	Manual	**Version 1.0** 16.01.2022
		Editor D. Bergmann

TSC12 / TSC12-ISO

Version	Date	Editor	Description
1.0	16.03.2021	Bergmann	Initial Release

1. TSC12 / TSC12-ISO

1.1. GENERAL DESCRIPTION

The TSC12 temperature scanners are capable of measuring 12 thermocouple voltage signals simultaneously, featuring high accuracy and a minimal offset drift. Each device can be individually customized according to customer specifications.

All common thermocouple socket types (K, T, J, B, E, N, R, S, V*, W*) are supported. Depending on the socket type, measurable temperatures stretch from -210°C to 1798°C.

The data is transmitted as ASCII text in the unit degrees Celsius [°C]. The transmission rate can be set in the range between 1 and 100Hz.

Power for TSC devices equipped with USB or CAN interface is supplied via USB-, respectively CAN-port itself. For the LAN interface version, an external power supply (8-24V, 0.5A) has to be connected to the device.

Both standard and -CAN TSC devices provide an USB interface for comfortable configuration. When connected via USB the temperature scanner identifies itself to the host PC as virtual COM port. Thus, any software supporting serial protocols can be used for communication. The LAN-version uses TCP-IP protocol for data transmission and configuration. A direct connection can be set up via **Telnet (Port 10001)**.

A recording software and an example program in LabVIEW (source code) are shipped with the device. For devices with CAN bus interface a DBC-file is included in the shipment.

1.2. DIFFERENCE BETWEEN TSC12 AND TSC12-ISO

Feature	TSC12	TSC12-ISO
Galvanic channel isolation	No (input protection +/-40V)	each channel individually isolated
Analog-digital converter	19bit	24bit
Max. scanrate	50Hz	100Hz
Housing dimensions	130 x 55 x 95 mm (B x H x T)	130 x 55 x 170 mm (B x H x T)

1.3. SERIAL INTERFACE

Command	Function	Answer
`EE_LOAD`	Load calibration data from EEPROM	`#EEPROM:loaded`
`EE_SAVE`	Save calibration data to EEPROM	`#EEPROM:saved`
`*IDN?`	Read device ID	`TYPE `**`PSC8-USB`**` VERSION `**`1.0`**` SERNUM `**`#SN31xxxxx`**
`RATE x`	Set sample rate for streaming mode range x = 10...5000 [ms] default: 1000[ms] ~> 1[Hz]	`#Rate=x ms` `#Error: Rate-Range`
`RATE?`	Read sample rate	`#Rate=x ms`
`RATE 0`	Activate request and trigger mode Actual values are read only after manual command „?" is sent	`#Request-Mode active`
`?`	Read actual values (request-mode only)	
`*RST`	Reset scanllist settings	`#RESET`

`SCAN_A x`
`SCAN_B x`
`SCAN_C x`
	Defines a scanlist (channel selection)
	Binary, each bit represents one channel	

`FILTER x`	Activate exponential filter 0 = deactivated; >0 = filter range in ms	`#Filter=x`

`TC x K`	Set thermocouple type of channel x to type K (available: K, T, J, B, E, N, R, S, V*, W*)
	x = -1: set thermocouple type of all channels
		`#TC x K`
		
		`#TC K K K K K K K K K K K K`

`TC? x`	Read thermocouple type of channel x	`#TC x K`

TSC12 / TSC12-ISO

Command	Function	Answer
	x = -1: read thermocouple type of all channels.	`#TC K K K K K K K K K K K K`
`TC_OFS x`	Set cold junction temperature offset range x = -7.95...8 [K] default: 0.7 / according to calibration	`#TC_OFS x`
`TC_OFS?`	Read cold junction temperature offset	`#TC_OFS x`
`tx 1`	Start streaming mode	`#TX ON`
`tx 0`	Stop streaming mode	`#TX OFF`

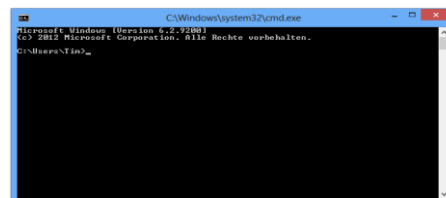| **- for CAN bus version only -**		
`CAN_ID x`	Set CAN-ID	`#OK`
`CAN_IT x`	Set interface x = 0: normal (11bit, CAN 2.0A) x = 1: extended 23bit (23bit, CAN 2.0B)	`#OK`
`CAN?`	Read actual CAN configuration	`#ID:0x[...]_Speed:[baud]_IDT:[0,1]`
`CAN_SPEED x`	Set CAN bus rate 0: 125 kBaud 1: 250 kBaud 2: 500 kBaud 3: 1 MBaud	`#OK`

Each command is terminated by a line break (CR, LF or CR+LF). The sensor enumeration of all devices starts at 1.

