



# **TC-08**

# **Thermocouple Logger**

User's Guide

# Contents

1 Introduction .....	1
<b>1 Overview</b> .....	1
<b>2 Installing the driver</b> .....	1
<b>3 Connecting the TC-08</b> .....	1
<b>4 Legal information</b> .....	4
<b>5 Safety warning</b> .....	5
2 Product information .....	6
<b>1 Specifications</b> .....	6
<b>2 Principles of operation</b> .....	7
3 Technical reference .....	8
<b>1 Introduction</b> .....	8
<b>2 Serial port settings</b> .....	8
<b>3 Connections</b> .....	8
<b>4 Protocol</b> .....	9
<b>5 Modem operation</b> .....	11
<b>6 Operating systems</b> .....	11
<b>1 Windows XP and Vista</b> .....	11
<b>7 Driver routines</b> .....	12
<b>1 Introduction</b> .....	12
<b>2 tc08_open_unit</b> .....	13
<b>3 tc08_get_cycle</b> .....	13
<b>4 tc08_close_unit</b> .....	13
<b>5 tc08_poll_driver</b> .....	13
<b>6 tc08_set_channel</b> .....	14
<b>7 tc08_set_resolution</b> .....	15
<b>8 tc08_get_version</b> .....	15
<b>9 tc08_get_temp</b> .....	15
<b>10 tc08_get_cold_junction</b> .....	16
4 Programming support .....	17
<b>1 C/C++ Windows</b> .....	17
<b>2 Delphi</b> .....	18
<b>3 Excel</b> .....	18
<b>4 LabVIEW</b> .....	18
<b>5 Visual Basic</b> .....	18
<b>6 Agilent VEE</b> .....	19
<b>7 Linux</b> .....	19
Index.....	20

# 1 Introduction

## 1.1 Overview

The Pico TC-08 is a complete thermocouple input device for use with IBM-compatible computers. It can be used with the supplied PicoLog data logging program. Alternatively, you can use the TC-08 driver software to develop your own programs to collect and analyse data from the unit.

The TC-08 software provides all of the calculations necessary for cold junction compensation and for thermocouple curve normalisation. The TC-08 is a highly accurate without calibration, and the software does contain facilities to make minor adjustments to the gain and offset.

PicoLog and the drivers support up to nine TC-08 units.

This manual describes the physical and electrical properties of the TC-08, and explains how to use the software drivers. For information on PicoLog software, please consult the PicoLog help file.

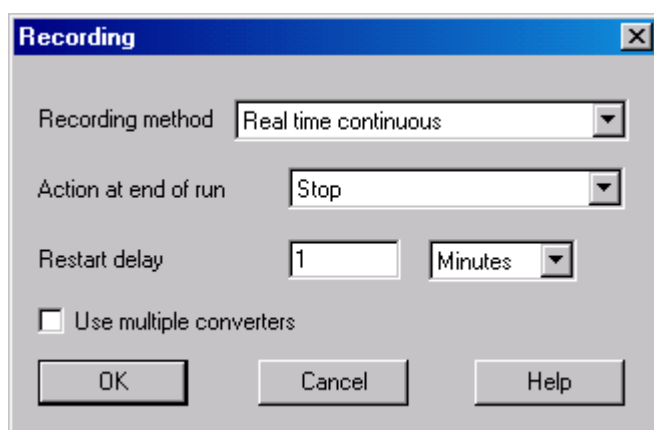
## 1.2 Installing the driver

Installation of the driver is done automatically when you install the PicoLog software. Alternately, you can download the driver from our website at <http://www.picotech.com>.

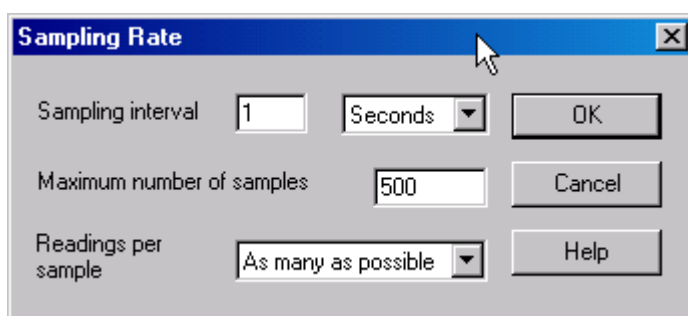
## 1.3 Connecting the TC-08

To begin using the TC-08, you should connect the D-connector on the TC-08 to the serial port on your computer using the cable provided. If you have a 25-way serial port, use a 9 to 25 way adaptor. Next, connect a thermocouple to one or more of the thermocouple input connectors. Now, to set up the unit with your chosen thermocouple, do the following:

