LWS10

Leaf Wetness Sensor (SDI-12 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

Fax

+86-411-66831953

2 Introduction

2.1 Brief

LWS10 measures leaf wetness. It sealed with resin packaged plastic body can be used for long time monitor. Wetness measurement value is proportional to the percentage of sensing area covered with water. The sensor is applicable for science research, bacteria prevention, foliage dressing spraying, greenhouse, smart agriculture etc.

The SDI-12 output provides universal compatibility with any SDI-12-enabled data logger and low power applications.

Features

- Integrated with leaf wetness and temperature measurement
- SDI-12 Output Interface with low power design
- Mimics the real leaf thermal resistance by sensing shape and thickness
- Water, ice, foggy detection
- Water proof to IP68 ratings
- High accuracy with excellent stability
- Reverse power protection and Built-in TVS/ESD protection

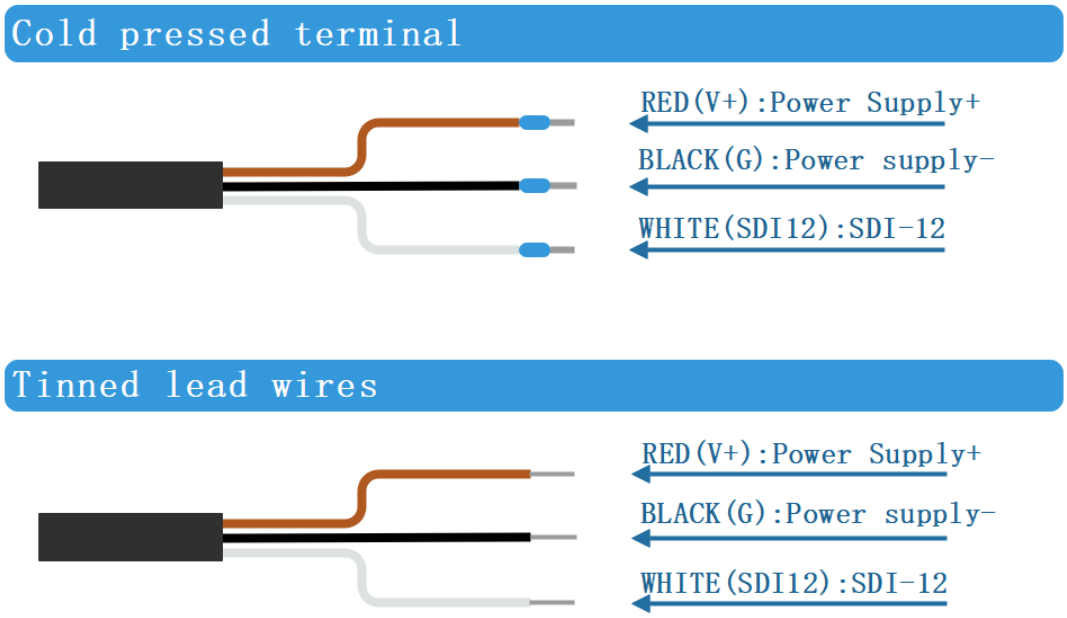
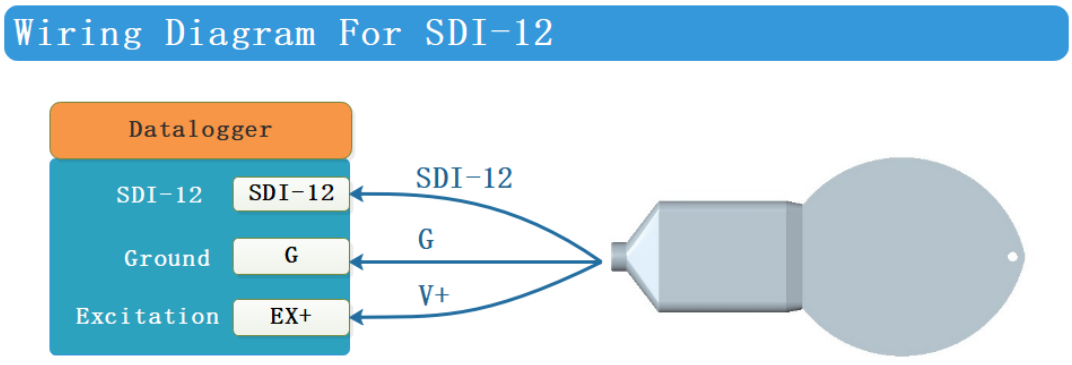
Applications

