TSC12 Mehrkanal-Temperaturscanner

Eigenschaften:

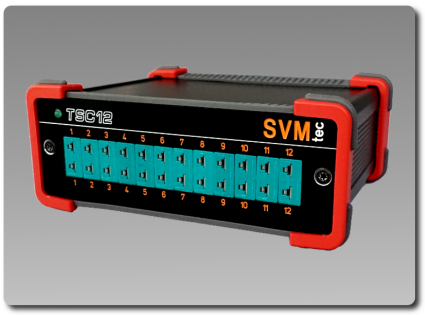
- Simultane Erfassung **mehrerer Thermoelement-spannungs Signale**
- Typen: **K, T, J, B, E, N, R, S**
- Nichtlinearität & Hysterese **max. +/- 1K**
- Datenübertragung und Stromversorgung **kombiniert** via **USB-Anschluss**
- Optional mit **CAN Bus, LAN oder RS232** lieferbar
- Inkl. **Software und Treiber für LabVIEW**

Kundenspezifische Anpassungen:

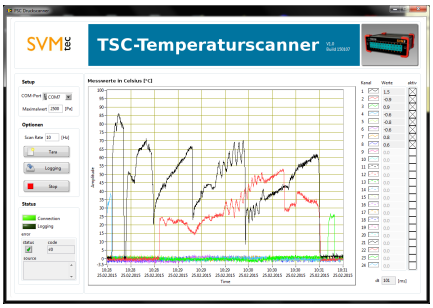
- Auswahl des **Thermoelementanschluss-Typs** mit verschiedenen Messbereichen
- Auswahl der der **Schnittstellen zur Daten-kommunikation**

Anwendungen:

- Mehrkanal Temperaturmessungen in der **Automobil Industrie**
- Temperaturmessung in **Kühlern und thermischen Prozessen** mit vielen Temperaturmessstellen



TSC12



Software

SVMtec GmbH | Bergnelkenstraße 7 | 70563 Stuttgart | Deutschland | www.svm-tec.de

Kontakt: +49 (0)711 391867-0 | info@svm-tec.de

Stand: 01.08.2018

Temperaturscanner Optionen

Verfügbare Thermoelement Anschlüsse		
Typ	**min °C**	**max °C**
K	-270	1300
J	-210	1200
N	-270	1300
E	-270	1000
T	-270	400
R	-50	1768
S	-50	1768
B	0	1800
Genauigkeit und Abtastraten		
Nichtlinearität & Hysterese	max +/- 1K	
Auflösung	19-bit	
Abtastrate pro Kanal	1-100Hz	
Stromversorgung		
TSC12	über USB	
TSC12-LAN/-CAN	7-24V, 1A	
Umgebungsbedingungen		
Temperatur	5°C...50°C	
Luftfeuchtigkeit	0...95%, nicht kondensierend	
Maße		
Gehäuse	130 x 55 x 95 mm (B x H x T)	
Treiber und Software		
Virtueller COM-Port-Treiber		
Konfigurationssoftware		
LabVIEW-Beispielprogramm als Sourcecode		
Unterstützte Betriebssysteme		
Windows XP, 7, 8, 10, Linux		
Optionen		
Alle TSC-Systeme sind optional mit CAN Bus, LAN oder RS232 lieferbar		

SVMtec GmbH | Bergnelkenstraße 7 | 70563 Stuttgart | Deutschland | www.svm-tec.de

Kontakt: +49 (0)711 391867-0 | info@svm-tec.de
Stand: 01.08.2018

Allgemeine Beschreibung

Die Temperaturscanner der TSC-Reihe eignen sich zur simultanen Überwachung von 12 Temperaturmessstellen mit hohen Anforderungen bzgl. der Genauigkeit.

Es werden alle gängigen Thermoelement Typen (K, T, J, B, E, N, R, S) unterstützt. Je nach Typ können Temperaturen von -270°C bis +1800°C verarbeitet und überwacht werden.

Der Anschlusstyp kann individuell nach Kundenvorgabe gewählt werden, wobei alle Kanäle standardmäßig mit dem gleichen Sensor-Typ bestückt werden.

Die Datenübertragung erfolgt als ASCII-Text in der Einheit Grad Celsius [°C]. Über ein einfaches Protokoll kann die Übertragungsrate im Bereich zwischen 1 und 100Hz eingestellt werden. Die ASCII Daten lassen sich in Excel importieren und bearbeiten.

Die Stromversorgung der Druckscanner mit USB/CAN-Schnittstelle erfolgt über den USB bzw. CAN-Anschluss. Für die Variante mit LAN-Schnittstelle ist ein externes Netzteil notwendig.

Geräte der TSC Standard und -CAN Variante sind serienmäßig mit einem USB-Anschluss ausgestattet, über welchen sie sich leicht konfigurieren lassen. Über USB mit dem Messrechner verbunden meldet sich der Temperaturscanner als virtueller COM-Port im System an. Damit kann er mit jeder Software verwendet werden, die ein serielles Protokoll unterstützt. Ein Beispielprogramm zur Verwendung mit LabVIEW wird mitgeliefert.