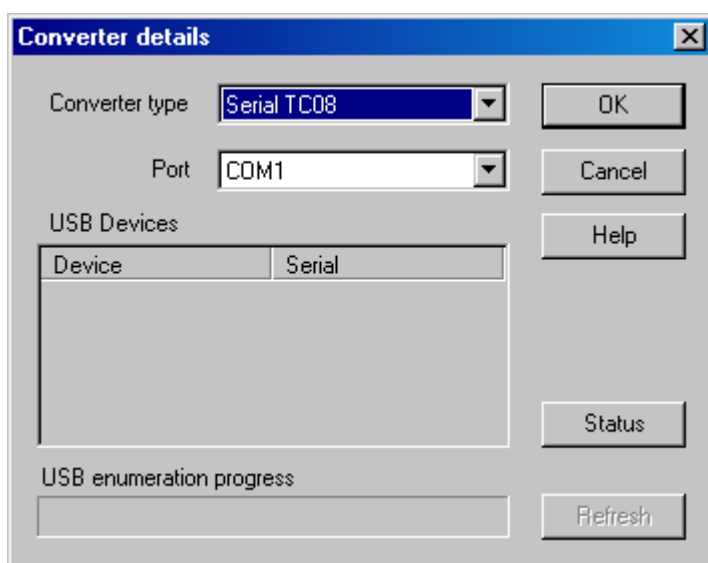
1. Open PicoLog Recorder
2. From the **File** menu, select **New settings**. The Recording dialog box appears:



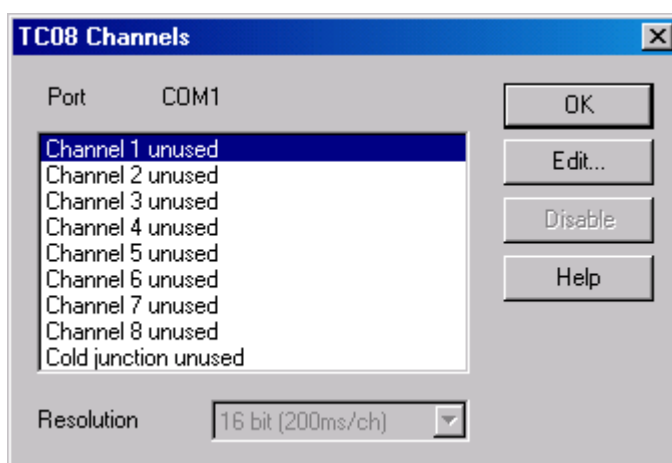
3. Click **OK**. The Sampling Rate dialog box appears:



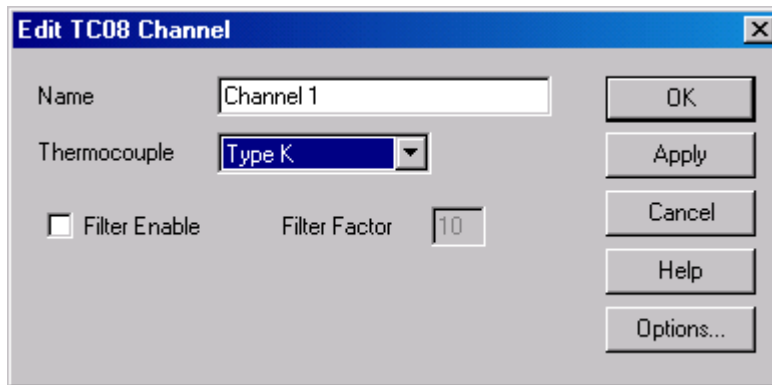
4. Click **OK**. The Converter details dialog box appears:



5. From the **Converter type** drop-down list, select **Serial TC08**.
6. From the **Port** drop-down list, select the port to which the TC08 is connected and click **OK**. The TC08 Channels window appears:



7. In the TC08 Channels window, double-click on **Channel 1 unused**. The Edit TC08 Channel dialog box appears:



8. From the **Thermocouple** drop-down list, select the thermocouple type you wish to use, then click **OK**. The recorder view should now display the temperature.

