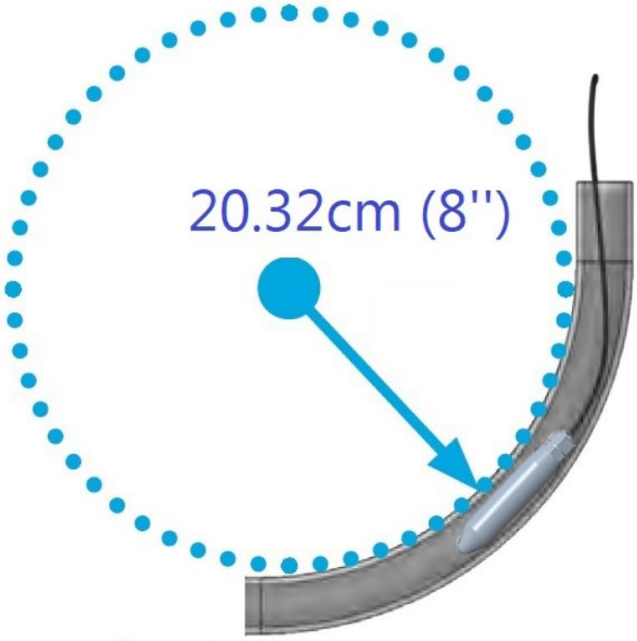
5 Safty ,Care and Installation

5.1 Care and Safty

- Do not pull the sensor out of the soil by its cable.
- Do not pull the sensor if you feel any resistance when pulling the sensor out of the conduit.

5.2 Installation

This sensor is ideal for high-accuracy readings in water, soil, and air. It features fully-potted components, robust stainless steel housing, making the sensor ideal for harsh environments. The sensor is small enough to easily deploy through standard 1” (2.5cm) PVC conduit with 8” (20.3cm) factory bend corners. It also offers a loop hole which can be used to mount weights or pull the sensor through pipes or other small areas.



6 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	s	Maximum measurement time (fixed width of 3)
<TAB>	-	Tab character
<SPACE>	-	Space character
<CR>	-	Carriage return character
<LF>	-	Line feed character
<temperature>	-	Temperature
<CRC>	-	SDI-12 protocol CRC Checksum

6.1 SDI-12 Interface and Protocol

6.2.1 SDI-12 Interface

Please refer to SDI-12 standard user manual V1.3.

6.2.2 Protocol

Request	Response	Comment
a!	a<CR><LF>	**Acknowledge Active** a: Sensor address **Example:** Request: 0! Response: 0<CR><LF>
aI!	allccccccccmmmmmmmvvvxxxxxxxxxxxxxx<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 DGTEMP1.01909250001000<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!	atttn<CR><LF> a:Sensor address ttt: Measurement data will be ready in ttt seconds n:Number of measurement data <CR><LF>:terminates the response	**Temperature Measurement** **Example:** Start Measurement Command. 1 data will be ready in 001 seconds. Request: 0M! Response: 00011<CR><LF> Response: 0<CR><LF> Request: 0D0! Response: 0+23.80<CR><LF> <temperature>=+23.80
aMC!	atttn<CR><LF> a:Sensor address ttt: Measurement data will be ready in ttt seconds n:Number of measurement data <CR><LF>:terminates the response	**Temperature Measurement with CRC** **Example:** Start Measurement and Request CRC. 1 data will be ready in 001 seconds. Request: 0MC! Response: 00011<CR><LF> Response: 0<CR><LF> Request: 0D0! Response: 0+23.8A]p<CR><LF>
aC!	atttn<CR><LF> a:Sensor address ttt: Measurement data will be ready in ttt seconds	**Temperature Concurrent Measurement** **Example:** Start Measurement Command. 1 data will be ready in 001 seconds. Request: 0M!