- Smart agriculture
- Greenhouse monitoring
- Science research
- Bacteria prevention
- Foliage dressing spraying

Specifications	
Output Interface	SDI-12, V1.3
Power Supply	4.5-28V/DC
Power Consumption	SDI-12 Interface: Quiescent Current : <10uA Measuring Current : 10mA during 50ms measurement
Temperature	Range:-40~80°C, Resolution:0.1°C, Accuracy: +/-0.5°C
Wetness	Range:0~100%, Resolution:0.1%, Accuracy: +/-5%
Operating	Temperature: -40~80°C, Humidity: 0-100%
IP Ratings	IP68
Cable Length	2 meters or Customize
Dimension	Sensor Body 65*13*145mm

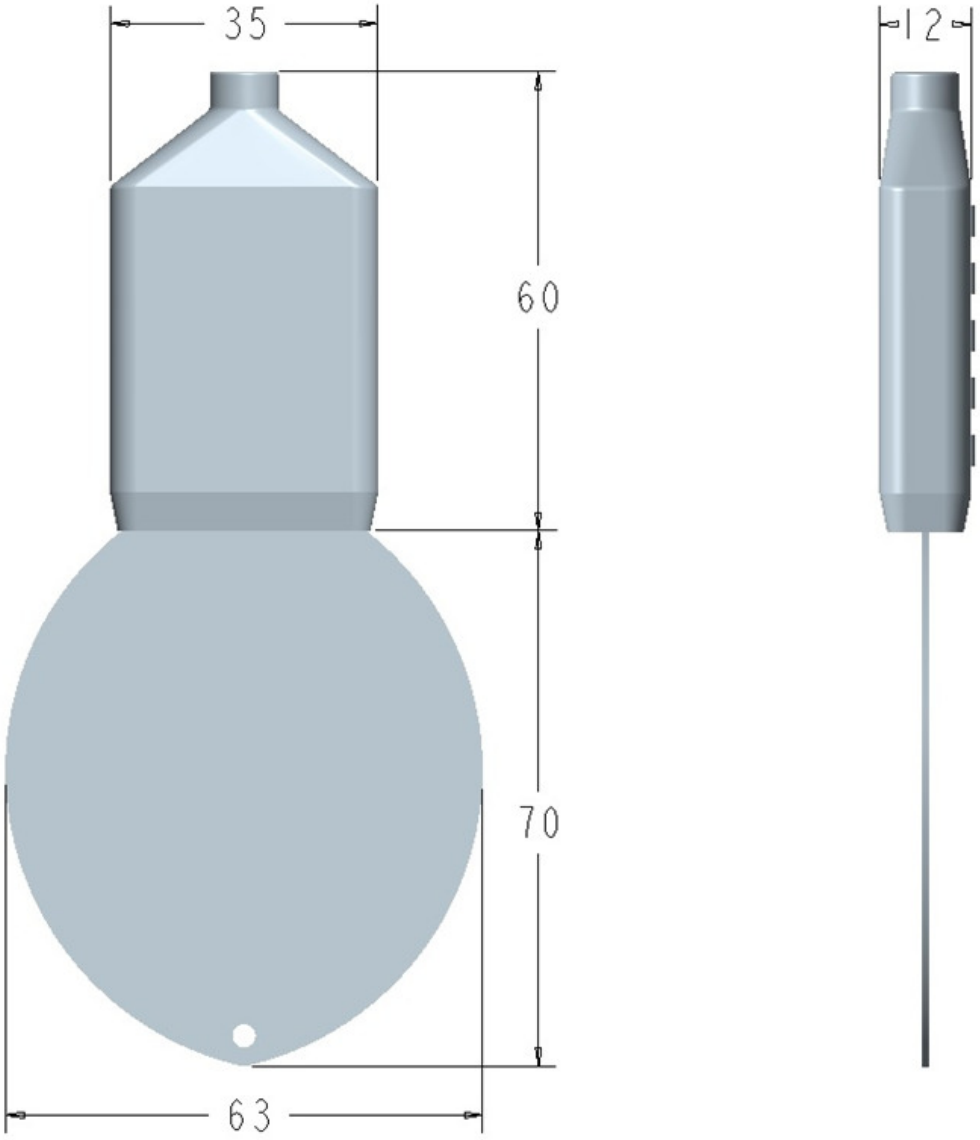
3 Wiring diagrams

3.1 SDI-12 Interface

Type	Wiring diagram
SDI-12 Interface	
Connections	

4 Dimension and Ordering Infomation

4.1 Dimension



Unit: mm

4.2 Ordering Information

Parameters	Code	Comments
Code 1: Product Series	LWS10	LWS10 leaf wetness and temperature Sensor
Code 2: Output Interface	F	SDI-12
Code 3: Power Supply	E	4.5-28V DC
Code 4: Cable Length	002 XXX	2 meters Customize, XXX is required cable length(Unit: meter)
Ordering Code Example: Leaf wetness and temperature sensor, Output Interface SDI-12, Power Supply 4.5-28V DC, Cable Length 2 meters. Ordering Code is : LWS10 – FE002		

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
<±temperature_calibed>	℃ ℉	Temperature Calibed, the value is output according to the temperature unit setting.
<±temperature>	℃ ℉	Temperature Original, the value is output according to the temperature unit setting.
<±TOFFSET>	℃ ℉	Temperature Offset, <±temperature_calibed>=<±temperature> + <±TOFFSET>
<+wetness>	%	Leaf wetness 0.00-100.00

5.1 SDI-12 Interface and Protocol

5.1.1 SDI-12 Interface

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

5.1.2 Protocol

Request	Response	Comment
a!	a<CR><LF>	**Acknowledge Active** a: Sensor address **Example:** Request: 0! Response: 0<CR><LF>