1.4. TELNET TCP COMMUNICATION

Establish Telnet connection on TSP Port 10001

Install or activate telnet (on Windows: enable telnet feature, see https://social.technet.microsoft.com/wiki/contents/articles/910.windows-7-enabling-telnet-client.aspx)

Open a terminal (on Windows: cmd.exe)

Enter "`telnet 192.168.1.200 10001`" (use the TSC's IP. **The communication port is 10001**)

1.5. DATA TRANSFER MODES

A Software trigger mode

Type "`rate 0`" to enter trigger mode (followed by `<enter>`)

Type "?" (followed by `<enter>`) → the TSC sends the most recent data in CSV format

TCP Command	Answer
rate 0	#Request-Mode active. Send '?'
?	21.1200 22.2422 -10.2350 0.0210 -12.7820 ...

B External trigger mode

Type "`rate 0`" to enter trigger mode (followed by `<enter>`)

Connect external trigger to the scanner → on every trigger signal the TSC sends the most recent data in CSV format

TCP Command	Answer
rate 0	#Request-Mode active. Send '?'
trigger signal	21.1200 22.2422 -10.2350 0.0210 -12.7820 ...

C Streaming mode

For continuous output set the output rate to any value from 10 to 5000[ms], e.g. "`rate 100`"

TCP Command	Answer
rate 200	#rate=200ms
tx 1	#TX ON
	21.1200 22.2422 -10.2350 0.0210 -12.7820 ...
	21.1200 22.2422 -10.2350 0.0210 -12.7820 ...
	21.1200 22.2422 -10.2350 0.0210 -12.7820 ...
	21.1200 22.2422 -10.2350 0.0210 -12.7820 ...
	21.1200 22.2422 -10.2350 0.0210 -12.7820 ...
	...

Data Format

The output data sent in CSV format. Values are tab "`/t`" separated and lines are terminated by a new line and a carriage return character "`/n/r`".

Scanlist

To choose the channels that are to be sent use the `SCAN` command.

SCAN_A sets the first 8 channels SCAN_B the following 8 etc. The adjacent number is the 8-bit representation of the 8 channels (each bit one channel)

→ SCAN_A 3: only channel 1 and 2 are read (3 = 1 1 0 0 0 0 0 0)

TCP Command	Answer
rate 200	#rate=200ms
tx 1	#TX ON
	21.1200 22.2422 -10.2350 0.0210 -12.7820 ...
	21.1200 22.2422 -10.2350 0.0210 -12.7820 ...
	21.1200 22.2422 -10.2350 0.0210 -12.7820 ...
	21.1200 22.2422 -10.2350 0.0210 -12.7820 ...
	21.1200 22.2422 -10.2350 0.0210 -12.7820 ...
	...
tx 0	#TX OFF
SCAN_A 3	
tx 1	#TX ON
	21.1200 22.2422
	21.1200 22.2422
	21.1200 22.2422
	21.1200 22.2422
	21.1200 22.2422
	...
*RST	#RESET
	21.1200 22.2422 -10.2350 0.0210 -12.7820 ...
	21.1200 22.2422 -10.2350 0.0210 -12.7820 ...
	21.1200 22.2422 -10.2350 0.0210 -12.7820 ...
	21.1200 22.2422 -10.2350 0.0210 -12.7820 ...
	21.1200 22.2422 -10.2350 0.0210 -12.7820 ...
	...

1.6. VB .NET EXAMPLE CODE

```
' TSC12-Example application -- Continuous mode
' Opens a TCP network stream and gathers the data continuously.
' -> while running, hit any key to exit

Imports System
Imports System.IO
Imports System.Net
Imports System.Net.Sockets
Module TSC_streaming
```

```
Sub Main()
Dim IP As String = "192.168.1.102" ' Enter IP address here
Dim Client = New TcpClient(IP, 10001)
Dim values() As Double
Dim d As Double
Dim strArray() As String
If Client.Connected Then
Dim ns = Client.GetStream()
Dim SR = New StreamReader(ns)
Dim sw = New StreamWriter(ns)
Dim line As String
Dim quitNow = 0
Dim count = 0
sw.WriteLine("") ' If there was something in the send-buf fer, we can clear that with
one linefeed
sw.WriteLine("RATE 300") ' Command to set the scanrate to 300ms
sw.WriteLine("TX 1") ' Command to start the streaming mode
' Add commands if needed
sw.Flush()
While (Not Console.KeyAvailable) ' every key pressed exits this demo
line = SR.ReadLine()
Console.WriteLine(line)
strArray = line.Split(vbTab)
If strArray.All(Function(number) Decimal.TryParse(number, d)) Then
values = Array.ConvertAll(strArray, Function(c As String) Val(c)) 'convert string to
doubles for further use
' do something with your values here...
End If
End While
Console.Write("Just as example: last data[0] was: ")
Console.WriteLine(values(0))
Console.WriteLine("Closing connection and exiting demo")
sw.WriteLine("TX 0") ' Command to stop the streaming mode
sw.Flush()
Threading.Thread.Sleep(3000)
SR.Close()
sw.Close()
ns.Close()
Client.Close()
End If
End Sub

End Module
```