## 1.4 Legal information

The material contained in this release is licensed, not sold. Pico Technology Limited grants a license to the person who installs this software, subject to the conditions listed below.

### **Access**

The licensee agrees to allow access to this software only to persons who have been informed of these conditions and agree to abide by them.

### **Usage**

The software in this release is for use only with Pico products or with data collected using Pico products.

### **Copyright**

Pico Technology Limited claims the copyright of, and retains the rights to, all material (software, documents etc.) contained in this release. You may copy and distribute the entire release in its original state, but must not copy individual items within the release other than for backup purposes.

### **Liability**

Pico Technology and its agents shall not be liable for any loss, damage or injury, howsoever caused, related to the use of Pico Technology equipment or software, unless excluded by statute.

### **Fitness for purpose**

No two applications are the same: Pico Technology cannot guarantee that its equipment or software is suitable for a given application. It is your responsibility, therefore, to ensure that the product is suitable for your application.

### **Mission-critical applications**

This software is intended for use on a computer that may be running other software products. For this reason, one of the conditions of the license is that it excludes usage in mission-critical applications, for example life-support systems.

### **Viruses**

This software was continuously monitored for viruses during production, but you are responsible for virus-checking the software once it is installed.

### **Support**

If you are unsatisfied with the performance of this software, please contact our technical support staff, who will try to fix the problem within a reasonable time scale. If you are still dissatisfied, please return the product and software to your supplier within 28 days of purchase for a full refund.

### **Upgrades**

We provide upgrades, free of charge, from our web site at [www.picotech.com](http://www.picotech.com). We reserve the right to charge for updates or replacements sent out on physical media.

### **Trademarks**

Pico Technology Limited, PicoScope, PicoLog, DrDAQ and EnviroMon are trademarks of Pico Technology Limited, registered in the United Kingdom and other countries

Pico Technology acknowledges the following product names as trademarks of their respective owners: **Windows, Excel, Visual Basic, LabVIEW, Agilent VEE, HP VEE, Delphi.**

## 1.5 Safety warning

We strongly recommend that you read the general safety information below before using your product for the first time. If the equipment is not used in the manner specified, then the protection provided may be impaired. This could result in damage to your computer and/or injury to yourself or others.

### **Maximum input range**

The TC-08 is designed to measure voltages in the range of  $\pm 60\text{mV}$ . Any voltages in excess of  $\pm 10\text{V}$  may cause permanent damage to the unit.

### **Mains voltages**

Pico products are not designed for use with mains voltages. To measure mains we recommend the use of a differential isolating probe specifically designed for such measurements.

### **Safety grounding**

The ground of every product is connected directly to the ground of your computer via the interconnecting cable provided. This is done in order to minimise interference. If the PC (especially laptop) is not grounded, reading stability cannot be guaranteed and it may be necessary to manually ground the equipment.

As with most oscilloscopes and data loggers, you should take care to avoid connecting the inputs of the product to anything which may be at a hazardous voltage. If in doubt, use a meter to check that there is no hazardous AC or DC voltage. Failure to check may cause damage to the product and/or computer and could cause injury to yourself or others.

Take great care when measuring temperatures near mains equipment. If a sensor is accidentally connected to mains voltages, you risk damage to the converter or your computer and your computer chassis may become live.

You should assume that the product does not have a protective safety earth. Incorrect configuration or use of the device to measure voltages outside the maximum input range can be hazardous.

### **Repairs**

The unit contains no user-serviceable parts: repair or calibration of the unit requires specialised test equipment and must be performed by Pico Technology Limited or their authorised distributors.

