SDI12ELF20

SDI-12 to USB Converter

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 SDI-12 Introduction

SDI-12 is a microprocessor-based data logger interface standard. SDI-12 stands for 1200 baud serial digital interface. It can use a data logger to connect multiple sensors for data communication, and support a cable up to 60 meters between the data logger and the sensor. For more information about SDI-12, please refer to the document published by the SDI-12 Group.

2.2 Product Brief

The SDI12ELF20 implements a plug-and-play solution for connecting SDI-12 sensors to personal computers for controlling or testing devices with SDI-12 interface. The SDI12ELF20 is a communication conversion device used to connect the USB master device and the SDI-12 device. The USB master device can be a host that supports the USB interface, such as a computer or Raspberry PI, and the SDI-12 sensor can be a smart sensor device that supports the SDI-12 interface. The SDI12ELF20 can be used for system integration, sensor testing, or data acquisition systems based on SDI-12 sensors, with the following features:

- SDI-12 to USB communication interface
- SDI-12 communication data transmission and monitoring
- USB powered converter does not require additional power supply
- Integrated 5V and 12V power output for sensor testing
- SDI-12 communication interface with surge and lightning protection
- Configurable communication baud rate, parity, stop bit
- Configurable SDI-12 protocol BREAK and MARKING time
- ODM/OEM service

Application field

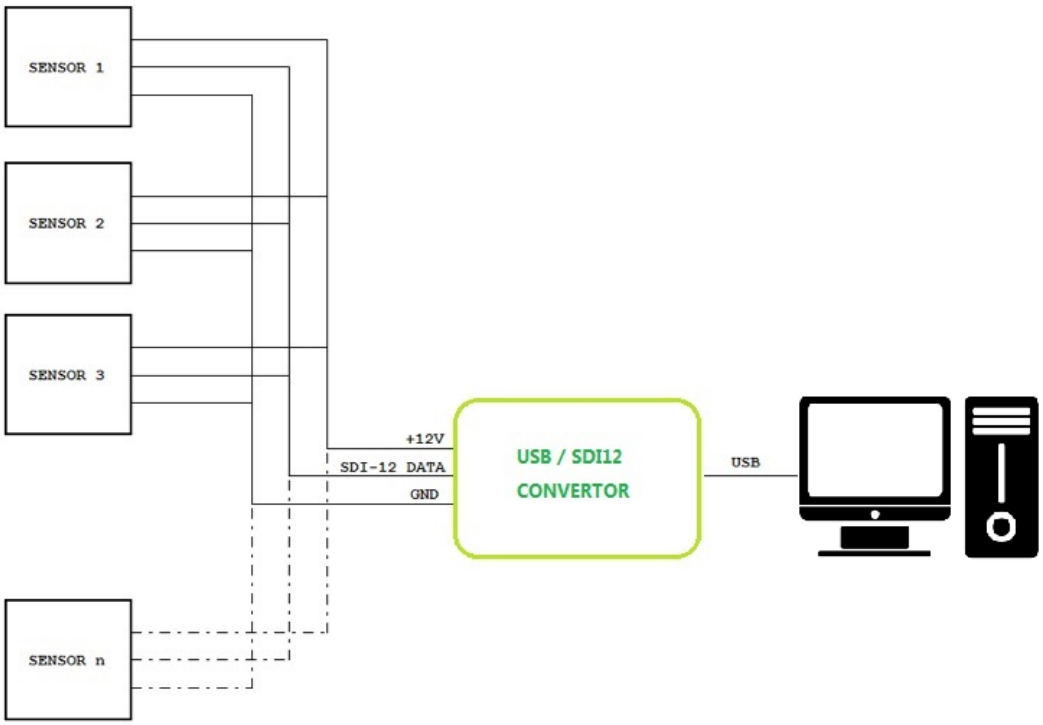
- SDI-12 data recording
- SDI-12 sensor testing
- Debugging the SDI-12 interface
- SDI-12 installation