	n:Number of measurement data <CR><LF>:terminates the response	Response: 00011<CR><LF> Request: 0D0! Response: 0+23.80<CR><LF> <temperature>=+23.80
aCC!	atttn<CR><LF> a:Sensor address ttt: Measurement data will be ready in ttt seconds n:Number of measurement data <CR><LF>:terminates the response	**Temperature Concurrent Measurement with CRC** **Example:** Start Measurement and Request CRC. 1 data will be ready in 001 seconds. Request: 0MC! Response: 00011<CR><LF> Request: 0D0! Response: 0+23.8A]p<CR><LF>
aV!	atttn<CR><LF> a:Sensor address ttt: Measurement data will be ready in ttt seconds n:Number of measurement data <CR><LF>:terminates the response	**Sensor Verification Command** **Example:** Start Verification. 1 data will be ready in 001 seconds. Request: 0V! Response: 00011<CR><LF> Response: 0<CR><LF> Request: 0D0! Response: 0+0<CR><LF>, "+0" indicate sensor normal, "+1" means sensor error.
aD0!	a[<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!	a<svvvv><CR><LF> <svvvv>:<temperature>	**Temperature Continuous Measurements, and return data** **Example:** Request: 0R0! Response: 0+23.8<CR><LF>
aRC0!	a<svvvv><CRC><CR><LF> <svvvv>:<temperature> <CRC>:CRC checksum	**Temperature Continuous Measurements and Request CRC, and return data.** **Example:** Request: 0RC0! Response: 0+23.8A]p<CR><LF>
aXR_TUNIT!	aTUNIT=<X> <X> is temperature unit: C: degrees centigrade F: degrees fahrenheit K: degrees kelvin	**Query temperature unit** **Example:** Request: 0XR_TUNIT! Response: 0TUNIT=C<CR><LF>
aXW_TUNIT_<X>!	aTUNIT=<X>	**Configure temperature unit** **Example:** Request: 0XW_TUNIT_C!

		Response: 0TUNIT=C<CR><LF>
aXR_TOFFSET!	aTOFFSET=<svvvv> <svvvv>: temperature offset value between -10.00~10.00, it will be effective when issuing a new measurement command. The temperature display value equals to the original sensor measurement value added with the temperature offset value.	**Query temperature offset value** **Example:** Request: 0XR_TOFFSET! Response: 0TOFFSET=+1.00<CR><LF>
aXW_TOFFSET_<saaaa>!	aTOFFSET=<svvvv>	**Configure temperature offset value** **Example:** Request: 0XW_TOFFSET_+1.00! Response: 0TOFFSET=+1.00<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>

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 effective after re-power up the sensor.

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 Addr. (HEX/DEC)	Data Type	Modbus Function Code(DEC)	Range and Comments	Default Value
TEMPRATURE	0x0000 /0	INT16 RO	3/4	-4000-12500 for -40.00~1250.00°C.	N/A
RESERVED	0x0001 /1	UINT16 RO	3/4	Reserved	0
RESERVED	0x0002 /2	UINT16 RO	3/4	Reserved	0
RESERVED	0x0003 /3	UINT16 RO	3/4	Reserved	0
RESERVED	0x0004 /4	UINT16 RO	3/4	Reserved	0
RESERVED	0x0005 /5	UINT16 RO	3/4	Reserved	0
TEMPUNIT	0x0020 /32	UINT16 R/W	3/6/16	0:°C 1:°F	0

				2:K	
TEMPCALIB	0x0021 /33	INT16 R/W	3/6/16	-1000-1000 for -10.00~10.00	0
SLAVEADDRESS	0x0200 /512	UINT16 R/W	3/6/16	0-255	1
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
PROTOCOL	0x0202 /514	UINT16 R/W	3/6/16	0 0:Modbus RTU	0:Modbus RTU
PARITY	0x0203 /515	UINT16 R/W	3/6/16	0-2 0:None 1:Even 2:Odd	0:None Parity
DATABITS	0x0204 /516	UINT16 R/W	3/6/16	1 1:8 databits	1:8 databits
STOPBITS	0x0205 /517	UINT16 R/W	3/6/16	0-1 0:1 stopbit 1:2 stopbits	0:1 stopbit
RESERVED	0x0206 /518	UINT16 R/W	3/6/16	Reserved	0
RESERVED	0x0207 /519	UINT16 R/W	3/6/16	Reserved	0

NOTE: UINT16:16 bit unsigned integer, INT16:16bit signed integer

NOTE: RO: Register is Read Only, R/W: Register is Read/Write

NOTE: HEX is Hexadecimal (data with 0x/0X prefix), DEC is Decimal

7.3 Modbus Register Detail Descripton

TEMPERATURE		
Data Range	-4000-12500 For -40.00~125.00°C **Attention: The long term operating temperature is -40~80°C**	Default: N/A
Power Down Save	N/A	

Note:Temperature value (Binary complement).