## 2 Product information

### 2.1 Specifications

<b>Thermocouple types</b>	B,E,J,K,N,R,S,T	
<b>Voltage mode (type X)</b>	$\pm 60$ mV	
<b>Number of input channels</b>	8	
	Version 1	Version 2
<b>Conversion time -per active channel</b>	800 ms	200 ms
<b>Conversion time-cold junction compensation</b>	1500 ms	200 ms
<b>Cold junction compensation</b>	done by driver	
<b>Uncalibrated accuracy</b>	the sum of $\pm 0.3\%$ and $\pm 0.5^{\circ}\text{C}$	
<b>Common mode range</b>	$\pm 5$ V	
<b>Overvoltage protection</b>	$\pm 10$ V	
<b>Input impedance</b>	2 megaohm	
<b>Input connectors</b>	8 x miniature thermocouple	
<b>Output connector</b>	D9 female to computer serial port	
<b>Power requirements</b>	No power supply required	
<b>Environmental conditions</b>	0 to 50 °C 0 to 95% humidity NOT water-resistant	

The resolution and accuracy depend upon the thermocouple type and the temperature range. The following table shows the overall range that each thermocouple type is calibrated for, and the ranges over which resolutions of  $0.1^{\circ}\text{C}$  and  $0.025^{\circ}\text{C}$  can be achieved. Actual resolution is typically better than these figures at higher temperatures.

<b>Thermocouple type</b>	<b><math>0.1^{\circ}\text{C}</math> Resolution</b>	<b>Overall Range (<math>^{\circ}\text{C}</math>)</b>	<b><math>0.025^{\circ}\text{C}</math> Resolution</b>
B	1030..1800	100..1800	-
E	-240..790	-270..790	-140..790
J	-210..1050	-210..1050	-120..1050
K	-220..1370	270..1370	-20..1150
N	-210..1300	-260..1300	340..1260
R	-333..1760	-50..1760	-
S	250..1760	-50..1760	-
T	-230..400	-270..400	-

The TC-08 can also be used to measure voltages by specifying type X, and the channel can then be used as a differential input with a voltage range of  $\pm 60$  mV.



## 2.2 Principles of operation

An electric current flows in a closed circuit of two dissimilar metals when the two junctions are held at different temperatures.

In such a circuit, called a thermocouple, the magnitude and direction of the current are functions of the temperature difference between the junctions and of the thermal properties of the metals used in the circuit. This phenomenon is known as the Seebeck Effect.

The conductors can be made of any two dissimilar metals, and when the hot junction is heated the current flow can be observed. If the positions of the hot and cold junctions are reversed, current will flow in the opposite direction.

In fact, a thermocouple circuit will actually generate a measurable, low-voltage output that is almost directly proportional to the temperature difference between the hot junction and the cold junction. A unit change in this temperature difference produces some net change in the voltage.

Note: More information on choosing and using thermocouples can be found [here](#).

## 3 Technical reference

### 3.1 Introduction

- [Serial port settings](#)
- [Connections](#)
- [Protocol](#)
- [Modem operation](#)

### 3.2 Serial port settings

The following table shows the standard serial port settings for COM ports.

Port	Base address	Interrupt	Standard?
COM1	3F8	4	Yes
COM2	2F8	3	Yes
COM3	3E8	4	de facto
COM4	2E8	3	de facto
COM5...			No

Note: on most computers, it is not possible to use the same interrupt for two serial ports at the same time. If, for example, you wish to use COM1 and COM3 at the same time, it is necessary to use a serial port card which can be set to an interrupt other than 4. These can be obtained either from Pico Technology or your computer supplier.

### 3.3 Connections

The information presented here is necessary only if you wish to connect the TC-08 to the PC in some unusual way (for example, via a radio modem).

The TC-08 uses the following RS-232 data lines (pin connections as on TC-08)

Pin	Name	Usage
3	TX	Data from the PC to the TC-08
2	RX	Data from the TC-08 to the PC
7	RTS	Held at a positive voltage ( $> 7$ V) to power the TC-08
5	GND	0 V line
4	DTR	Held at a negative voltage ( $< -7$ V) to power the TC-08

The driver powers up the TC-08 by enabling RTS and disabling DTR to provide the correct polarity power supply. If these are set incorrectly no damage will occur to either PC or TC-08.

### 3.4 Protocol

About a second after powering on the TC-08, the driver can communicate with the TC-08 as a normal RS-232 device. The TC-08 operates at 9600 baud with 1 stop bit and no parity.

The driver controls the TC-08 using the following sequence:

1. Switch RTS on and DTR off to provide power.
2. Wait for more than 1 second for the TC-08 to settle
3. Send an single control byte to the TC-08
4. Wait for the 3-byte response from the TC-08

Steps 3 and 4 are repeated for each measurement.