2.3 Specification

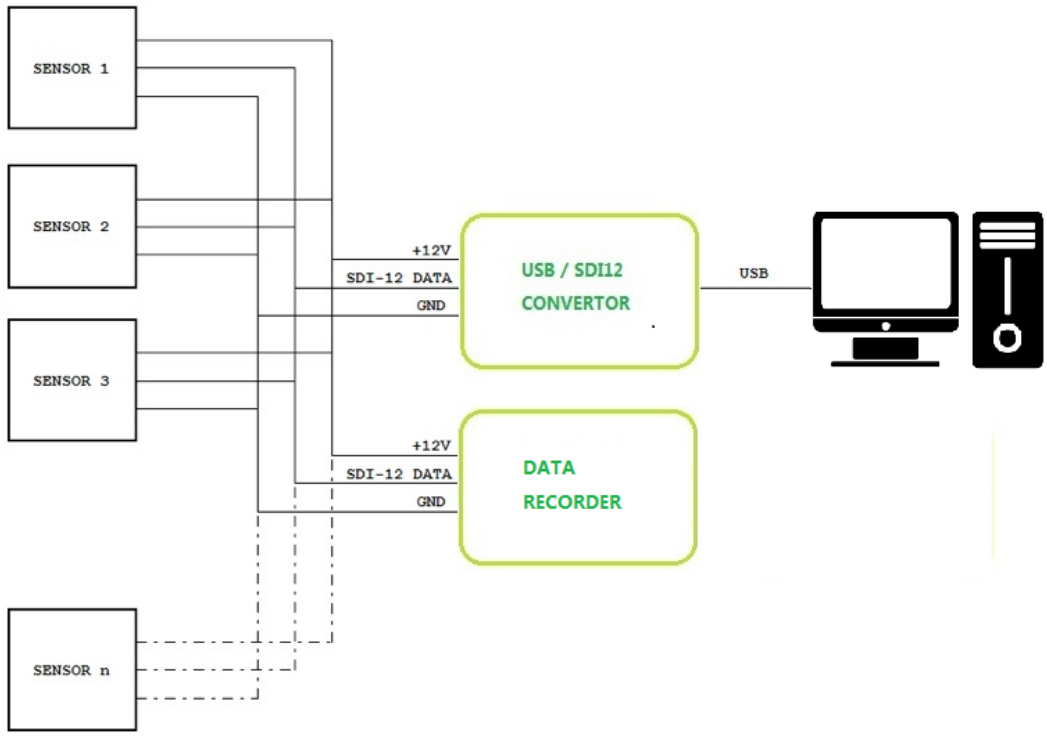
Specification	
Function	Transmission mode and listening mode of SDI-12 bus
USB Interface and power input	USB Type-B female connector, Virtual COM interface TX Buffer: 250 bytes RX Buffer: 250 bytes Baudrate: 4800,9600,19200,38400,57600,115200 bps Parity: none, odd, even Stop bit: 1 bit, 2 bit
Power Supply	+5V,Over current protection +12V,Over current protection
SDI-12 Interface	TX Buffer: 250 bytes RX Buffer: 250 bytes Surge and lightning protection BREAK TIME：Configurable MARKING TIME：Configurable
IP Rating	IP20 NEMA1
Operating Environment	-40~85℃
Dimension	81*46*26mm

2.4 Application Diagram

Sensor testing or data recording based on USB host



USB host based data collector and sensor data monitoring



3 Functions

The SDI-12 standard defines a set of commands for configuring sensors and measurements. After receiving a specific SDI-12 command, the device performs an internal task, responding the information, converting time, or sending measurement data.

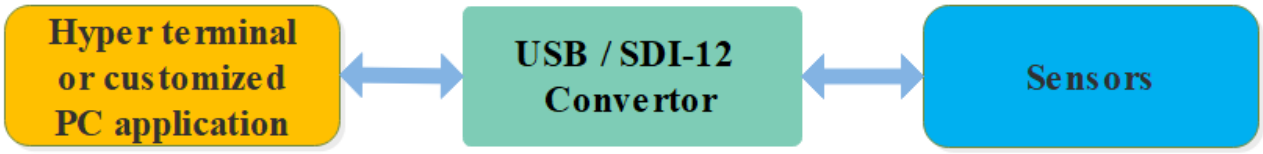
The SDI-12 command is an ASCII string send out by the data collector. The SDI12ELF20 can be controlled by a PC, Laptop or other USB master device, converts the command string to the logic level and baudrate specified in the SDI-12 standard.

In addition, SDI12ELF20 handles BREAK, MARKING, and all other details of the SDI-12 protocol. When receiving data or status information from the sensor, SDI12ELF20 extracts the corresponding ASCII string and sends it to the virtual COM interface of the USB host.

