

# LWS10

## Leaf Wetness Sensor (SDI-12 Interface) User Manual





## Index

1 Customer Support	3
2 Introduction	4
<ul><li>2.1 Brief</li><li>3 Wiring diagrams</li><li>3.1 SDI-12 Interface</li></ul>	4 
4 Dimension and Ordering Infomation	7
4.1 Dimension 4.2 Ordering Information	7 8
5 SDI-12 Communication	9
<ul><li>5.1 SDI-12 Interface and Protocol</li><li>5.1.1 SDI-12 Interface</li><li>5.1.2 Protocol</li></ul>	<b>10</b> 10 10
Appendix A SDI-12 Sensor Testing and Settings	15
A.1 Testing SDI-12 Sensors with SDI12ELF20 Converter A.2 Testing Example	15 16
Copyright and Trademark	
Version Control	



## **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	SDI-12 Interface: Quiescent Current : <10uA	
Consumption	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





## **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	Е	4.5-28V DC	
Code 4: Cable Length	002	2 meters	
XXX 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>	-	Tab character
<sapce></sapce>	-	Space character
<cr></cr>	-	Carriage return character
<lf></lf>	-	Line feed character
<checksum></checksum>	-	SUM Checksum
<crc_adi></crc_adi>		ADI protocol CRC Checksum
<crc></crc>	-	SDI-12 protocol CRC Checksum
<verify_status></verify_status>	-	Sensor Verification status
<pre>&lt;±temperature_calibed&gt;</pre>	°C	Temperature Calibed, the value is output according
	°F	to the temperature unit setting.
<±temperature>	°C	Temperature Original, the value is output according
	°F	to the temperature unit setting.
<=TOFFSET>	°C	Temperature Offset,
	°F	<±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></lf></cr>	Acknowledge Active
		a: Sensor address
		Example:
		Request: 0!
		Response: 0 <cr><lf></lf></cr>
aI!	allecccccccmmmmmvvvxxxxxxx	Send Identification
	xxxx <cr><lf></lf></cr>	a: Sensor address
		ll:SDI-12 Version Number
		ccccccc: 8 characters vendor identification
		mmmmmm: 6 characters specifying the sensor model
		number
		vvv: 3 characters specifying the sensor version
		xxxxxxxxxxx 13 characters serial number
		<cr><lf>: terminates the response</lf></cr>
		Example:
		Request: 0I!
		Response: 013INFWIN DGTHP
		2.02305170016000 <cr><lf></lf></cr>
?!	a <cr><lf></lf></cr>	Sensor Address Query
		a:Sensor address
		Example:
		Request: ?!
		Response: 0 <cr><lf></lf></cr>
aAb!	b <cr><lf></lf></cr>	Change Sensor address
		a:Current Sensor address
		b:New Sensor address
		Example:
		Request: 0A1!



		Response: 1 <cr><lf></lf></cr>
aM!, aMC!	a0012 <cr><lf></lf></cr>	Temperature Calibed, Wetness Measurement
	a:Sensor address	Example:
	001: Measurement data will be ready	Request: 0M!
	in 001 seconds	Response: 00012 <cr><lf></lf></cr>
	2: Number of measurement data	Response: 0 <cr><lf></lf></cr>
	returned by aD0!	Request: 0D0!
	<cr><lf>:terminates the response</lf></cr>	Response: 0+23.96+29.31 <cr><lf></lf></cr>
	aD0! Response data format:	
	a<±temperature_calibed><+wetness>	
	[ <crc>]<cr><lf></lf></cr></crc>	
aM1!, aMC1!	a0012 <cr><lf></lf></cr>	Temperature Original, Wetness Measurement
	a:Sensor address	Example:
	001: Measurement data will be ready	Request: 0M1!
	in 001 seconds	Response: 00012 <cr><lf></lf></cr>
	2: Number of measurement data	Response: 0 <cr><lf></lf></cr>
	returned by aD0!	Request: 0D0!
	<cr><lf>:terminates the response</lf></cr>	Response: 0+23.96+29.31 <cr><lf></lf></cr>
	aD0! Response data format:	
	a<±temperature><+wetness>[ <crc>]</crc>	
	<cr><lf></lf></cr>	
aM9!, aMC9!	a0014 <cr><lf></lf></cr>	Temperature Original, Temperature Calibed,
		Wetness, Wetness Measurement
	a:Sensor address	
	001: Measurement data will be ready	Example:
	in 001 seconds	Request: 0M9!
	4: Number of measurement data	Response: 00014 <cr><lf></lf></cr>
	returned by aD0!	Response: 0 <cr><lf></lf></cr>
	<cr><lf>:terminates the response</lf></cr>	Request: 0D0!
		Response: 0+22.59+22.59+0.20+0.20 <cr><lf></lf></cr>
	aD0! Response data format:	
	a<=temperature<=temperature_callb	
	CR > IF>	
aC!, aCC!	a00102 <cr><lf></lf></cr>	Temperature Calibed, Wetness Measurement
	a:Sensor address	Example:
	001: Measurement data will be ready	Request: 0C!
	in 001 seconds	Response: 000102 <cr><lf></lf></cr>
	02: Number of measurement data	Request: 0D0!