Example:When REGISTER = 0x0702 (HEX format), then VALUE=(0x07*256+0x02)/100=17.94°C.When REGISTER=FF05H (HEX format),then VALUE=((0xFF*256+0x05)-0xFFFF-0x01)/100 =(0xFF05-0xFFFF-0x01)/100=-2.51°C.

TEMPUNIT		
Data Range	0:°C 1:°F 2:K	Default: 0
Power Down Save	YES	

Note: Temperature Unit

TEMPOFFSET		
Data Range	-1000-1000 for -10.00~10.00°C	Default: 0
Power Down Save	YES	

Note: Temperature Offset Value, When sensor temperature measured is 21℃/70℉/295K, and temperature offset set to 10.00, then the temperature register(TEMPERATURE) will be 21℃/70℉/295K+10.00=31℃/80℉/305K。

SLAVEADDRESS --- Modbus Slave Address		
Data Range	0-255	Default: 1
Power Down Save	YES	

Note: Please re-power on the sensor to take effective after set.

BAUDRATE --- Serial Comm Baudrate		
Data Range	0-5 0:1200bps 1:2400bps 2:4800bps 3:9600bps 4:19200bps 5:38400bps	Default: 3
Power Down Save	YES	

Note: Please re-power on the sensor to take effective after set.

PROTOCOL --- Serial Comm Protocol

Data Range	0 0:Modbus RTU	Default: 0
Power Down Save	YES	

Note: Please re-power on the sensor to take effective after set.

PARITY --- Serial Comm Parity		
Data Range	0-2 0:NONE 1:EVEN 2:ODD	Default: 0
Power Down Save	YES	

Note: Please re-power on the sensor to take effective after set.

DATABITS --- Serial Comm Databits		
Data Range	1 1:8 databits	Default: 1
Power Down Save	YES	

Note: Please re-power on the sensor to take effective after set.

STOPBITS --- Serial Comm Stopbits		
Data Range	0-1 0:1 stopbit 1:2 stopbits	Default: 0
Power Down Save	YES	

Note: Please re-power on the sensor to take effective after set.

7.4 Modbus Function Code

For description below, data started with 0X/0x means that it's in HEX format.

7.4.1 Function Code 3 Protocol Example

Master Request:AA 03 RRRR NNNN CCCC

AA	1 byte	Slave Address,0-255

0x03	1 byte	Function Code 3
RRRR	2 byte	Starting Register Addr
NNNN	2 byte	Quantity of Register to read
CCCC	2 byte	CRC CHECKSUM

Slave Response:AA 03 MM VV0 VV1 VV2 VV3… CCCC

AA	1 byte	Slave Address,0-255
0x03	1 byte	Function Code 3
MM	1 byte	Register Data Byte Count
VV0,VV1	2 byte	Register Value (High8bits first)
VV2,VV3	2 byte	Register Value (High8bits first)
…	…	Register Value (High8bits first)
CCCC	2 byte	CRC CHECKSUM

Example:Read register 0x0200-0x0201,that is slave address and baudrate.