3.1 Transmission Mode

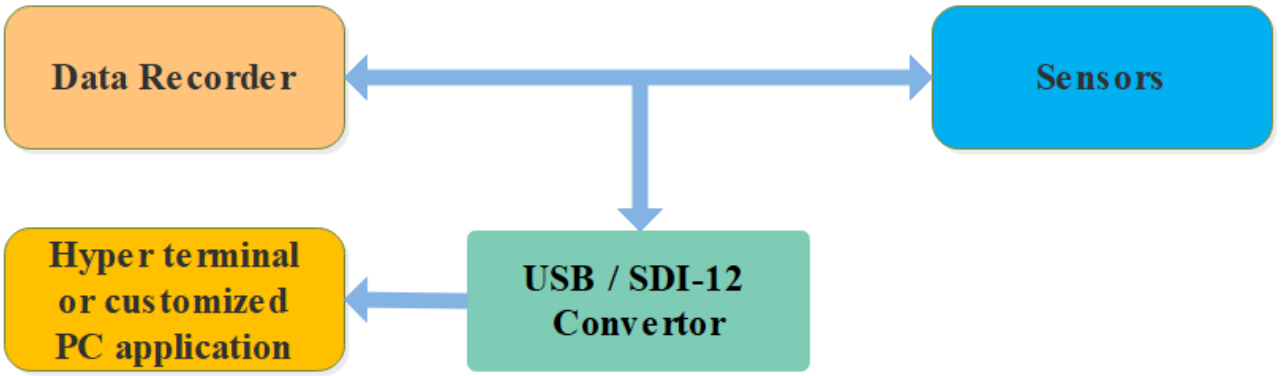
Converter in this mode transfers the SDI-12 command from the USB host to the SDI-12 interface. When the USB host does not send any data to the converter, the converter will automatically be in monitoring mode.

The application is a module built on USB/SDI-12 converter. It receives commands from the USB interface (e.g. the user via a PC application), transmits the command and sends it to the SDI-12 interface, waits for the sensor to respond and the response (measurement results, etc.) is sent back to the USB interface, and the data can be accessed via the PC application. The converter supports all SDI-12 instructions.



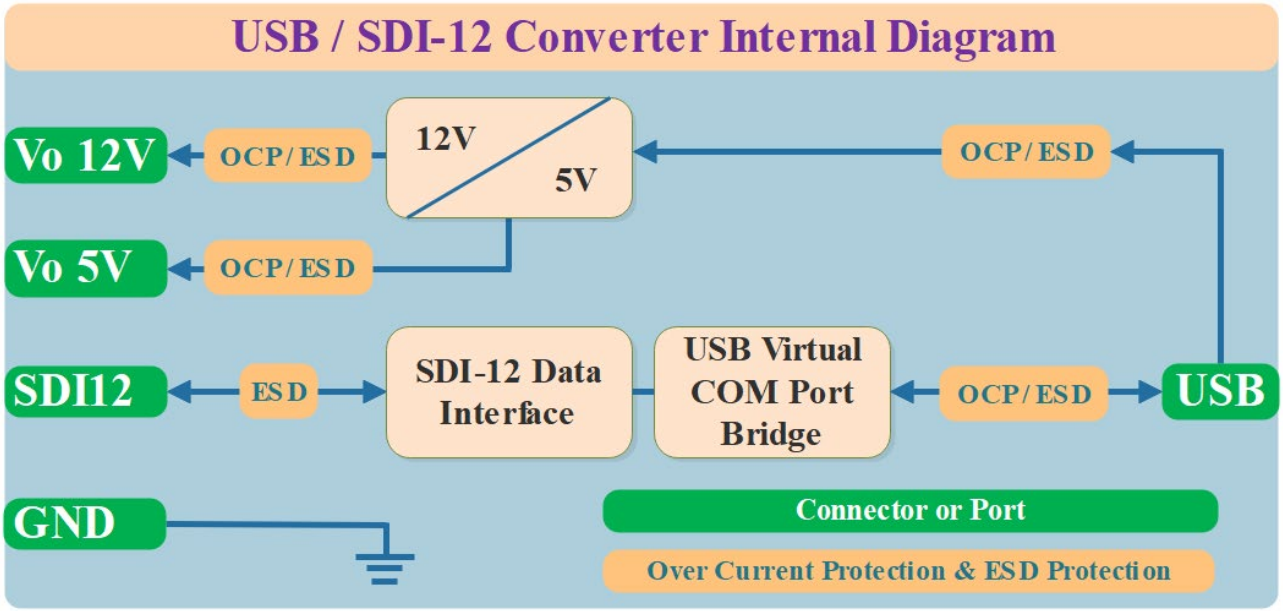
3.2 Monitoring Mode

This mode will listen to all communication data between the data collector and the sensor. When the USB host does not send any data to the converter, the converter will automatically be in listening mode.