aI!	allccccccccmmmmmmvvvxxxxxxxxxxxxxx<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 xxxxxxxxxxxx: 13 characters serial number <CR><LF>: terminates the response **Example:** Request: 0I! Response: 013INFWIN DGTHP 2.02305170016000<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!	a0012<CR><LF> a:Sensor address 001: Measurement data will be ready in 001 seconds 2: Number of measurement data returned by aD0! <CR><LF>:terminates the response aD0! Response data format： a<±temperature_calibed><+wetness>[<CRC>]<CR><LF>	**Temperature Calibed, Wetness Measurement** **Example:** Request: 0M! Response: 00012<CR><LF> Response: 0<CR><LF> Request: 0D0! Response: 0+23.96+29.31<CR><LF>
aM1!，aMC1!	a0012<CR><LF> a:Sensor address 001: Measurement data will be ready in 001 seconds 2: Number of measurement data returned by aD0! <CR><LF>:terminates the response aD0! Response data format： a<±temperature><+wetness>[<CRC>]<CR><LF>	**Temperature Original, Wetness Measurement** **Example:** Request: 0M1! Response: 00012<CR><LF> Response: 0<CR><LF> Request: 0D0! Response: 0+23.96+29.31<CR><LF>
aM9!，aMC9!	a0014<CR><LF> a:Sensor address 001: Measurement data will be ready in 001 seconds 4: Number of measurement data returned by aD0! <CR><LF>:terminates the response aD0! Response data format： a<±temperature><±temperature_calibed><+wetness><+wetness>[<CRC>]<CR><LF>	**Temperature Original, Temperature Calibed, Wetness, Wetness Measurement** **Example:** Request: 0M9! Response: 00014<CR><LF> Response: 0<CR><LF> Request: 0D0! Response: 0+22.59+22.59+0.20+0.20<CR><LF>
aC!，aCC!	a00102<CR><LF> a:Sensor address 001: Measurement data will be ready in 001 seconds 02: Number of measurement data	**Temperature Calibed, Wetness Measurement** **Example:** Request: 0C! Response: 000102<CR><LF> Request: 0D0!