	<cr><lf>:terminates the response</lf></cr>	
	aD0! Response data format:	
	a<±temperature_calibed><+wetness>[	
	<crc>]<cr><lf></lf></cr></crc>	
aC1!, aCC1!	a00102 <cr><lf></lf></cr>	Temperature Original, Wetness Measurement
	a:Sensor address	Example:
	001: Measurement data will be ready	Request: 0C1!
	in 001 seconds	Response: 000102 <cr><lf></lf></cr>
	02: Number of measurement data	Request: 0D0!
	returned by aD0!	Response: 0+23.96+29.31 <cr><lf></lf></cr>
	<cr><lf>:terminates the response</lf></cr>	
	aD0! Response data format:	
	a<±temperature><+wetness>[ <crc>]</crc>	
	<cr><lf></lf></cr>	
aC9!, aCC9!	a00104 <cr><lf></lf></cr>	Temperature Original, Temperature Calibed,
		Wetness, Wetness Measurement
	a:Sensor address	
	001: Measurement data will be ready	Example:
	in 001 seconds	Request: 0C9!
	04: Number of measurement data	Response: 000104 <cr><lf></lf></cr>
	returned by aD0!	Request: 0D0!
	<cr><lf>:terminates the response</lf></cr>	Response: 0+22.59+22.59+0.20+0.20 <cr><lf></lf></cr>
	aD0! Response data format:	
	a<±temperature><±temperature_calib	
	ed><+wetness><+wetness>[ <crc>]&lt;</crc>	
	CR> <lf></lf>	
aV!	A0021 <cr><lf></lf></cr>	Sensor Verification Command
		Example:
	a:Sensor address	Request: UV!
	in 002 accords	Response: 00021 <cr><lf></lf></cr>
	in 002 seconds	Response: 0 <ck><lr></lr></ck>
	CP>(F):terminates the response	Request. 0D0: $P_{asponse}$ $0+0 "+0" indicate sensor por$
	CR CR CR Chi - actiminates the response	mal, "+1" means sensor error.
	aD0! Response data format:	
	a <verify_status><cr><lf></lf></cr></verify_status>	
aD0!	[ <svvvv><svvvv>][<crc< td=""><td>Send Data since the last aM, aMC, aC, aCC, aV</td></crc<></svvvv></svvvv>	Send Data since the last aM, aMC, aC, aCC, aV
aD1!	>] <cr><lf></lf></cr>	command, The data returned depends on the command
aD2!	[ <svvvv>]: data value</svvvv>	sent most recently.

Unleash Your Imagination

returned by aD0!

INFWIN

http://www.infwin.com

Response: 0+23.96+29.31<CR><LF>

	ash,Your gination	http://www.infwin.com
	[ <crc>]: Optional 3 characters CRC checksum, <cr><lf>:terminates the response</lf></cr></crc>	
aR0!, aRC0!	Response data format: a<±temperature_calibed><+wetness>[ <crc>1<cr><lf></lf></cr></crc>	Temperature Calibed, Wetness Measurement
		Request: 0R0! Response: 0+23.96+29.31 <cr><lf></lf></cr>
aR1!, aRC1!	Response data format: a<±temperature><+wetness>[ <crc>] <cr><lf></lf></cr></crc>	Temperature Original, Wetness Measurement Example: Request: 0R1!
aR9!, aRC9!	Response data format: a<±temperature><±temperature calib	Response: 0+23.96+29.31 <cr><lf> Temperature Original, Temperature Calibed, Wetness, Wetness Measurement</lf></cr>
	ed><+wetness><+wetness>[ <crc>]&lt; CR&gt;<lf></lf></crc>	<b>Example:</b> Request: 0R9! Response: 0+22.59+22.59+0.20+0.20 <cr><lf></lf></cr>
aXR_TUNIT!	aTUNIT= <x> <x> is temperature unit: C: degrees centigrade F: degrees fahrenheit</x></x>	Query temperature unit Example: Request: 0XR_TUNIT! Response: 0TUNIT=C <cr><lf></lf></cr>
aXW_TUNIT_ <x>!</x>	aTUNIT= <x></x>	Configure temperature unit Example: Request: 0XW_TUNIT_C! Response: 0TUNIT=C <cr><lf></lf></cr>
aXR_TOFFSE T!	aTOFFSET=<±TOFFSET> <±TOFFSET>: temperature offset value between -10.00~10.00, it will be effective when issuing a new measurement command. <±temperature_calibed>=<±temperatu re>+<±TOFFSET>	Query temperature offset value Example: Request: 0XR_TOFFSET! Response: 0TOFFSET=+1.00 <cr><lf></lf></cr>
aXW_TOFFSE T_<±TOFFSET >!	aTOFFSET=<±TOFFSET>	Configure temperature offset value Example: Request: 0XW_TOFFSET_+1.00! Response: 0TOFFSET=+1.00 <cr><lf></lf></cr>
aXR_SN!	aSN= <sssssss> <ssssssss> is 8-digits serial number</ssssssss></sssssss>	Query serial number Example: Request: 0XR_SN! Response: 0SN=12345678 <cr><lf></lf></cr>
aXW_SN_ <sss sssss&gt;!</sss 	aSN= <sssssss></sssssss>	Configure serial number Example:

Endeavour Technology Co.,Ltd. Dalian, CHINA. Website: http://www.infwin.com Call: +86-411-66831953 +86-4000-511-521 Email : infwin@163.com

-



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.

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



Terminal v1.93b - 201410302- by Br@v++	– n	×
Disconnect         COM Port         Baud rate         C 600         C 14400         C 57600         C 5         C 6         C none         C 1         C none         C 7         C odd         C 8         C 9600         C 15         C 800/×015         C 8         C 15         C 800/×015         C 8         C 15         C 800/×015         C 800/×015		
Set font       Auto Dis/Connect       ✓ Time       Stream log       custom BR       Rx Clear       ASCII table       Scripting         Set font       ✓ AutoStart Script       CR=LF       Stay on Top       9600       -1	CTS	CD RI
Receive     CuteAR     AutoScroll     Reset Cnt     13     Cnt = 5     C HEX     LogDateStamp     Dec     Bin       CLEAR     Image: AutoScroll     Reset Cnt     13     Cnt = 5     Image: ASCII     StartLog     StartLog     Req/Resp     Hex		
11:41:56.399> 013INFWIN_DGTEMP1.02302280001000 11:42:02.637> 00011 11:42:02.637> 0 11:42:07.688 0+20.12 11:42:07.688 0+20.12		
Transmit CLEAR Send File 0  CR=CR+LF BREAK	DTR	RTS
Macros         M1         M2         M3         M4         M5         M6         M7         M8         M9         M10         M11         M12           M13         M14         M15         M16         M17         M18         M19         M20         M21         M22         M23         M24		
0R0!	🗆 +CR [	
0I! 0M!		^
0D0 ! 0R0 !		~
Connected Rx: 63 Tx: 14 Rx 0K		

#### 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.

🔛 SensorOneSetSDI12	X
Search Device	🔀 Exit 🕧 About 🎡 Language(语言)
-Local SerialPort	MT10A MT10B MT20A MT20B MT22A MT22B MT21A SlabSense DigiTEMP DigiTHP LWS10 ALS20 PYR20 DigiGas-CD DigiGas-OX
- COM1 - COM15 - COM14	DigiTEMP-SDI12-Rugged Temperature Sensor
	Current Status 2024/3/1 11:45:06: Read Data Success 1 Come Come Sh
	Device InfoMeasurement
	SensorInfo         DGTEMF-V1.0 (SDI12-V13) SN:23022800         Temperature         *20.35
	SDI-12 Address 0 Set
	User SN INFWIN Set
	Sensor ID 131NFWIN DGTEMP1.02302280001000
	Temp. Unit C Set
	Temp. Offset 0.00 Set
	: 💾 Save Log 🕍 Clear Log 🕍 Disable Scroll 🛛 🖓 Reset Count Comm Count:23 Fail Count:0
	2024/3/1 11:45:05 Request: 0R0! 2024/3/1 11:45:05 Respons: 0+20.37
	2024/3/1 11:45:06 Request: 0R0!
	2024/3/1 11:45:06 Respons: 0+20.35
	رانه ۱۱



### **Copyright and Trademark**

This document is copyrighted, by Dalian Endeavour Technology Co., Ltd. All rights are reserved. Dalian Endeavour Technology Co., Ltd. reserves the right to make improvements to the products described in this manual at any time without notice. No part of this manual may be reproduced, copied, translated or transmitted in any form or by any means without the prior written permission. Information provided in this manual is intended to be accurate and reliable. However, Dalian Endeavour Technology Co., Ltd. assumes no responsibility for its use, nor for any infringements upon the rights of third parties, which may result from its use.

INFWIN® is the trademark of Dalian Endeavour Technology Co., Ltd.

### **Version Control**

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