3.3 Power Supply

The converter obtains its internal operating power supply voltage from the USB interface and provides +5V and +12V power supply to the sensor. The +5V power output is provided directly by the USB input power supply and has overcurrent protection, The +12V supply voltage is generated by an internal boost and has overcurrent protection. If the sensor requires high power supply current, use an external power supply to power the sensor, and connect the GROUND of external power supply with the converter's.



4 Electrical Characteristic

4.1 Connector

This table shows the converter's connector name, type (DIO - digital input/output, P - power supply), and corresponding description.

#	Connector	Type	Description
1	GND		Signal Ground
2	Vo 12V	P	SDI-12 Power Supply：+12V
3	Vo 5V	P	SDI-12 Power Supply：+5V
4	GND		Signal Ground
5	SDI-12	DIO	SDI-12 Data Line

4.2 Electrical Characteristic

Unless otherwise stated, the test condition is Temperature TA = 25°C.

Symbol Name	Parameters	Condition	MIN	TYP	MAX	UNIT
USB Power Input						
Vusb	Vusb		4.5	5.0	5.5	V
Iusb	Iusb				500(1)	mA
SDI-12 Data Line						
VIL	INPUT LOW VOLTAGE LEVEL		-0.5	0.0	1	V
VIH	INPUT HIGH VOLTAGE LEVEL		2.0	5.0	5.5	V
VOL	OUTPUT LOW VOLTAGE LEVEL		-0.5	0.0	1	V
VOH	OUTPUT HIGH VOLTAGE LEVEL		2.9	3.6	3.6	V
Power Supply						

Vo_12V	SENSOR SUPPLY VOLTAGE (12V)		11.0	12	12.9	V
I_Vo_12V	SENSOR SUPPLY CURRENT (12V)				100 (2)	mA
Vo_5V	SENSOR SUPPLY VOLTAGE (5V)		4.7	5	5.5	V
I_Vo_5V	SENSOR SUPPLY CURRENT (5V)				200 (3)	mA
OPERATING TEMPERATURE RANGE			-40 to +85			°C

Note 1: Internal polyfuse: 500mA; Itrip=1000mA; Trip time at 8000mA: 0.1s

Note 2: Internal polyfuse: 100mA; Itrip=200mA; Trip time at 500mA: 1s

Note 3: Internal polyfuse: 200mA; Itrip=460mA; Trip time at 8000mA: 0.08s

5 SDI-12 Basics

SDI-12 is a serial data communication standard for interfacing multiple sensors with a data recorder. The SDI-12 uses a shared bus with 3 wires, power, data, ground. Data rate at 1200 baud.

Each sensor at the bus gets a unique address which is in the range ASCII [0-9, a-z, A-Z,?]. The default address of every sensor is ASCII[0]. When setting up a SDI-12 sensor network, every sensor needs to be configured with a unique address. This can be done using the "Change Address Command". A sensor can typically measure one or more parameters. Sensor manufacturers usually specify "Extended Commands" to configure or calibrate sensors. These commands are specified by the manufacturer, but they follow the command structure specified by SDI-12.

Each SDI-12 command is an ASCII string, starting with the sensor address and terminated by a "!" character.

A typical recorder/sensor measurement sequence proceeds as follows:

1) The data recorder wakes all sensors on the SDI-12 bus with a break.

2) The recorder transmits a command to a specific, addressed sensor, instructing it to make a measurement.

3) The addressed sensor responds within 15.0 milliseconds returning the maximum time until the measurement data will be ready and the number of data values it will return.

4) If the measurement is immediately available, the recorder transmits a command to the sensor instructing it to return the measurement result(s). If the measurement is not ready, the data recorder waits for the sensor to send a request to the recorder, which indicates that the data are ready. The recorder then transmits a command to get the data.

