**GeoPrecision GmbH**
Am Dickhäuterplatz 8
D-76275 Ettlingen
Germany

T +49(0)7243-505419-0
F +49(0)7243-505419-9
E info@geoprecision.de

www.geoprecision.de
www.geo-precision.com

# Two-Wire to SDI12 Converter

Manual

Table of Contents

# 1. Version History

21.09.18: Initial

## 2. Introduction

The “Two-Wire to SDI12 Converter” is a configurable universal interface to connect up to 48 “Two-Wire sensors” to your SDI12 infrastructure. The converter acts as a SDI12-Sensor (slave) and can be connected to every compatible SDI12-recorder or host-interface.

This document guides through the required steps of

- Connecting the sensors.
- Connecting the SDI12-cable to a host.
- Configuration and communication via SDI12-commands.

➔ Delivery of the converter in combination with a **Thermistorstring**:
In this case the device is already configured for correct operation. Please refer directly to the chapters “Connection” and the measurement-command (and examples) at “List of SDI-commands”.

# 3. Specification

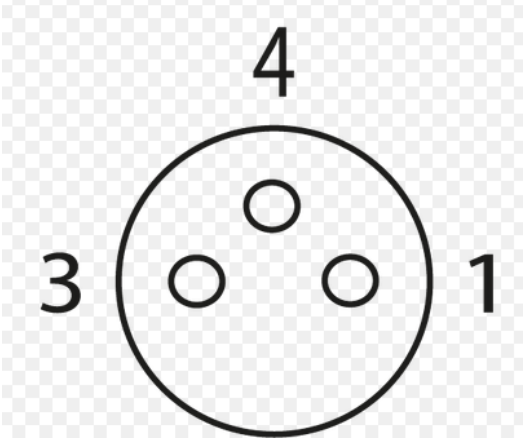

- SDI-12 Standard (see http://www.sdi-12.org for more information)
- 6V … 14V supply voltage
- Power consumption during measurement: >60 mA
- Power consumption idle: 2 mA
- Overvoltage protection by TVS-Surge Absorber 400 Watt
- Up to 48 Sensors can be connected to one Interface
- IP69 enclosure
- Operating Temperature Range: - 40°C … + 85°C

# 4. Connection

**Two-Wire sensors:**

Electrical interface: M8 industrial 3 pole connector.

The following schematic shows the pin-usage at sensor-side (cable with male plug).




Brown (1): Data

Black (4): Ground (GND)

Blue (3): not connected

The connection “Two-Wire-Sensors” follows the “BUS-topology”. Each sensor is electrically connected to the same Data- and GND-line!

**SDI12-connection:**

Brown: Supply (8 to 14 Volt, >60 mA cont.)

Black: GND

Blue: Data

# 5. Communication

In case of a preconfigured converter (in combination with a Thermistorstring) directly use the M- and D-commands to start and read a measurement.
Also have a look at the example-code for measurement.

→The converter must stay powered until all operations are finished. In case of a power loss (e.g. between the M- and the D-command) you have to repeat the whole procedure.
→A **power up wait time** of at least 500 ms is required!

## 5.1 Basic SDI12 commands

The command set is based on extended SDI12 (V1.1) command set.
→'**a**' represents the SDI-address, this might also be '?' (as wild card). Default SDI-address: '**1**'.

**aAn!**
Change address from 'a' to 'n'.
**aI!**
Identify Node.
**aM! / aM0!**
Start measure of **all** configured sensors. All values are stored at the internal cache. This must be the "initial" M-command!
**aD! / aD0!**
This will read up to 4 measures from the first „M" / "M0"- command.
**aMn!**
Prepare the next values (up to 4) for output. 'n' must be between 1 and 12.
**aDn!**
This will read the next measures (up to 4) from the preceding „Mn"- command. 'n' must be between 1 and 12, corresponding to the number 'n' of the M-command!.

**Error codes:**

-98.00 : communication error or sensor missing

-99.00 : channel not activated. E.g.: channel 1 and 3 are configured to read values from a sensor, channel 2 is deactivated. The second value, requested by “D”-command is “-99.00”.

## 5.2 Extended SDI12 (X)-commands

The extended commands are not compatible to the SDI12-specification. They can be used to configure the converter for a set of new Two-Wire sensors.

→During execution of X commands the power of the interface must not be interrupted!

→This is an overview of available X commands, for usage and detailed information see next chapter!

**aXS!**

Scan Two-Wire bus for the number connected sensors.

**aXUn!**

Read detailed information from the specific sensor ‘**n**’.

**aXCk,n,t!**

Allocate channel ‘**k**’ with sensor ‘**n**’ of type ‘**t**’.

**aXCk,0,0!**

Deactivate channel ‘**k**’.

## 5.3 Example commands for measure

The following commands and responds will demonstrate how to measure and read values from the converter. The output depends on the connected and configured Two-Wire sensors.
“>>” marks the command to the sensor, “<<” is the response. Each Command and response ends with <CR><LF>. Default address “1” is used.

```
>>   "1M0!"
<<   "1064"
<<   "1"
>>   "1D0!"
<<   “1[Value 0-3]”
>>   "1M1!"
<<   "1064"
<<   "1"
>>   "1D1!"
<<   “1[Value 4-7]”
>>   "1M2!"
<<   "1062"
<<   "1"
>>   "1D2!"
<<   “1[Value 8-9]”
```

# 6. Two-Wire sensor configuration (X-Command)

To configure the converter for correct measurement and output of the connected Two-Wire sensors, a special not-SDI-conform X-command is used. The configuration is stored in non-volatile memory, so it will resist a power loss of the device. Changing sensors requires a reconfiguration!