Master Request:01 03 0200 0002 C5B3

Slave Addr.	1 byte	0x01
Function Code	1 byte	0x03
Starting Register Addr.	2 byte	0x0200
Quantity of Register to read	2 byte	0x0002
Checksum	2 byte	0xC5B3

Slave Response:01 03 04 00 01 00 03 EB F2

Slave Addr.	1 byte	0x01
Function Code	1 byte	0x03
Register Data Byte Count	1 byte	0x04
Register Value: Address	2 byte	0x00(HIGH 8 Bits)
		0x01(LOW8 Bits)
Register Value: Baudrate	2 byte	0x00(HIGH 8 Bits)
		0x03(LOW8 Bits)
Checksum	2 byte	0xEBF2

7.4.2 Function Code 4 Protocol Example

Master Request:AA 04 RRRR NNNN CCCC

AA	1 byte	Slave Address,0-255
0x04	1 byte	Function Code 4
RRRR	2 byte	Starting Register Addr
NNNN	2 byte	Quantity of Register to read
CCCC	2 byte	CRC CHECKSUM

Slave Response:AA 04 MM VV0 VV1 VV2 VV3… CCCC

AA	1 byte	Slave Address,0-255
0x04	1 byte	Function Code 4
MM	1 byte	Register Data Byte Count
VV0,VV1	2 byte	Register Value (High8bits first)
VV2,VV3	2 byte	Register Value (High8bits first)
…	…	Register Value (High8bits first)
CCCC	2 byte	CRC CHECKSUM

Example:Read register 0x0000-0x0002,that is temperature, reserved, reserved.

Master Request:01 04 0000 0003 B00B

Slave Addr.	1 byte	0x01
Function Code	1 byte	0x04
Starting Register Addr.	2 byte	0x0000
Quantity of Register to read	2 byte	0x0003
Checksum	2 byte	0xB00B

Slave Response: 01 04 06 08 54 00 00 00 00 50 17

Slave Addr.	1 byte	0x01
Function Code	1 byte	0x04
Register Data Byte Count	1 byte	0x04
Register Value: Temperature	2 byte	0x08(HIGH 8 Bits)
		0x54(LOW8 Bits)

Register Value:	2 byte	0x00(HIGH 8 Bits)
Reserved		0x00(LOW8 Bits)
Register Value:	2 byte	0x00(HIGH 8 Bits)
Reserved		0x00(LOW8 Bits)
Checksum	2 byte	0x5017

7.4.3 Function Code 6 Protocol Example

Master Request:AA 06 RRRR VVVV CCCC

AA	1 byte	Slave Address,0-255
0x06	1 byte	Function Code 6
RRRR	2 byte	Register Addr (High8bits first)
VVVV	2 byte	Register Value (High8bits first)
CCCC	2 byte	CRC CHECKSUM

Slave Response:AA 06 RRRR VVVV CCCC

AA	1 byte	Slave Address,0-255
0x06	1 byte	Function Code 6
RRRR	2 byte	Register Addr (High8bits first)
VVVV	2 byte	Register Value (High8bits first)
CCCC	2 byte	CRC CHECKSUM

Example:Write Register 0x0200,that is change modbus slave address to 2.

Master Request: 01 06 0200 0002 09B3

Slave Addr.	1 byte	0x01
Function Code	1 byte	0x06
Register Addr.	2 byte	0x0200
Register Value	2 byte	0x0002
Checksum	2 byte	0x09B3

Slave Response: 01 06 0200 0002 09B3

Slave Addr.	1 byte	0x01
Function Code	1 byte	0x06
Register Addr.	2 byte	0x0200
Register Value	2 byte	0x0002

Checksum	2 byte	0x09B3

7.4.4 Function Code 16 Protocol Example

Master Request:AA 10 RRRR NNNN MM VVVV1 VVVV2 …CCCC

AA	1 byte	Slave Address,0-255
0x10	1 byte	Function Code 0x10
RRRR	2 byte	Starting Register Addr
NNNN	2 byte	Quantity of Register to write
MM	1 byte	Register Data Byte Count
VVVV1	2 byte	Register Value(High8bits first)
VVVV2	2 byte	Register Value(High8bits first)
…	…	Register Value(High8bits first)
CCCC	2 byte	CRC CHECKSUM

Slave Response:AA 10 RRRR NNNN CCCC

AA	1 byte	Slave Address,0-255
0x10	1 byte	Function Code 0x10
RRRR	2 byte	Starting Register Addr
NNNN	2 byte	Quantity of Register to write
CCCC	2 byte	CRC CHECKSUM

Example:Write Register 0x0200-0x0201,that is set slave address to 1,and baudrate to 19200bp.