5) The sensor responds, returning one or more measurement results

SDI-12 command example:

Request	Response	Comment
a!	a<CR><LF> Acknowledge Active a: Sensor address <CR><LF>: terminates the response	**Example:** Request: 0! Response: 0<CR><LF>
aI!	allccccccccmmmmmmvvvxxxxxxxxxxxxxxx<CR><LF> Send Identification a: Sensor address ll:SDI-12 Version Number	0 is the sensor address (sensor zero). Upon receiving this command, the sensor will send an ASCII string containing sensor address, a SDI-12 version, company name, sensor model number, sensor version number and sensor serial 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 SlabSense 1.01909250001000<CR><LF>
?!	a<CR><LF> Sensor Address Query a:Sensor address <CR><LF>: terminates the response	**Example:** Request: ?! Response: 0<CR><LF>
aAb!	b<CR><LF> Change Sensor address a:Current Sensor address b:New Sensor address <CR><LF>: terminates the response	**Example:** Request: 0A1! Response: 1<CR><LF>
aM!	a0302<CR><LF> a:Sensor address 030: Measurement data will be ready in 030 seconds 2: Number of measurement data returned by aD0! <CR><LF>:terminates the response aD0! Possible Response data format： a<±DataValue0><±DataValue1><CR><LF>	The standard process to carry out a measurement is to send a measurement request upon which the sensor responds with the time that is required to carry out the measurement and the number of data items being returned. After waiting the time that the sensor requires to carry out the measurement, the data recorder sends a “Read Command” to get the measurement results. **Example:** Request: 0M! Response: 00302<CR><LF> Wait ~30 seconds Response: 0<CR><LF> Request: 0D0! Response: 0+27+1050<CR><LF> +27+1050 is the two measurement results which may be 27°C and 1050 milibar.

6 Installation and Usage

6.1 Hardware Connection and Driver Installation

A PC or laptop with a USB port, or other device that can be used as a USB host, such as the Raspberry PI. Connect to the converter through its USB port.

■ 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

6.2 Testing Software

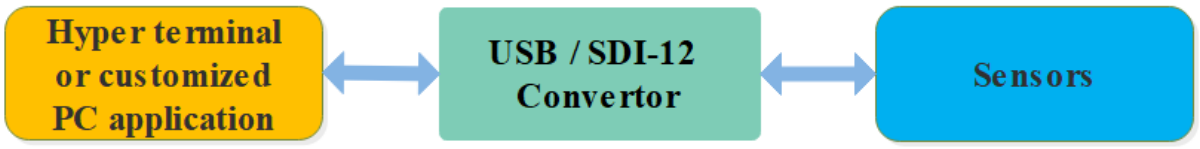
Users can use any serial communication debugging software for communication debugging, take Windows as an example, such as Terminal, etc., the converter's default communication parameter is 9600bps, None parity, 8 data bits, 1 stop bit. Use the ASCII mode to send and receive data.

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

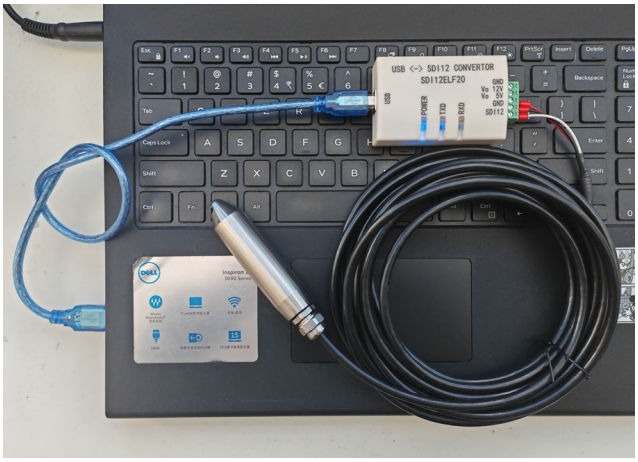
6.3 Testing Example

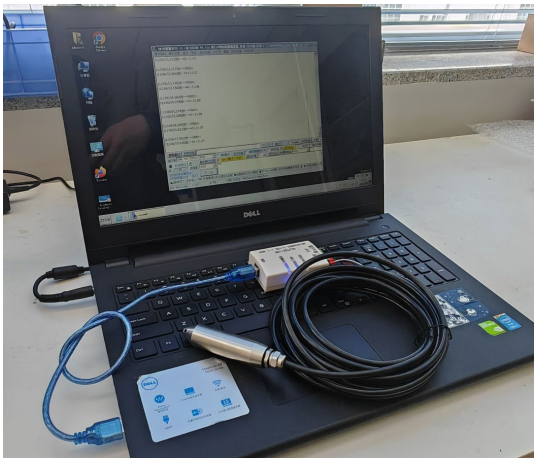
In this example, the SDI12ELF20 converter is connected to the computer USB interface to communicate with the rugged temperature sensor DigiTEMP. The SDI12ELF20 converter provides power supply for the sensor and reads device information and data on the computer.

