TH16MI

General

The multiplexed thermocouple amplifier *TH16MI* puts out a voltage which is proportional to the currently selected type-K-thermocouple's temperature. Up to 16 thermocouples of type K with miniature plug can be connected to the *TH16MI*. These thermocouples are electrically isolated.
The *TH16MI* amplifies the voltage difference which is caused by the thermocouple's temperature and the temperature of the amplifier *TH16MI*. Since the absolute temperature of the thermocouple is subject of interest, the temperature of the *TH16MI* is internally measured and the output signal is correspondingly compensated.
The selection of measurement range and channel number is done by a controller unit (e.g. signal converter *SICO2* or data logger *DL16CAN*). The controller unit transmits an 8-bit multiplex code to the *TH16MI* and stops A/D conversion for a settling time of 20 ms.

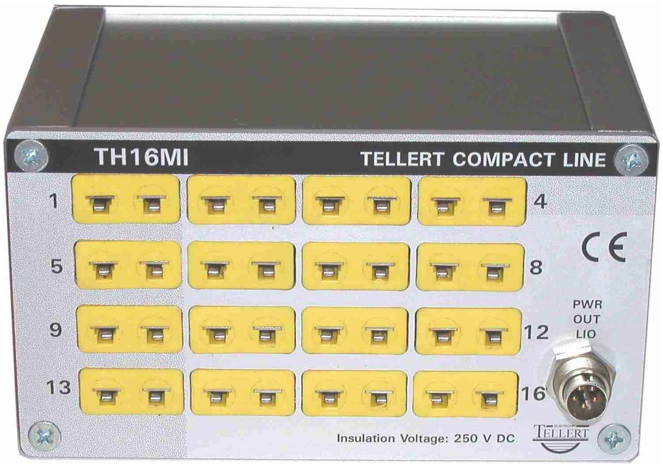

Figure 1: Thermocouple Amplifier *TH16MI*.

Output Signal

Depending on the temperature range *R* and the thermocouple's temperature *T*, the output voltage V_O is defined as $V_O(T) = a(R) \cdot (T + T_O)$.

The output voltage lies within the range from 0.2 V to 4.8 V.

TH16MI does not correct the linearity error of the type-K-thermocouples. *TH16MI* is calibrated for range *R* = 12 at 0 °C and 300 °C. Within this range *TH16MI* has a maximal absolute error (incl. thermocouples' linearity errors) of 3 °C. If not working within this range, the measurement error is received by taking both, the maximal absolute error of the internal temperature compensation (= 2 °C) and the maximal relative error of the amplifier (= 1 %) into account.

Range *R*	Gain $a(R)$ [mV/°C]	Offset T_0 [°C]	$T_{\min}(R)$ [°C]	$T_{\max}(R)$ [°C]
1	3	0	66.7	~~1600~~ 1372
2	4	0	50	1200
3	5	0	40	960
4	10	0	20	480
5	20	0	10	240
6	30	0	6.7	160
7	40	0	5	120
8	50	0	4	96
9	3	50	16.7	~~1550~~ 1372
10	4	50	0	1150
11	5	50	-10	910
12	10	50	-30	430
13	20	50	-40	190
14	30	50	-43.3	110
15	40	50	-45	70
16	50	50	-46	46

Note: The temperature range of type K thermocouples ends at 1372 °C.

Multiplex Code

The 8-bit multiplex code contains the measurement range *R* in its most significant nibble and the channel number *C* in its least significant nibble.

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
R_3	R_2	R_1	R_0	C_3	C_2	C_1	C_0

with range $R = 1 + \sum_{i=0}^{3} R_i \cdot 2^i$, and

with channel number $C = 1 + \sum_{i=0}^{3} C_i \cdot 2^i$

Pin Assignment

The plug of the *TH16MI* is manufactured by *Binder* and part of *Binder Series 712*. The plug pins (in front view) are numbered anti-clockwise starting with the first pin after 12 o'clock position. The first pin is respectively labeled at the solder side (back view).

Pin	Assignment	Description
1	+12 V	Supplying voltage (10 V to 16 V)
2	Ground	
3	Analog Out	Output voltage
4	SDA	Serial data line
5	SCL	Serial clock line

Technical Data

Power supply:	10 V to 16 V DC
Current consumption:	typical 65 mA at 12 V DC
Output voltage:	from 0.2 V to 4.8 V
Insulation voltage:	250 V DC
Settling time:	20 ms
Box size without plug:	100 × 55 × 70 mm

TH16MI CAN

The multiplexed thermocouple amplifier *TH16MI CAN* is an extended version of *TH16MI*. Beneath the properties of the *TH16MI*, it has a built-in signal converter *SICO2* to put out the temperatures over a CAN (Controller Area Network). The voltage output signal of the *TH16MI* unit is connected to the first voltage input AI1 of the internal *SICO2* unit.

Pin Assignment

The sockets and plugs of the *TH16MI CAN* are manufactured by *Binder* and parts of *Binder Series 719*. The socket pins (in front view) are numbered clockwise starting with the first pin after 12 o'clock position. The plug pins are numbered correspondingly anti-clockwise. The first pin is respectively labeled at the solder side (back view).

PWR/CAN: This plug supplies the *TH16MI CAN* with voltage and connects it to a CAN.

Pin	Assignment [Cable Color]
1	Supplying voltage (10 V to 16 V DC inverse-polarity protected) [red]
2	Ground [brown]
3	CAN_L [black]
4	CAN_H [orange]

RS1/2: This plug provides two serial ports. Note, that the *TH16MI CAN* is either shipped with option *GPS* (standard) or option *RS232* (on request).