To apply SDI12-Commands to the converter you have to connect it to a power-supply and a PC-Interface for your preferred SDI12-Terminal (e.g. **SDI Win** or **SDI Term**). Or your data-recorder provides a command line to send SDI12 commands directly to its interface.
It must be continuously powered during execution of all steps!

The following shows the "**basic**" assignment of a sensor to a channel. For **Thermistorstrings** a special, position related assignment is shown afterwards.

→">>" marks the command to the sensor, "<<" is the response. Each Command and response ends with <CR><LF>. Default address "1" is used.

## 6.1 "Basic" sensor to channel assignment

Step 1)
Scan Two-Wire bus for connected sensors.
>>**1XS!** <CR><LF>
<<**1nn** <CR><LF>
The response is the number of connected sensors (nn).

Step 2)
Assign a specific sensor to a channel of the converter.
>>**1XCk,n,t!**<CR><LF>
Allocate channel '**k**' with sensor '**n**' of type '**t**'. Select the type from "Table of sensor types" below.

→'**n**' represents a specific sensor, counting from '1' to the number of connected sensors ('**nn',** see Step 1).

<<**1kk,nn,ttt**<CR><LF>

Remark: To deactivate a channel (k) of the converter (e.g. if it's unused, where '**k**' is the channel number) send: >>**1XCk,0,0!**<CR><LF>

Example:

Sensor 2 (e.g. a "TNode EX") should be assigned to Channel 1 (first value of data output). The complete command for this operation:

| | |
|---|---|
| >> | "1XC1,2,140!" |
| << | "101,02,140" |

Step 3)

Repeat Step 2 for all connected sensors by adjusting channel (k) and sensor number (n).

## 6.2 "Position related" sensor to channel assignment for Thermistorstrings

A Thermistorstring can contain up to 48 separate temperature-sensors. In most cases it is useful to assign the **first sensor** in the string to **channel 1** of the converter, **second sensor** to **channel 2** and so on.

The position of each sensor within the string is factory-programmed to the sensor itself.

A complete sample-code for a 3 sensor Thermistorstring is attached.

Step 1)

Scan Two-Wire bus for connected sensors.

>>**1XS!** <CR><LF>

<<**1nn** <CR><LF>

The response is the number of connected sensors (nn).

Step 2)

Read positioning information from one specific sensor.

>>**1XUn!**<CR><LF>

‘**n**’ represents a specific sensor, counting from ‘1’ to the number of connected sensors (see Step 1: ‘**nn’**).

<<1**I**nn,**A$**aaaa,**T**tttt,**P**pp<CR><LF>

‘**I**’: Followed by the sensor number (nn).

‘**A**’: 2Wire address of the sensors (aaaa).

‘**T**’: Type of the sensor (ttt). Remark: This is not the type required at the next steps!

‘**P**’: Sensor position (pp).

Step 3)

Repeat Step 2 by incrementing ‘n’ at the request-command till the number of connected sensors is reached. Create a table with the sensor-number (n) and the related sensor-position (pp), e.g.:

| Sensor Number (n) | Sensor Position (pp) |
|---|---|
| 1 | 3 |
| 2 | 1 |
| 3 | 2 |

Step 4)

Allocate channel ‘**k**’ with sensor ‘**n**’ of type ‘**t**’. Select the type from “Table of sensor types” below.

Use the information from table above (step 3) to assign a channel to its corresponding sensor.

>>**1XCk,n,t!**<CR><LF>

Step 5)

Repeat Step 4 for each sensor and channel, like channel 1 (k=1) to sensor number 2 (n=2), channel 2 to sensor number 3 and channel 3 to sensor number 1.

## 6.3 Example of positioning a Thermistorstring

A Thermistorstring containing 3 sensors of type 140 (TNode EX) should be assigned to the corresponding channel numbers of the converter. This is the full code for all 3 sensors. Default address “1” is used.

```
>>   "1XS!"
<<   "103"
>>   “1XU1!”
<<   “1I01,A$3DA1,T0256,P03”
>>   “1XU2!”
<<   “1I02,A$3D41,T0256,P01”
>>   “1XU3!”
<<   “1I03,A$3AE1,T0256,P02”
>>   “1XC1,2,140!”
<<   “101,02,140”
>>   “1XC2,3,140!”
<<   “102,03,140”
>>   “1XC3,1,140!”
<<   “103,01,140”
```

## 6.4 Table of sensor-types

| Sensor-Type | Description |
|---|---|
| 140 | TNode EX (outdated Version) |
| 141 | Counter 32Bit |
| 142 | Keller-LDS (Pres./Temp.) |
| 143 | Baro (Pres./Temp.) |
| 144 | SHT (%rh/Temp.) |
| 145 | TNode |
| 147 | TNode EX + TNode HD |