Application Diagram



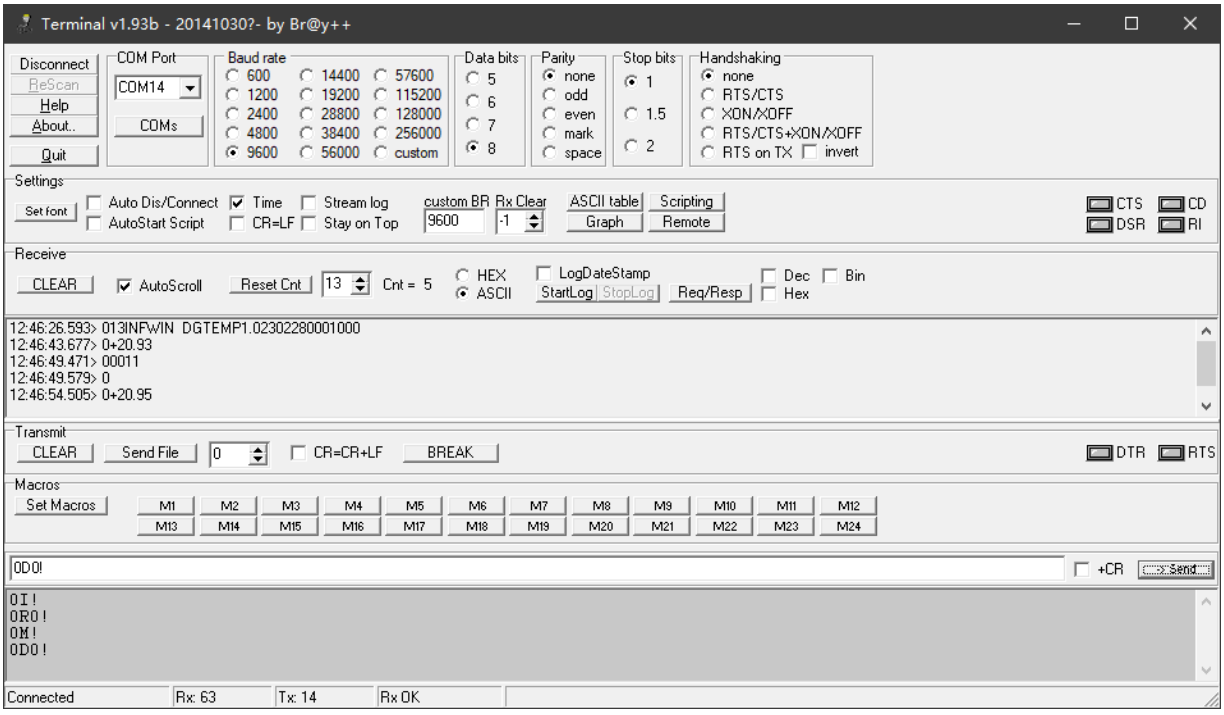
Device Connection





Testing Software

Take “Terminal” serial communication utility as example, select proper serial port, set the baud rate to 9600bps, no parity, 8 data bits, and 1 stop bit (the default communication setting of SDI12ELF20), open the serial port, enter the SDI-12 command, and send the command. Note that you need to use ASCII format for data communication.



7 Parameter Settings and factory Settings

7.1 Parameter Settings

The parameters of the converter can be set as follows. The user can use any serial debugging software to send AT instructions for parameter setting; Note that after modifying the parameters, use the AT+SAVE command to save the parameters and the ATZ command to restart the device for the Settings to take effect.

The following table describes the meanings of common identifiers in parameter Settings:

Symbols	Unit	Description
<TAB>	-	Tab Character
<SPACE>	-	Space Character
<CR>	-	Carriage return character
<LF>	-	Line feed character
EEPROM	-	Power down storage memory

The following parameter setting instructions are supported by the converter:

Functions	Command	Description
Restart the device	ATZ<CR><LF>	**Example:** **Request:** ATZ<CR><LF> **Response:** Restarting... <CR><LF>
Restore the factory Settings	ATR<CR><LF>	**Example:** **Request:** ATR<CR><LF> **Response:** Factory Parameters Restored, Restarting...<CR><LF>
Querying Device Information	AT+VER<CR><LF>	**Example:** **Request:** AT+VER=?<CR><LF> **Response:** ProductCode: SDI12ELF20 FirmWareVersion: 2.1 Manufactor: INFWIN Website: www.infwin.com<CR><LF>OK<CR><LF>
Save device parameters to	AT+SAVE<CR><LF>	**Example:** **Request:** AT+SAVE<CR><LF>