Pin	Assignment	SUB-D-Plug of host PCs [Comment]
1	TX1	Pin 2
2	Ground	Pin 5
3	RX1	Pin 3
4	U_B – 1 V	Option *GPS*
	TX2	Option *RS232* [Serial data output]
5	RX2	[Serial data input, i.e. GPS]
		Pins 7 and 8 are bridged Pins 1, 4, 6 and 9 are bridged

LOGIC I/O: This socket provides access to an I²C-bus for logical input and output signals.

Pin	Assignment
1	Supplying voltage – 1 V
2	Ground
3	SDA (serial data line)
4	SCL (serial clock line)

Technical Data

Power supply:	10 V to 16 V DC
Current consumption:	typical 130 mA at 12 V DC
Output voltage:	from 0.2 V to 4.8 V
Insulation voltage:	250 V DC
Settling time:	20 ms
Box size without socket and plugs:	100 × 55 × 88 mm

SICO2 Application

Unfortunately the multiplexer support is not easily accessible within TEMES. The first step is to include a text block within the *device comment* of the TEMES parameter set. This text block adds further properties to the parameter set which are not currently accessible within the user interface. The text block looks as follows:

```
<DPF>
[ExParameters]
MpxMode=1
MpxCode=b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 ba bb bc
  bd be bf
AnalogInSuppressionDurationCount=8
AnalogInSuppressionDuration0=20
AnalogInSuppressionDuration1=0
AnalogInSuppressionDuration2=0
AnalogInSuppressionDuration3=0
AnalogInSuppressionDuration4=0
AnalogInSuppressionDuration5=0
AnalogInSuppressionDuration6=0
AnalogInSuppressionDuration7=0
MpxOverlay=0
ActionCode=41 7a a2 ff ff a4 a6 a8 aa ac ae b0
  b2 b4 b6 b8 ba bc be c0 c2
</DPF>
```

Note that the hex value sequences *MpxCode* and *ActionCode* must not be line wrapped. The properties are defined as follows:

Property	Description
MpxMode	0: no multiplexer support 1: multiplexer synchronized with fast cycle 2: multiplexer synchronized with slow cycle
MpxCode	List with hexadecimal multiplex codes. The values of this list are transmitted circularly. If not all 16 channels are used, the list may be accordingly truncated.
AnalogInSuppressionDurationCount	Number of *AnalogInSuppressionDuration* items (always 8)
AnalogInSuppressionDuration*X*	Settling time in ms for the $(X+1)$-th voltage input.
MpxOverlay	Bit mask for overlaying the voltage input value with the channel number (overlay AI1 if bit0 is set ...).

The signals are then accessible via TEMES constants with the corresponding fixed signal addresses. The signal address of a TEMES constant can be assigned by inserting a text block within the comment of the corresponding constant. It looks as follows:

```
[Parameters]
DeviceAddr=160
```

Where 160 is an example for the decimal signal address within the target device.

With the inclusion of the above mentioned *ActionCode*, the following TEMES constants must be defined:

Signal Address	Signal Description
122	MpxCode: Last transmitted multiplex code.
162	Multiplexed temperature signal: This signal must be generated with operation no. 4 (linear transform). It is used as signal source for the following temperature channels.
164	Channel 1 (updated from signal at address 162 if multiplex code indicates channel 1)
166	Channel 2
168	Channel 3
170	Channel 4
172	Channel 5
174	Channel 6
176	Channel 7
178	Channel 8
180	Channel 9
182	Channel 10
184	Channel 11
186	Channel 12
188	Channel 13
190	Channel 14
192	Channel 15
194	Channel 16

Note: If the *ActionCode* mentioned above is included in the parameter set, it is obligatory to define the corresponding constants. Otherwise calculated signals or CAN input signals might be assigned to the predefined signal addresses and may therefore be corrupted by the *ActionCode*.

SIOC2/DL16CAN Example Files

The example parameter sets on the CD-ROM acquire one multiplexed temperature signal. This is realised with the settings for the voltage input AI1, with the device comment entry as mentioned above, and with the calculated signals F1 to F18.

The calculated signals F19 to F21 are used to generate two signals. One CAN output signal for the multiplex code, and one trigger signal for CAN message output.

The parameter sets can easily be adjusted for other measurement ranges. Only the physical assignment of AI1 and the high nibble of the *MpxCode* property must be changed.

DL16CAN Application

A data logger application needs only to record one signal (namely the raw multiplexed temperature signal) if the corresponding bit of the *MpxOverlay* property is set. The 12 most significant bits of the recorded signal represent then the temperature, and the 4 least significant bits represent the channel number. The recorded measurement block must be demultiplexed, and the extracted signals must be linearized. This can be done with an offline calculation (e.g. with *dempx.cdf*).

Mixing measurement ranges

In principal it is possible (but not recommended) to use a different measurement range for each temperature channel. However, the acquisition of a multiplexed temperature signal assumes a main measurement range. This measurement range corresponds to the settings of the voltage input. The resulting restrictions are as follows:

- No linearized online temperature values
- Same measurement range for same channel numbers (if using several multiplexers)
- Linear transform required to obtain the bit value from the main measurement range value, and to obtain the individual measurement range value from the bit value.

If more than one multiplexer is connected to the same controller unit, it is important to use the same measurement ranges for same channel numbers. This restriction is caused by the controller unit since it can only transmit one multiplex code for all connected multiplexers.