Smart Temperature Sensor

The Stevens Smart Temperature Sensor is a high precision digital temperature sensor that meets USGS guidelines.

The Smart Temperature Sensor features the ability to automatically sample temperature and calculate minimum, maximum, and average temperatures with a flexible sampling period and window.

The Smart Temperature Sensor is ideal for high-accuracy readings in water, soil, and air. SDI-12 output provides universal compatibility with any SDI-12-enabled data logger. The Digital Temperature Sensor features fully-potted components, robust PVC Type II housing, and marine-grade stainless steel, 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.

Every Smart Temperature Sensor can interface over the industry standard SDI-12 bus for low power applications, or use the standard SDI-12 command set over an RS-485 physical interface for applications that require long cable runs, many sensors, and high reliability.

The Smart Temperature Sensor features the ability to automatically sample temperature and calculate minimum, maximum, and average temperatures with a flexible sampling period and window.

The Smart Temperature Sensor features a durable 316 grade stainless steel tine for easy soil insertion (model 51126) or a blunt nose with pull-though loop hole that fits through standard 1″ conduit with 8″ factory bend corners (model 51163).

Operating Conditions

Parameter	Conditions	Unit	Min	Max
Supply voltage	Operation	V	6	18
	Programming	V	4.5	18
Temperature	Operation	°C	-40	85
	Storage	°C	-55	150
Maximum number of sensors per bus	RS-485			256
	SDI-12			10

Operating Characteristics

Parameter	Conditions	Unit	Min	Typ	Max
Accuracy	-5°C to +50°C	°C	-0.1		+0.1
	-40°C to +85°C	°C	-0.5		+0.5
Supply current	Average, SDI-12, one measurement per minute	uA		30	
	Average, RS485, one measurement per minute	uA		300	
	Idle, SDI-12, sampling period=1	uA		40	
	Idle, SDI-12, sampling period=0	uA		6.25	
	Peak	mA			25
Auto-sampling window		Samples	1		3600
Auto-sampling period		Seconds	1		

Basic Commands

Example	Typical Output	Description
?!	a	Check if the sensor is alive
0Aa!	a	Change sensor address from '0' to 'a'
aI!	a14STEVENSW_SVT01_1234	Identify the sensor
aM! aD0!	a0001 a+23.94	Read the current temperature, averaged over 10 seconds (default)
aM1! aD0!	a0004 a+25.22+24.33+26.42+0.37	Average, minimum, maximum, and standard deviation over the sampling window

About the Stevens implementation of SDI-12 over RS-485

The Smart Temperature sensor is fully compliant with SDI-12 v1.4. For SDI-12 over RS-485, there are two departures from the SDI-12 specification:

Configurable Baud Rate

The bitrate for SDI-12 over RS-485 defaults to 9600 baud, and can be configured to operate at 1200, 9600, or 115200 baud using the 'rs485' object described in the section, *Extended Commands*

More Permissive Wake Sequence

The full SDI spec requires the data line be held high for 12ms to generate a wake condition, and left idle for 8.33ms before sending the sensor address. To allow communication with a wider range of generic hardware, the sensor has looser requirements for wake on RS-485. The sensor will wake on any pulse sequence and the address can be sent after a 20.33ms delay, measured from the first pulse
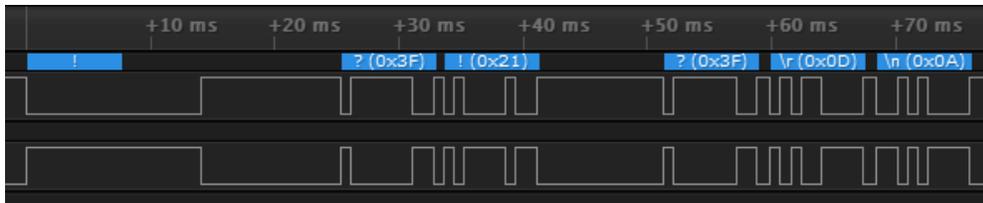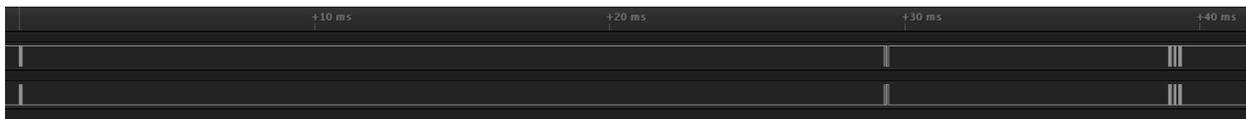