EEPROM		**Response:** Parameters Saved, Restarting...<CR><LF>
Restores device parameters from EEPROM to register	AT+RECALL<CR><LF>	**Example:** **Request:** AT+ RECALL<CR><LF> **Response:** Parameters Recalled, Restarting...<CR><LF>
Serial communication baudrate	Query available baudrate: AT+BAUD?<CR><LF> Query current baudrate: AT+BAUD=?<CR><LF> Set current baudrate: AT+BAUD=<BAUDRATE><CR><LF>	Available <BAUDRATE>: 4800：4800bps 9600：9600bps（Default） 19200：19200bps 38400：38400bps 57600：57600bps 115200：115200bps **Example:** **Request:** AT+BAUDRATE=?<CR><LF> **Response:** 9600<CR><LF>OK<CR><LF> **Request:** AT+BAUD=9600<CR><LF> **Response:** <CR><LF>OK<CR><LF>
Serial communication parity	Query available parity: AT+PARITY?<CR><LF> Query current parity: AT+ PARITY =?<CR><LF> Set current parity: AT+ PARITY =<PARITY><CR><LF>	Available <PARITY>: N：None（Default） E：Even O：Odd **Example:** **Request:** AT+ PARITY=?<CR><LF> **Response:** N<CR><LF>OK<CR><LF> **Request:** AT+ PARITY=N<CR><LF> **Response:** <CR><LF>OK<CR><LF>
Serial communication stopbits	Query available stopbits: AT+STOPBITS?<CR><LF> Query current stopbits: AT+ STOPBITS =?<CR><LF>	Available <STOPBITS>: 1：1 StopBits（Default） 2：2 StopBits **Example:** **Request:** AT+STOPBITS=?<CR><LF>

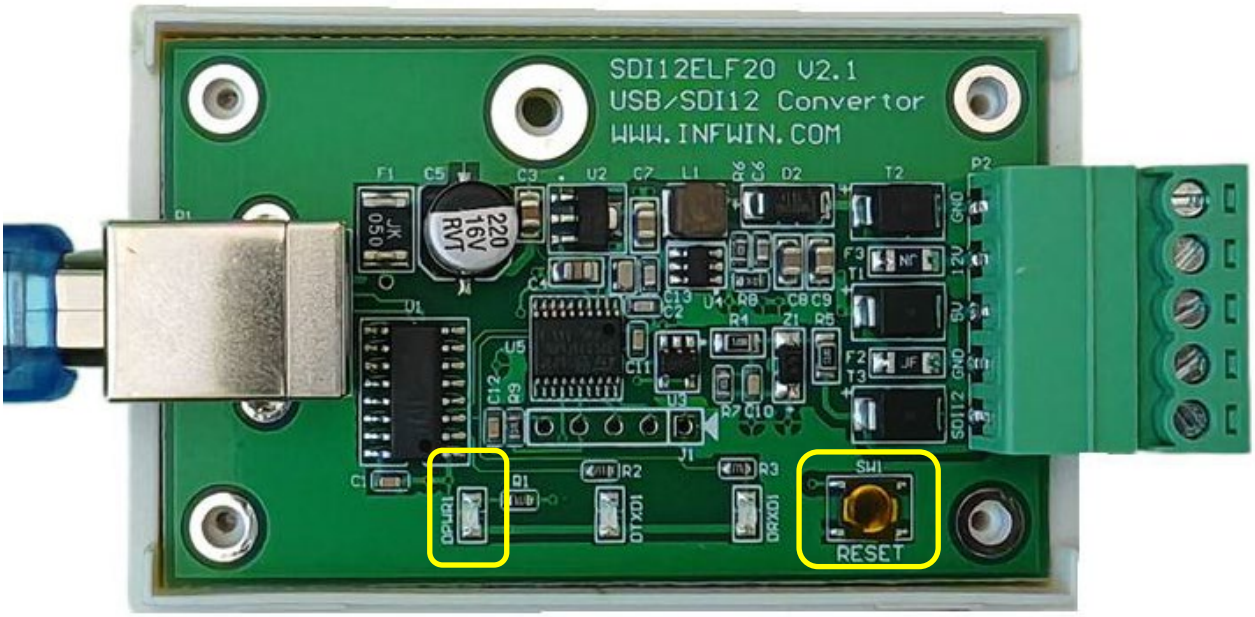
	Set current stopbits: AT+ STOPBITS =<STOPBITS><CR><LF>	**Response:** 1<CR><LF>OK<CR><LF> **Request:** AT+STOPBITS=1<CR><LF> **Response:** <CR><LF>OK<CR><LF>
SDI-12 BREAK Time	Query available break time: AT+SDI12BREAKTIME?<CR><LF> Query current break time: AT+ SDI12BREAKTIME=?<CR><LF> Set current break time: AT+SDI12BREAKTIME=<SDI12BREAKTIME><CR><LF>	Available <SDI12BREAKTIME>: 6000-255000us; default 20000us **Example:** **Request:** AT+SDI12BREAKTIME=?<CR><LF> **Response:** 20000<CR><LF>OK<CR><LF> **Request:** AT+SDI12BREAKTIME=20000<CR><LF> **Response:** <CR><LF>OK<CR><LF>
SDI-12 MARKING Time	Query available marking time: AT+ SDI12MARKINGTIME?<CR><LF> Query current marking time: AT+ SDI12MARKINGTIME =?<CR><LF> Set current marking time: AT+SDI12MARKINGTIME=<SDI12MARKINGTIME><CR><LF>	Available <SDI12MARKINGTIME>: 6000-255000us; default 10000us **Example:** **Request:** AT+SDI12MARKINGTIME=?<CR><LF> **Response:** 10000<CR><LF>OK<CR><LF> **Request:** AT+SDI12MARKINGTIME=10000<CR><LF> **Response:** <CR><LF>OK<CR><LF>