The TC-08 signals the end of conversion by sending three bytes. No data should be sent to the TC-08 during the conversion, as it may be lost or corrupted.

The following control codes are used:

```
0x00,      /* Channel 1 */
0x20,      /* Channel 2 */
0x40,      /* Channel 3 */
0x60,      /* Channel 4 */
0x80,      /* Channel 5 */
0xA0,      /* Channel 6 */
0xC0,      /* Channel 7 */
0xE0,      /* Channel 8 */
0x01,      /* version */
0x22,      /* Cold junction- reference */
0x42,      /* Cold junction- thermistor */
```

For the channels, the returned value is a three-byte sequence.

Byte 1 is the sign '+' or '-'

Byte 2 is the most significant byte of the reading

Byte 3 is the least significant byte of the reading.

The reading is a 16-bit plus sign number, where 0 microvolts is represented by a reading of zero, and +/- 59,524 microvolts are represented by +/- 65,535.

The cold junction temperature is calculated using the readings from the reference (ref) and the thermistor (th),

```
divisor = (th + 65535L) / 65536L

Result = (65536L * (th / divisor)) / (ref / divisor)
```

The result is converted to temperature using the following table:

```
/* 0000 */ 230216L,
/* 0001 */ 218272L,
/* 0002 */ 206947L,
/* 0003 */ 196210L,
/* 0004 */ 186030L,
/* 0005 */ 176378L,
/* 0006 */ 167491L,
/* 0007 */ 159051L,
/* 0008 */ 151037L,
```

```

/* 0009 */ 143427L,
/* 0010 */ 136200L,
/* 0011 */ 129533L,
/* 0012 */ 123192L,
/* 0013 */ 117161L,
/* 0014 */ 111426L,
/* 0015 */ 105971L,
/* 0016 */ 100929L,
/* 0017 */ 96127L,
/* 0018 */ 91554L,
/* 0019 */ 87198L,
/* 0020 */ 83049L,
/* 0021 */ 79207L,
/* 0022 */ 75543L,
/* 0023 */ 72048L,
/* 0024 */ 68715L,
/* 0025 */ 65536L,
/* 0026 */ 62587L,
/* 0027 */ 59771L,
/* 0028 */ 57081L,
/* 0029 */ 54513L,
/* 0030 */ 52060L,
/* 0031 */ 49780L,
/* 0032 */ 47600L,
/* 0033 */ 45516L,
/* 0034 */ 43523L,
/* 0035 */ 41618L,
/* 0036 */ 39844L,
/* 0037 */ 38146L,
/* 0038 */ 36520L,
/* 0039 */ 34964L,
/* 0040 */ 33474L,

```

To convert a thermocouple reading, in microvolts, into temperature, you will need to use a table containing microvolt readings at various temperatures for the type of thermocouple that you wish to use. You can obtain these tables from the British Standards Institute or thermocouple manufacturers.

Here is a section of the table for a type K thermocouple:

Temp (°C)	Thermocouple output (µV)
0	0
10	397
20	798
30	1203
40	1612
50	2023
60	2436

First, take the cold junction temperature and use this table to convert it to microvolts. Add this onto the microvolt reading from the thermocouple. Convert the total back to temperature using this table.

## 3.5 Modem operation

The TC-08 is normally connected directly to the computer, but it is also possible to access the TC-08 via a modem using the Windows driver.

It is necessary to provide power to the TC-08, either by instructing the modem to provide power or by connecting a power supply directly to the TC-08. See serial connections for information.

For some radio modems, there is a delay between sending text to the modem and its arrival at the other end, and a similar delay for the response from the TC08. If, for example, the maximum possible delay is 150 ms each way, 300 ms total, the following text should be added to `win.ini` so that the driver waits longer for each response.

```
[TC08]  
Turnround=300
```

**Warning:** In order to comply with current legislation, use only radio modems which comply with the RTTE directive.

## 3.6 Operating systems

### 3.6.1 Windows XP and Vista

Most applications running under Windows XP and Vista are 32-bit applications. The 32-bit Windows driver is the file `TC0832.dll`, installed in the `drivers` directory. If an application is unable to find the DLL, try moving the DLL to `windows/system`.