	returned by aD0! <CR><LF>:terminates the response aD0! Response data format: a<±temperature_calibed><+wetness>[<CRC>]<CR><LF>	Response: 0+23.96+29.31<CR><LF>
aC1!, aCC1!	a00102<CR><LF> a:Sensor address 001: Measurement data will be ready in 001 seconds 02: Number of measurement data returned by aD0! <CR><LF>:terminates the response aD0! Response data format: a<±temperature><+wetness>[<CRC>]<CR><LF>	**Temperature Original, Wetness Measurement** **Example:** Request: 0C1! Response: 000102<CR><LF> Request: 0D0! Response: 0+23.96+29.31<CR><LF>
aC9!, aCC9!	a00104<CR><LF> a:Sensor address 001: Measurement data will be ready in 001 seconds 04: Number of measurement data returned by aD0! <CR><LF>:terminates the response aD0! Response data format: a<±temperature><±temperature_calibed><+wetness><+wetness>[<CRC>]<CR><LF>	**Temperature Original, Temperature Calibed, Wetness, Wetness Measurement** **Example:** Request: 0C9! Response: 000104<CR><LF> Request: 0D0! Response: 0+22.59+22.59+0.20+0.20<CR><LF>
aV!	A0021<CR><LF> a:Sensor address 002: Measurement data will be ready in 002 seconds 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: 00021<CR><LF> Response: 0<CR><LF> Request: 0D0! Response: 0+0<CR><LF>, "+0" indicate sensor normal, "+1" means sensor error.
aD0! aD1! aD2!	[<svvvv><svvvv><svvvv>…][<CRC>]<CR><LF> [<svvvv>]: data value	Send Data since the last aM, aMC, aC, aCC, aV command, The data returned depends on the command sent most recently.

	[<CRC>]: Optional 3 characters CRC checksum, <CR><LF>:terminates the response	
aR0!，aRC0!	Response data format： a<±temperature_calibed><+wetness>[<CRC>]<CR><LF>	**Temperature Calibed, Wetness Measurement** **Example:** Request: 0R0! Response: 0+23.96+29.31<CR><LF>
aR1!，aRC1!	Response data format： a<±temperature><+wetness>[<CRC>]<CR><LF>	**Temperature Original, Wetness Measurement** **Example:** Request: 0R1! Response: 0+23.96+29.31<CR><LF>
aR9!，aRC9!	Response data format： a<±temperature><±temperature_calibed><+wetness><+wetness>[<CRC>]<CR><LF>	**Temperature Original, Temperature Calibed, Wetness, Wetness Measurement** **Example:** Request: 0R9! Response: 0+22.59+22.59+0.20+0.20<CR><LF>
aXR_TUNIT!	aTUNIT=<X> <X> is temperature unit: C: degrees centigrade F: degrees fahrenheit	**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=<±TOFFSET> <±TOFFSET>: temperature offset value between -10.00~10.00, it will be effective when issuing a new measurement command. <±temperature_calibed>=<±temperature>+<±TOFFSET>	**Query temperature offset value** **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_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>

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.

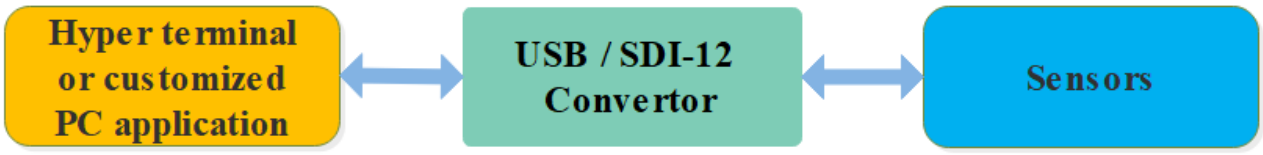
This chapter mainly introduces the communication or parameter setting on a computer for sensor through the SDI-12 converter (SDI12ELF20).

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, 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.