Im Lieferumfang der TSC-CAN Geräte befindet sich darüber hinaus eine passende DBC-Datei.

Auf Nachfrage sind individuelle Anpassungen möglich:

- Auswahl von verschiedenen Anschlusstypen in einem Gerät
- Kombination mehrerer Schnittstellen
- Einbau einer Trigger- oder Alarmfunktion bei bestimmten Temperaturen

SVMtec GmbH | Bergnelkenstraße 7 | 70563 Stuttgart | Deutschland | www.svm-tec.de

Kontakt: +49 (0)711 391867-0 | info@svm-tec.de
Stand: 01.08.2018

Serielle Schnittstelle

Befehl	Funktion	Antwort
EE_LOAD	Kalibrierdaten aus EEPROM laden	#EEPROM:loaded
EE_SAVE	Kalibrierdaten in EEPROM speichern	#EEPROM:saved
*IDN?	Gerätekennung abfragen	TYPE **PSC8-USB** VERSION **1.0** SERNUM **#SN31xxxxx**
RATE x	Abtastrate definieren für Streaming Modus Bereich x = 10...5000 [ms] Standard: 1000[ms] ~> 1[Hz]	#Rate=x ms #Error: Rate-Range
RATE 0	Abfrage- und Trigger-Modus aktivieren Durch Senden von „?“ wird der aktuelle Wert ausgegeben	#Request-Mode active
?	aktuelle Werte anfordern (nur im Request-Mode)	
*RST	Scanliste resetten	#RESET
SCAN_A x SCAN_B x SCAN_C x	Scanlist (Kanalauswahl) definieren Binär, jedes Bit steht für einen Kanal	
FILTER	exponentiellen Filter aktivieren 0 = deaktivert; >0 = Bereich des Filters in ms	#Filter=x
TC x K	Thermoelement Typ von Kanal x auf Typ K setzen (verfügbar: K, T, J, B, E, N, R, S) x=-1 kann verwendet werden, um alle Kanäle auf den gleichen Typ zu setzen	#TC x K #TC K K K K K K K K K K K
TC_OFS x	Offset des Kaltstellen-Sensors setzen Bereich: x= -7,95 ...8 [K], ab Werk voreingestellt	#TC_OFS x
TC_OFS?	Offset des Kaltstellen-Sensors auslesen	#TC_OFS x
tx 1	Streaming Modus starten	#TX ON
tx 0	Streaming Modus stoppen	#TX OFF

- nur bei TSC-CAN Variante -		
CAN_ID x	CAN-ID zuweisen	#OK
CAN_IT x	Interface setzen x = 0: Normal (11bit, CAN 2.0A) x = 1 Extended 23bit (23bit, CAN 2.0B)	#OK
CAN?	CAN-Konfiguration abfragen	#ID:0x[...]_Speed:[baud]_IDT:[0,1]
CAN_SPEED x	0: 125 kBaud 1: 250 kBaud 2: 500 kBaud 3: 1 MBaud	#OK

Ein Befehl wird immer mit einem Zeilenumbruch (CR oder LF oder CR+LF) terminiert. Die Sensornummerierung beginnt in allen Fällen mit der Nummer „1“.

SVMtec GmbH | Bergnelkenstraße 7 | 70563 Stuttgart | Deutschland | www.svm-tec.de

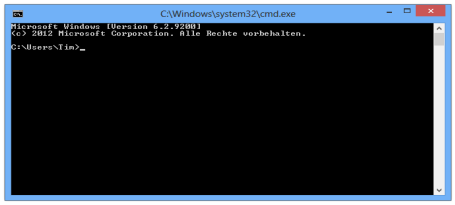
Kontakt: +49 (0)711 391867-0 | info@svm-tec.de
Stand: 01.08.2018

Telnet TCP-Kommunikationsbeispiel

Telnet Verbindung aufbauen

Telnet installieren oder aktivieren (Windows: https://social.technet.microsoft.com/wiki/contents/articles/910.windows-7-enabling-telnet-client.aspx)

Terminal öffnen (Windows: cmd.exe)



“`telnet 192.168.1.200 10001`” eingeben (die IP des TSC’s benutzen. Der Kommunikations **Port** ist **10001**)

Datenübertragunsmodi

A Software Trigger Modus

“`rate 0`” eingeben um in Trigger Modus zu gelangen (bestätigen mit `<enter>`)

“?” eingeben (bestätigen mit `<enter>`) → der TSC sendet die neuesten Messdaten im CSV-Format

TCP Befehl	**Antwort**
`rate 0`	`#Request-Mode active. Send '?'`
`?`	`21.1200 22.2422 -10.2350 0.0210 -12.7820 ...`

B Externer Trigger Modus

“`rate 0`” eingeben, um in Trigger Modus zu gelangen (bestätigen mit `<enter>`)