## 3.7 Driver routines

### 3.7.1 Introduction

The TC-08 is supplied with driver routines that you can build into your own programs. Drivers are supplied for the following operating systems:

#### [Windows XP/Vista](#)

Once you have installed the software, the [DRIVERS](#) directory contains the drivers and a selection of examples of how to use the drivers. It also contains a copy of this manual as a text file. If you installed under Windows, the Pico Technology group contains a help file for the drivers. See the Readme.doc file in the [DRIVERS](#) directory for the filenames.

The driver routine is supplied as a Dynamic Link Library for Windows.

The object files use Pascal linkage conventions and do not require any compiler run-time routines. They can therefore be used with most real-mode and some protected-mode C compilers.

The Windows DLL can be used with C, Delphi and Visual Basic programs: it can also be used with programs like Microsoft Excel, where the macro language is a form of Visual Basic. More than one application can access the Windows DLL at the same time, as long as the applications do not change the settings for channels that they are not using.

The following table specifies the function of each of the routines in the driver:

Routine	Description
<a href="#">tc08_open_unit</a>	Open the driver to use a specified serial port(s)
<a href="#">tc08_close_unit</a>	Close the port (ALWAYS DO THIS!)
<a href="#">tc08_poll_driver</a>	Poll the driver (not usually necessary)
<a href="#">tc08_get_cycle</a>	Find out when the driver has taken a new set of readings
<a href="#">tc08_set_resolution</a>	Specify the resolution (and speed) for conversions
<a href="#">tc08_set_channel</a>	Specify the thermocouple type and filtering for a channel
<a href="#">tc08_get_temp</a>	Get the most recent temperature reading from a channel
<a href="#">tc08_get_cold_junction</a>	Get the cold junction temperature for a TC08
<a href="#">tc08_get_version</a>	Get the version number of this TC-08

The normal calling sequence for these routines is as follows:

1. Open driver
2. Set channels
3. While you want to measure temperatures:
  4. Get temperature
5. End While
6. Close Driver

### 3.7.2 tc08\_open\_unit

```
unsigned short tc08_open_unit (  
    unsigned short port);
```

This routine specifies the serial port number with an TC-08 unit. If you wish to use more than one TC-08, you should call the routine once for each TC-08.

The port must be 1 for COM1, 2 for COM2, etc.

The base address and IRQ information is defined in your `WIN.INI` file, so it is not necessary to specify a value.

This routine returns TRUE if the driver successfully opens the TC-08.

### 3.7.3 tc08\_get\_cycle

```
unsigned short tc08_get_cycle(  
    long          * cycle,  
    unsigned short port);
```

This routine returns the number of complete cycles of readings taken from a particular TC-08.

When you call [tc08\\_get\\_temp](#), it returns immediately with the most recent reading for the specified channel. If you call it repeatedly, it will return the same reading repeatedly, until the driver takes the next reading from that channel.

If you wish to record values only when the driver has taken a new reading, you can use this routine to find out how many complete cycles of readings the driver has taken, then you can call [tc08\\_get\\_temp](#) only when a cycle has completed.

Note: each TC-08 is polled independently, so the cycle numbers for multiple TC-08s may not keep in step.

### 3.7.4 tc08\_close\_unit

```
void tc08_close_unit (unsigned short port);
```

This routine disconnects the driver from the specified serial port.

If you successfully open any serial ports, you MUST call [tc08\\_close\\_unit](#) for each port before you exit from your program. If you do not, your computer may misbehave until you next reboot it.

### 3.7.5 tc08\_poll\_driver

```
void tc08_poll_driver (void);
```

It is not normally necessary to call this routine, as the driver uses the timer to poll the TC-08. Some programs, like Excel, appear block the timer and so it is necessary to poll the driver periodically whilst waiting for data.

### 3.7.6 tc08\_set\_channel

```
void tc08_set_channel (  
    unsigned short port,  
    unsigned short channel,  
    char          tc_type,  
    unsigned short filter_factor,  
    short         offset,  
    short         slope);
```

You should call this routine once for each channel that you would like to take readings from. You can do this any time after calling [tc08\\_open\\_unit](#).