7.2 Restore the factory Settings

If user would like to restore the converter to factory Settings, or forgets the communication parameters and cannot communicate with the converter, please restore the converter to factory settings by method below:

（1）Send the ATR command to restore factory Settings, please refer to the previous chapter.

（2）When powering on the converter, press and hold the RESET button (as shown in the following figure) until the POWER indicator blinks and then release the button to restore factory Settings.



Appendix Ordering Information

Parameters	Code	Comments
Code 1: Product Series	SDI12ELF20	USB/SDI-12 Converter
Code 2: Version	A B	Standard Customize
Ordering Code Example： SDI12ELF20, USB / SDI-12 Converter, Standard Version. The ordering code is ： SDI12ELF20-A		

Appendix ASCII Table

Char	HEX	DEC	Char	HEX	DEC	Char	HEX	DEC	Char	HEX	DEC
nul	0	0	sp	20	32	@	40	64	'	60	96
soh	1	1	!	21	33	A	41	65	a	61	97
stx	2	2	"	22	34	B	42	66	b	62	98
etx	3	3	#	23	35	C	43	67	c	63	99
eot	4	4	$	24	36	D	44	68	d	64	100
enq	5	5	%	25	37	E	45	69	e	65	101
ack	6	6	&	26	38	F	46	70	f	66	102
bel	7	7	`	27	39	G	47	71	g	67	103
bs	8	8	(	28	40	H	48	72	h	68	104
ht	9	9	)	29	41	I	49	73	i	69	105
nl	0a	10	*	2a	42	J	4a	74	j	6a	106
vt	0b	11	+	2b	43	K	4b	75	k	6b	107
ff	0c	12	,	2c	44	L	4c	76	l	6c	108
cr	0d	13	-	2d	45	M	4d	77	m	6d	109
so	0e	14	.	2e	46	N	4e	78	n	6e	110
si	0f	15	/	2f	47	O	4f	79	o	6f	111
dle	10	16	0	30	48	P	50	80	p	70	112
dc1	11	17	1	31	49	Q	51	81	q	71	113
dc2	12	18	2	32	50	R	52	82	r	72	114
dc3	13	19	3	33	51	S	53	83	s	73	115
dc4	14	20	4	34	52	T	54	84	t	74	116
nak	15	21	5	35	53	U	55	85	u	75	117
syn	16	22	6	36	54	V	56	86	v	76	118
etb	17	23	7	37	55	W	57	87	w	77	119
can	18	24	8	38	56	X	58	88	x	78	120
em	19	25	9	39	57	Y	59	89	y	79	121
sub	1a	26	:	3a	58	Z	5a	90	z	7a	122

esc	1b	27	;	3b	59	[	5b	91	{	7b	123
fs	1c	28	<	3c	60	\	5c	92	|	7c	124
gs	1d	29	=	3d	61	]	5d	93	}	7d	125
re	1e	30	>	3e	62	^	5e	94	~	7e	126
us	1f	31	?	3f	63	_	5f	95	del	7f	127

Copyright and Trademark

Version Control

Date	Version	Comment	Updated by
2024-02-07	V1.0	Initial Creation	Sl51930