Externen Trigger mit Scanner verbinden → bei jedem Triggersignal sendet der TSC die neuesten Messdaten im CSV-Format

TCP Befehl	**Antwort**
`rate 0`	`#Request-Mode active. Send '?'`
Trigger Signal	`21.1200 22.2422 -10.2350 0.0210 -12.7820 ...`

C Streaming Modus

Um die Messwerte koninuierlich auszugeben, muss die Rate auf einen Wert zwishcen 10 und 5000[ms] gesetzt werden, e.g. “`rate 100`”

TCP Befehl	**Antwort**
`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 ...`
	`...`

SVMtec GmbH | Bergnelkenstraße 7 | 70563 Stuttgart | Deutschland | www.svm-tec.de
Kontakt: +49 (0)711 391867-0 | info@svm-tec.de
Stand: 01.08.2018

Datenformat

Die Ausgabedaten werden im CSV-Format gesendet. Die Messwerte sind durch Tabs “\t” getrennt und die Zeilen werden per new line und carriage return Zeichen beendet “/n/r”

Scanliste

Der SCAN Befehl wird verwendet, um die Kanäle auszuwählen, die gesendet werden sollen.

SCAN_A setzt dabei die ersten 8 Kanäle SCAN_B die darauf folgenden 8 usw. Die nebenstehende Zahl repräsentiert mit ihren 8 Bits die 8 Kanäle (jedes Bit ein Kanal)

→ SCAN_A 3: nur Kanal 1 und 2 werden gelesen und gesendet (3 = 1 1 0 0 0 0 0 0)

TCP Befehl	**Antwort**					
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	...
	...					

SVMtec GmbH | Bergnelkenstraße 7 | 70563 Stuttgart | Deutschland | www.svm-tec.de
Kontakt: +49 (0)711 391867-0 | info@svm-tec.de
Stand: 01.08.2018

VB .net Beispielcode

```
' 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
```


SVMtec GmbH | Bergnelkenstraße 7 | 70563 Stuttgart | Deutschland | www.svm-tec.de

Kontakt: +49 (0)711 391867-0 | info@svm-tec.de

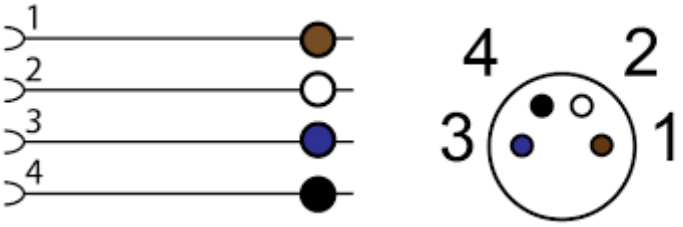
Stand: 01.08.2018

Steckerbelegung (M8-Stecker)

Standardversion:

Pin	Funktion	Kabel Farbe
1	+ Versorgung	braun
2	nicht verwendet	weiß
3	- Vers. (GND)	blau
4	nicht verwendet	schwarz

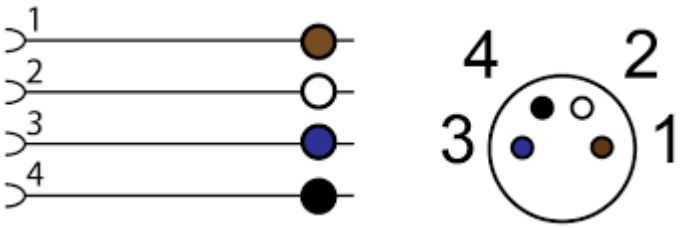
Table 1: Steckerbelegung Standardversion



Triggerversion:

Pin	Funktion	Kabel Farbe
1	+ Versorgung	braun
2	- Trigger	weiß
3	- Vers. (GND)	blau
4	+ Trigger	schwarz

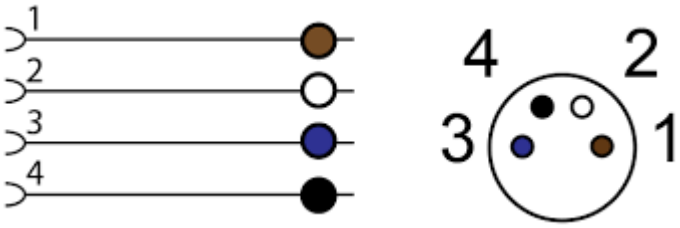
Table 2: Steckerbelegung Triggerversion



CAN Bus Version:

Pin	Funktion	Kabel Farbe
1	+ Versorgung	braun
2	- CAN	weiß
3	- Vers. (GND)	blau
4	+ CAN	schwarz

Table 3: Steckerbelegung CAN Bus Version



SVMtec GmbH | Bergnelkenstraße 7 | 70563 Stuttgart | Deutschland | www.svm-tec.de
Kontakt: +49 (0)711 391867-0 | info@svm-tec.de
Stand: 01.08.2018