The fewer channels are selected, the more frequently these channels will be updated: it takes about 2 seconds for cold junction compensation and 1 second per active channel.

#### Arguments:

`channel` -specifies which channel you want to set the details for. It should be between 1 and 8.

`tc_type` -specifies what type of thermocouple is connected to this channel. Set `tc_type` to one of B, E, J, K, N, R, S or T or X: X makes the channel a 60 mV input. You can also set it to blank to deactivate a channel that you have already been using.

`filter_factor` -controls the time constant of the filter. Each time the driver takes a reading from this channel, it updates the filtered value by adding a proportion of the difference between the measured and filtered values. The `filter_factor` sets the proportion that is added. A `filter_factor` of 1 means add all of the difference (effectively no filtering) and 100 means add 1/100 of the difference (very slow filtering). A factor of 10 gives a time constant of about a minute when all channels are selected.

The basic accuracy of the TC-08 is adequate for most purposes, but the slope and offset can be used to calibrate the unit to eliminate the effect of offsets and gain errors. For both parameters, a value of zero gives an unadjusted result.

The scale of the offset parameter depends on the type of thermocouple, but is typically about 0.02 degrees. The gain adjusts are added onto the gain, so zero gives the default gain. +1 gives a gain adjustment of +0.01%. The maximum gain adjust is about  $\pm 1\%$ . Note that the slope and offset have no effect when using type X (60 mV input).

For work in the 0 to 100 °C range, the offset is the main problem, and can easily be calibrated out without special equipment. Just short the thermocouple input that you wish to calibrate, then adjust the `offset` until the channel temperature is exactly the same as the cold junction temperature for that channel.

To adjust for gain errors, first calibrate out any offset on the channel, then connect the channel input to a thermocouple which is at a known temperature, at the opposite end of the range where you wish to work. Next, adjust the `slope` for the channel until the channel temperature reading is correct.



### 3.7.7 tc08\_set\_resolution

```
void tc08_set_resolution (
    unsigned short port,
    unsigned short resolution);
```

The TC-08 normally works at 16-bit resolution, which takes 200 ms per channel. If you want to collect data faster, you can call this routine to select a lower resolution. You can do this any time after calling [tc08\\_open\\_unit](#).

The choices are 13, 14, 15 or 16 bits. 13 bits is the fastest: 16 bits is the most accurate.

### 3.7.8 tc08\_get\_version

```
unsigned short tc08_get_version (
    unsigned short * version,
    unsigned short port);
```

This routine sets version to version number of the specified TC-08.

The upper byte of the version is always 8 for a TC-08: the lower byte is the two hex digits of the version and release. It provides a useful check that the link to the TC-08 is working correctly.

### 3.7.9 tc08\_get\_temp

```
unsigned short tc08_get_temp (
    long * temp,
    unsigned short port,
    unsigned short channel,
    unsigned short filtered);
```

Once you open the driver and define some channels, the driver constantly takes readings from the TC-08. When you call this routine, it immediately sets temp to the most recent reading for the specified channel.

Temperatures are returned in hundredths of a degree Celsius, and voltages (type X) in microvolts.

If a reading is available, it returns TRUE, otherwise it returns FALSE. It will normally return FALSE for a few seconds after you open the driver, until the driver has taken a reading from the specified channel.

`channel` should be 1 for channel 1, 2 for channel 2 and so on.

If you set filtered to TRUE, the driver returns a low-pass filtered value of the temperature. The time constant of the filter depends on the value of `filter_factor` for this channel, and on how many channels are active.

**3.7.10** tc08\_get\_cold\_junction

```
unsigned short tc08_get_cold_junction (  
    long          * temp,  
    unsigned short port);
```

This routine sets temp to the cold junction temperature for the specified TC-08, in hundredths of a degree Celsius.

Normally, you do not need to use the cold junction temperature, as the driver automatically compensates for it. It can, however, be useful as an indication of ambient temperature, and when calibrating the offset for a channel.

## 4 Programming support

### 4.1 C/C++ Windows

The C example source code is generic for windows applications - (ie - it does not contain a files specific to a compiler or IDE). However there are two library files supplied which are compiler dependant: `tc0832.lib` (for Borland and other OMF compatible compilers) and `tc08ms.lib` (for Microsoft Visual C++ and other COFF compatible compilers).