Master Request:01 10 0200 0002 04 0001 0004 BACC

0x01	1 byte	Slave Addr.
0x10(HEX)	1 byte	Function Code 0x10
0x0200	2 byte	Starting Register Addr
0x0002	2 byte	Quantity of Register to write
0x04	1 byte	Register Data Byte Count
0x0001	2 byte	Register Value: Slave Address 1
0x0004	2 byte	Register Value: Baudrate 19200bps
0xBACC	2 byte	CRC CHECKSUM

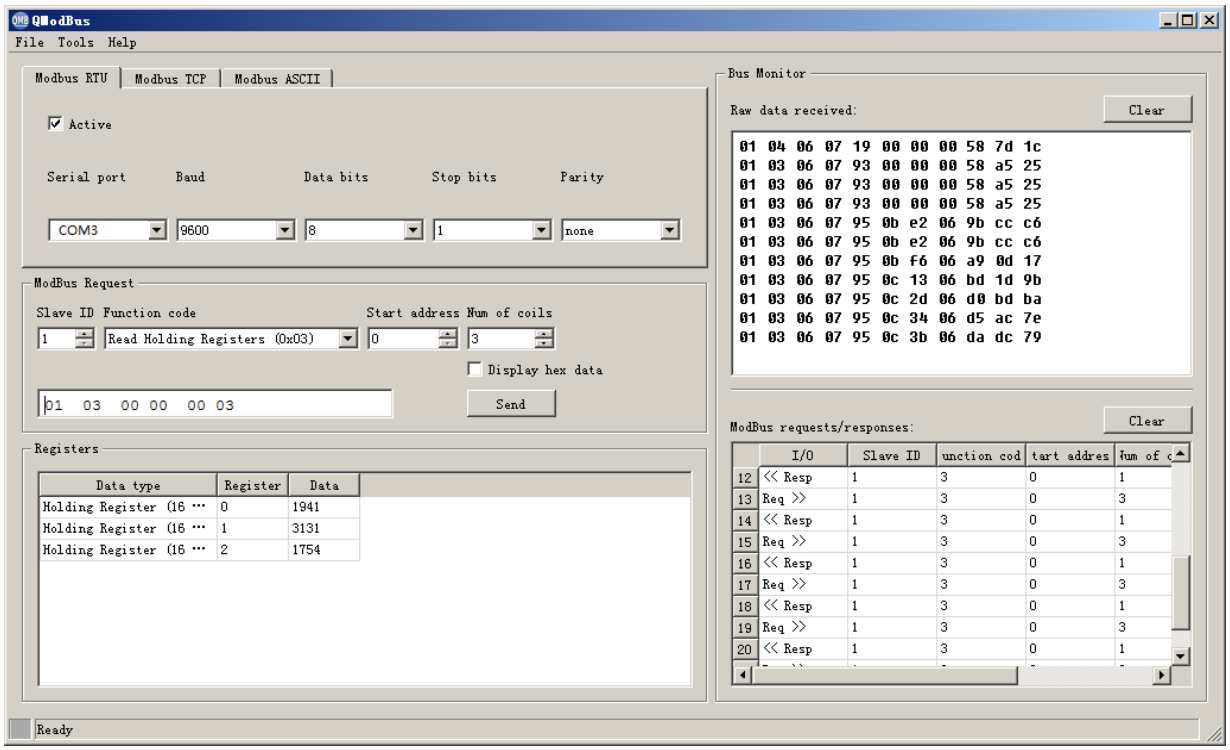
Salve Response:01 10 0200 0002 4070

0x01	1 byte	Slave Addr.
0x10(HEX)	1 byte	Function Code 0x10
0x0200	2 byte	Starting Register Addr(High8bits first)
0x0002	2 byte	Quantity of Register to write(High8bits first)
0x4070	2 byte	CRC CHECKSUM