1.7. PIN ASSIGNMENT (M8-CONNECTOR)

Standard version:
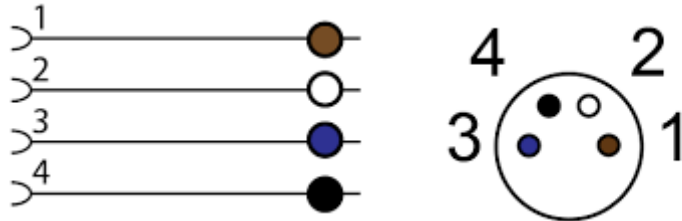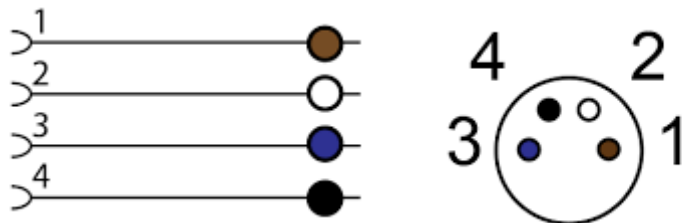

Pin	Function	Cable colour
1	+ Supply	brown
2	not used	white
3	- Supply (GND)	blue
4	not used	black

Table 1: Pin assignment standard version

Trigger version:

Pin	Function	Cable colour
1	+ Supply	brown
2	- Trigger (GND)	white
3	- Supply (GND)	blue
4	+ Trigger	black

Table 2: Pin assignment trigger version

CAN bus version:
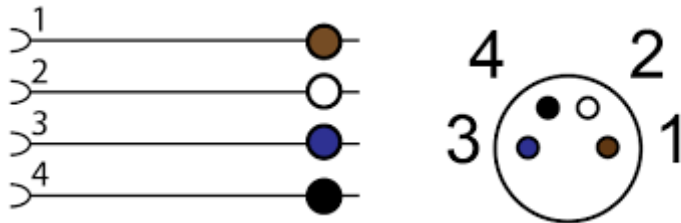
Pin	Function	Cable colour
1	+ Supply	brown
2	CAN low	white
3	- Supply (GND)	blue
4	CAN high	black

Table 3: Pin assignment CAN-Bus version

BNC Trigger Connector Pinout

Pin	Function	Pin
1	+ Trigger	Center pin
2	- Trigger (GND)	Shield

Power Specification

Connector	Pins	Voltage range
Power	1-3	7-24V DC / 100mA
Trigger	2-4	3-24V

Figure 1: Power connector pinout

1.8. TECHNICAL SPECIFICATIONS

Available Thermocouple Sockets			
Typ	min	max	Unit
K	-200	1372	°C
J	-210	1200	°C
N	-200	1300	°C
E	-200	1000	°C
T	-200	400	°C
R	-50	1768	°C
S	-50	1768	°C
B	95	1798	°C
V*	-140	140	mV
Optional: White copper sockets for all types (K, J, N, E, T, R, S, B, V*, W*)			

Accuracy and Sample Rates	
Accuracy	+/- 0.5K (with copper sockets 0.5% FS)
Sample rate per channel	1-100Hz
Resolution (internal)	24-bit

Power Supply	
TSC12	via USB
TSC12-LAN/-CAN	8-24V, 0.5A

Environmental Conditions	
Temperature	5°C...50°C
Humidity	0...95%, non-condensing

Dimensions	
Housing	130 x 55 x 95/170 mm (B x H x T) TSC-ISO: length 170mm

Driver and Software
Virtual COM-Port-Driver
Configuration software
LabVIEW-example program as source code

Supported Operating Systems
Windows 7, 8, 10, 11,Linux

Options
All TSC systems can optionally be equipped with CAN bus, LAN or RS232