#### C++ (Windows)

C++ programs can access all versions of the driver. If `tc08.h` is included in a C++ program, the `PREF1` macro expands to `extern "C"`. This disables name decoration, and enables C++ routines to make calls to the driver routines using C headers.

#### Windows GUI Application

To compile the programs, create a new project for an Application containing the following files:

```
tc08tes.c  
tc08tes.rc
```

and either

```
tc0832.lib - Borland (OMF) 32-bit applications
```

or

```
tc08ms.lib - Microsoft Visual C++ (COFF) 32-bit applications
```

The following files must be in the same directory:

```
tc08tes.rch  
tc08.h  
tc0832.dll (All 32-bit applications)
```

#### Windows Console Application

To compile the programs, create a new project for an Application containing the following files:

```
tc08con.c
```

and either

```
tc0832.lib - Borland (OMF) 32-bit applications
```

or

```
tc08ms.lib - Microsoft Visual C++ (COFF) 32-bit applications
```

The following files must be in the same directory:

```
tc08.h  
tc0832.dll (All 32-bit applications)
```

## 4.2 Delphi

The `WIN` subdirectory contains a simple program `tc08.dpr` which opens the drivers and reads temperatures from the three channels. You will need the following files to build a complete program.

- `tc08fm.dfm`
- `tc08fm.pas`
- `tc08.inc`

The file `tc08.inc` contains procedure prototypes for the driver routines: you can include this file in your application. This example has been tested with Delphi versions 1, 2 and 3.

## 4.3 Excel

The easiest way to get data into Excel is to use PicoLog.

If, however, you need to do something that is not possible using PicoLog, you can write an Excel macro which calls `tc08xx.dll` to read in a set of data values. The Excel Macro language is similar to Visual Basic.

### Excel 7

The example `TC0832.XLS` reads in 20 values of the cold junction temperature and channel 1 temperature, one per second, and assigns them to cells A1..B20.

## 4.4 LabVIEW

The routines described here were tested using LabVIEW for Windows 95 version 4.0.

While it is possible to access all of the driver routines described earlier, it is easier to use the special LabVIEW access routine. The `tc08.lib` library in the `DRIVERS` subdirectory shows how to access this routine.

To use this routine, copy `tc08.lib` and `tc0832.dll` to your LabVIEW `user.lib` directory.

You will then use the `tc08` sub-vi, and an example sub-vi which demonstrate how to use them. You can use one of these sub-vis for each of the channels that you wish to measure. The sub-vi accepts the port (1 for COM1), the channel (1 to 3) the thermocouple type ('K' for type K). The sub-vi returns a temperature for thermocouple types, and a voltage in microvolts for type X.

## 4.5 Visual Basic

### Version 4 and 5

The `DRIVERS` subdirectory contains the following files:

- `TC0832.VBP`
- `TC0832.BAS`
- `TC0832.FRM`

## 4.6 Agilent VEE

The example program `tc08.vee` shows how to collect a block of data from the TC-08. You will need to copy the following files to the program directory:

`tc08.vh`

## 4.7 Linux

See the `tc08.tar` file for more information.

# Index

## C

Cold junction 7  
Connecting the TC-08 1  
Connections 8

## D

Data lines 8  
Delphi 18

## E

Excel 18

## H

HP-Vee 19

## I

Installation 1

## L

Labview 18  
Legal information 4  
LINUX 19

## M

Modem operation 11

## P

Protocol 9

## S

Safety warning 5  
Seebeck Effect 7  
Serial port settings 8  
Specifications 6

## T

tc08\_close\_unit 13  
tc08\_get\_cold\_junction 16  
tc08\_get\_cycle 13  
tc08\_get\_temp 15  
tc08\_get\_version 15

tc08\_open\_unit 13  
tc08\_poll\_driver 13  
tc08\_set\_channel 14  
tc08\_set\_resolution 15  
Temperature table 9  
Thermocouple circuits 7

## V

Visual Basic 18

## W

Windows NT 11



## Pico Technology Ltd

The Mill House  
Cambridge Street  
St Neots PE19 1QB  
United Kingdom  
Tel: +44 (0) 1480 396 395  
Fax: +44 (0) 1480 396 296  
Web: [www.picotech.com](http://www.picotech.com)