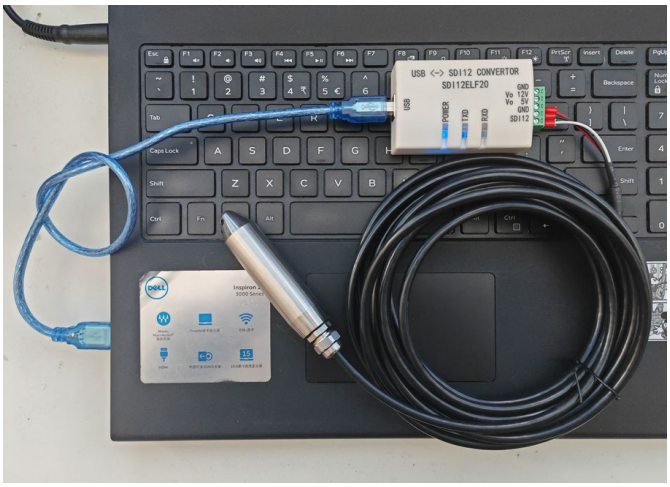
Testing Software Download	
Terminal (universal serial port debugging tool)	https://www.infwin.com/resource-serial-port-com-development-tool/
SensorOneSetSDI12 (SDI-12 sensor configuration utility)	https://www.infwin.com/resource-sensoronesetsdi12-configuration-utility-for-sdi-12-sensors/