SDI-12 style break at 1200 baud
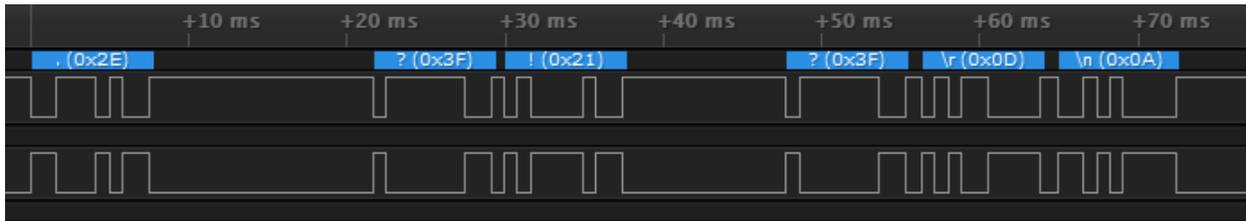
Wake character at 1200 baud
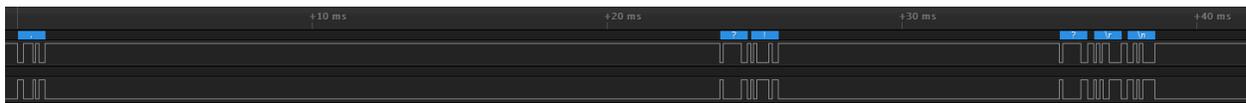
Wake character at 9600 baud

Wake character at 115200 baud

Ver. 3-Aug-17

How to get started – RS485

This guide will walk you through testing and configuring the sensor using a RS-485 adapter.

Install Teraterm
Download and install Teraterm from

Connect the sensor
Connect power, ground, and RS-485 A/B wires from the sensor to the RS-485 adapter.

Configure Teraterm
In Teraterm, connect to the CH340 device and configure it as 9600 baud 8n1 with 2ms character delay

Basic usage
Paste the following command into Teraterm:

```
.?!
```

To do that, select the above command in this document, copy it to the clipboard with <ctrl+c>, left click the Teraterm window to transfer focus, and right click in the black Teraterm window to paste the command. You should see the sensor respond with a single character:

```
0
```

Here's how that worked:

'.' – wake up the sensor. To wake a Stevens sensor on RS485, pulse the data line, for instance by sending a single character, and pause for one character duration before sending data. This pause is why the transmit delay is required in Teraterm.

'?' – this is a wildcard address. This can be used to identify and configure sensors when their address is not known.

'!' – this terminates the SDI command packet

'a' – the first character of the response is always the address of the sensor

If you didn't see any response, here are a couple things you can do to troubleshoot:
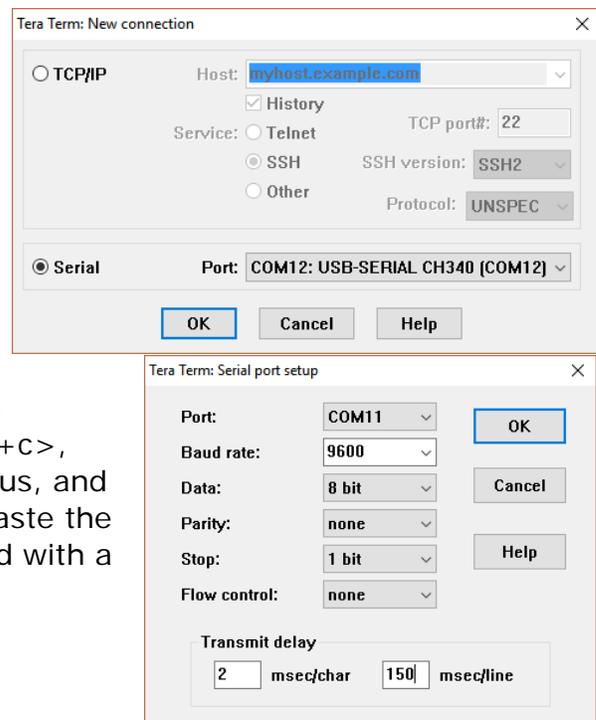
- Check the wiring. If RS-485 A and RS-485 B are reversed, the sensor won't respond. Also, check the power connections. If power is hooked up correctly, you should see the sensor draw very close to 300uA while in RS-485 mode
- Try configuring Teraterm for 1200, 9600, and 115200 baud. The factory default is 9600 baud, but it's possible to configure the sensor to operate at 1200 or 115200 baud.
- It's important to paste commands into the Teraterm window instead of typing them directly. That's because the sensor will time out and go back to sleep if there's more than a 100ms delay between characters.