7.5 Software Configuration Utility

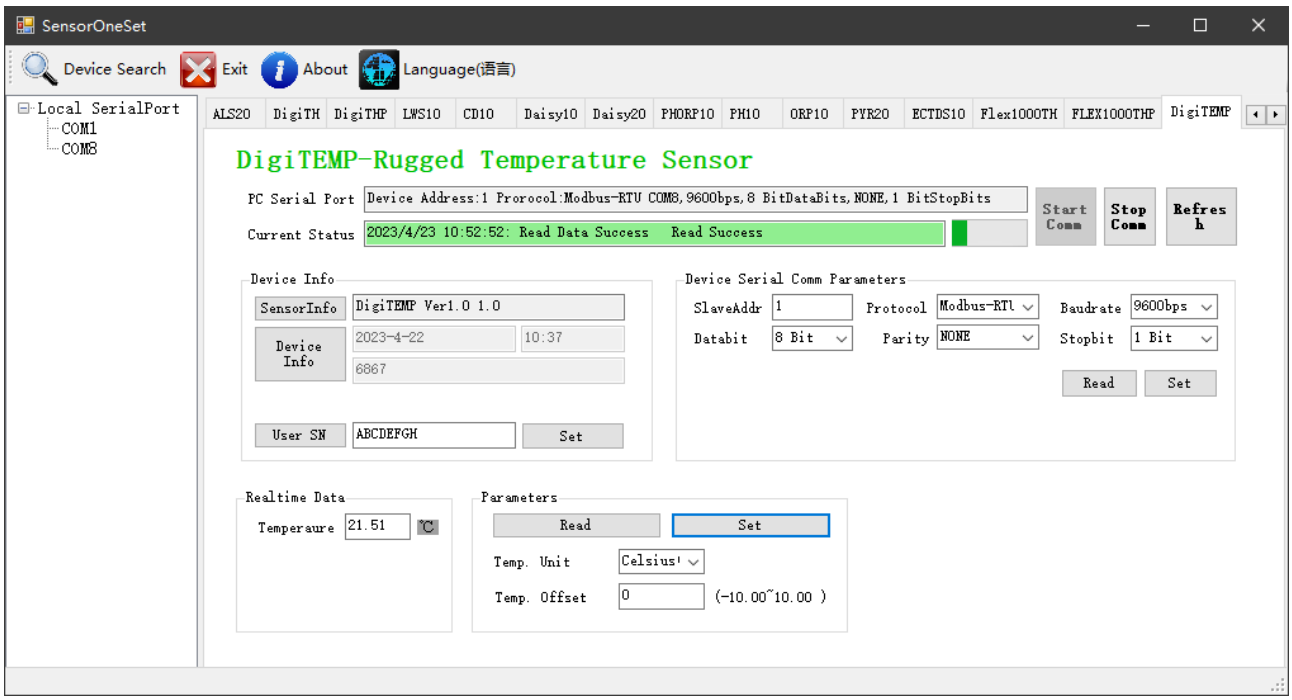
7.5.1 Universal Modbus Comm Utility

You can use software listed below to try reading/writing the register of sensor, https://github.com/ed-chemnitz/qmodbus/releases



7.5.2 SensorOneSet Configuration Utility

SensorOneSet is a configuration utility to read/set sensor config for all of our serial communication sensor products. Please download on our website: https://www.infwin.com


SensorOneSet
Device Search
Exit
About
Language(语言)
Local SerialPort
COM1
COM8
ALS20
DigiTH
DigiTHP
LWS10
CD10
Daisy10
Daisy20
PHORP10
PH10
ORP10
PYR20
ECTDS10
Flex1000TH
FLEX1000THP
DigiTEMP
DigiTEMP-Rugged Temperature Sensor
PC Serial Port
Device Address:1 Prorocol:Modbus-RTU COM8,9600bps,8 BitDataBits,NONE,1 BitStopBits
Current Status
2023/4/23 10:52:52: Read Data Success Read Success
Start Comm
Stop Comm
Refresh
Device Info
SensorInfo
DigiTEMP Ver1.0 1.0
Device Info
2023-4-22
10:37
6867
User SN
ABCDEFGH
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
21.51
℃
Parameters
Read
Set
Temp. Unit
Celsius
Temp. Offset
0
(-10.00~10.00)

Appendix

Copyright and Trademark

Version Control

Date	Version	Comment	Updated by
2023-04-23	V1.0	Initial Creation	sl51930