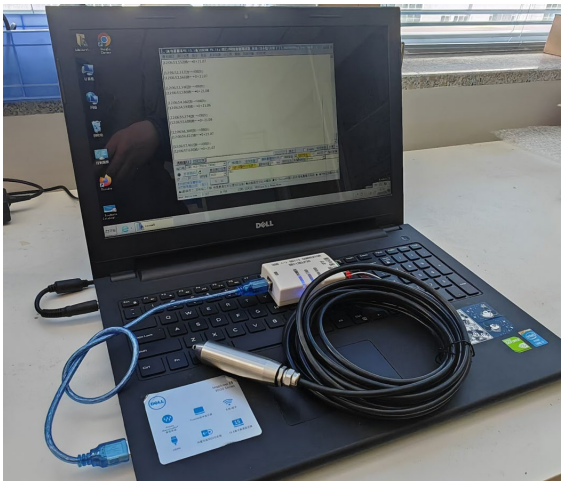
A.2 Testing Example

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/

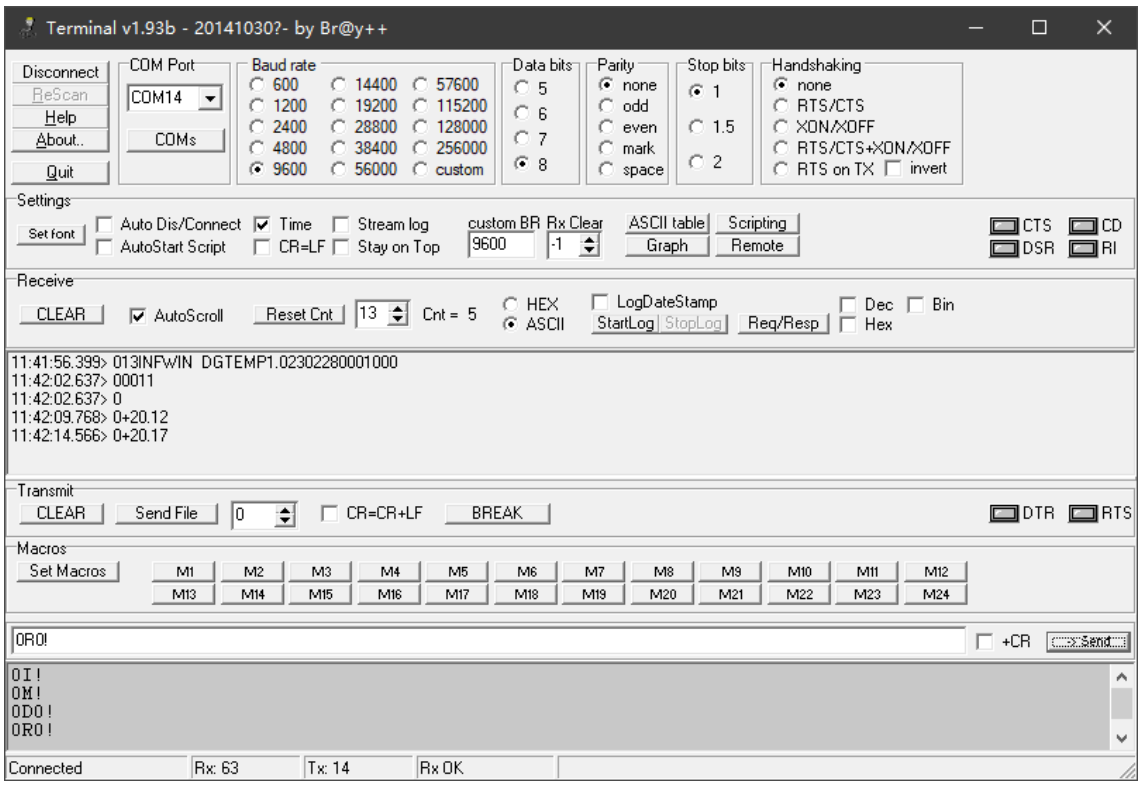
■ **Connections**





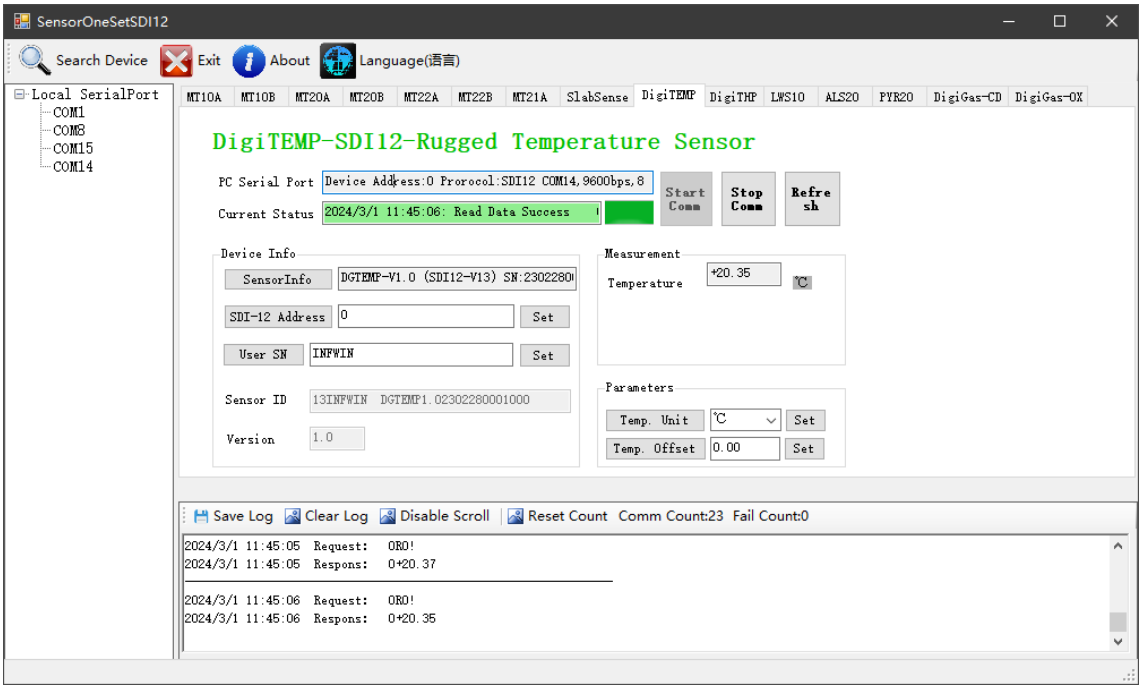
■ **Testing with serial port utility "Terminal"**

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.



Copyright and Trademark

Version Control

Date	Version	Comment	Updated by
2024-02-11	V1.0	Initial Creation	sl51930