Now let's try reading some data from the sensor:

```
.0R0!
```

The sensor will respond with its address, and a signed decimal representation of the current temperature:

```
0+23.94
```

Remember that the wake character '.' can only be used in RS-485 mode. SDI-12 requires an extended break condition to wake the sensor.

Ver. 3-Aug-17

Metadata Commands

Revision 1.4 of the SDI spec adds a set of commands to access metadata – descriptions of the return data including SHEF codes and units. The sensor fully supports the 1.4 spec. As a convenience, we extend the spec with the use of '*' as a wildcard character to return all data fields for a given measurement index.

```
aIM1!                                              a00003

aIM1_001!                         a,TN,DC,temperature (minimum);

aIM1_002!                         a,TX,DC,temperature (maximum);

aIM1_003!                         a,ZZ,DC,temperature (std dev);

aIM1_004!                                                    a

aIR0_*!                                 a,TW,DC,temperature;
```

Extended Commands

The SDI spec allows for manufacturer specific extensions to the basic command set. Stevens defines two extended commands: Xr and Xw. Both take an underscore-delimited set of parameters. The first parameter selects an object to read, write, or take action on. The second parameter, if present, provides additional information on what action to take.

As an example, let's configure the 'period' and 'window' objects to enable autosampling once a minute, with a five-minute averaging window.

We'll start by using the 'Xr' command to read the sampling interval from the 'period' object:

```
aXr_period!                                                      a1
```

Period is set to one, so sampling is enabled at a one second interval. Let's use the 'Xw' command to change the sampling period to 60 seconds:

```
aXw_period_60!                                                   a60
```

Let's read it back again to confirm that the value stuck:

```
aXr_period!                                                      a60
```

Finally, let's set the length of the averaging window to five samples:

```
aXw_window_5!                                                    a5
```

Some objects perform an action when invoked with 'Xw'. The 'factory' object will reset the device to a factory default state when invoked with a 'Xw' command:

```
aXw_factory!                                                     a
```

```
aXr_period!                                                      a1
```

Another common configuration option is the 'units' object. To change the units to Fahrenheit:

```
aXw_units_df!                                                    aDF
```

Ver. 3-Aug-17

Extended Command Objects

Object	R/W	Description
address	R/W	SDI address (same as configured with the 'A' command)
period	R/W	Sampling period in seconds (default: 1)
window	R/W	Averaging window (default: 10)
units	R/W	Units (dc, df, default: dc)
rs485	R/W	RS-485 bitrate (1200, 9600, 115200, default: 9600)
max	R	Maximum temperature seen over the lifetime of the sensor
min	R	Minimum temperature seen over the lifetime of the sensor
firmware	R	Firmware revision
hwid	R	Unique hardware identifier derived from the MCU silicon ID
cycles	R	Number of power cycles over the lifetime of the sensor
soft	W	Perform a soft power cycle. Takes no additional parameters
factory	W	Reset all writable configuration objects to the factory defaults. Takes no additional parameters
bootloader	W	Enter the bootloader. Takes no additional parameters

Auto-sampling

The Smart Temperature Sensor samples temperature at regular intervals and calculates a sliding window average. In addition to the smoothing and antialiasing benefits that averaging provides, consolidating data into key statistics can reduce the total number of bus transactions, saving power and making efficient use of limited bus capacity.

To configure the auto-sampling feature, there are two key parameters: sampling period and sampling window.

Sampling Period

The 'period' object defines the sampling period in seconds

If 'period' is set to a non-zero value, that's the interval in seconds at which the sensor will sample the temperature

If 'period' is set to zero, auto-sampling will be disabled and temperature data will not be updated

Sampling Window

The 'window' object defines the number of samples the sensor will store in the sampling buffer. 'M0!' always returns an average over the sampling buffer.

Bootloader

The Smart Temperature Sensor includes a bootloader that enables in-field firmware updates over RS-485, for bugfixes and feature updates. Further information may be available on the Stevens website.