TC-08 Manual v1.0

PicoLog Manual (Datalogging software)

Software Updates

Writing you own software overview

C | Turbo Pascal | C/C++ Win | Delphi | Visual Basic | Excel | HP Vee | LabView | Linux

Other Information

Contact Pico | Technical Support | Signal Conditioners | Safety | Legal | Print This Manual

Technical Information

Serial port settings	(Interrupts etc)
Serial port connections	(Pin connections on the serial port)
Protocol	(Directly driving the unit)
Modem operation	(Remote use with radio / telephone modems)

Safety Warning

The ground of the TC-08 is connected directly to the ground of your computer, in order to minimise electrical interference.

Take care when using bare-wire thermocouples that the thermocouple does not come into contact with voltages outside the operating ra TC-08, as you may risk damage to the TC-08 and your computer.

When measuring temperatures on mains equipment, we recommend the use of insulated thermocouple probes. If a sensor input is accide connected to mains, your computer chassis may become live.

Introduction

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: the software does contain facilities to make minor adjustments to the gain and offset.

Picolog and the drivers support up to four TC-08 units under DOS, and nine TC-08 units under Windows.

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.

Connecting the TC-08

To use 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 the 9 to 25 way adaptor supplied.

Next, connect a thermocouple to one or more of the thermocouple input connectors.

To check that the unit is working:

- start up PicoLog
- select the File menu
- select New Settings
- In the recording window, press OK
- In the sampling window, press OK
- In the converters window, Select TC-08
- Select the port that the TC-08 is connected to
- Press OK
- In the TC-08 Channels window, double click on ‘Channel 1 unused’
- Select the thermocouple type you wish to use in Thermocouple type and press OK
- Press OK

Readings from the TC-08 should appear in the monitor window.

Principles of operation

In 1822, an Estonian physician named Thomas Seebeck discovered (accidentally) that the junction between two metals generates a voltage which is a function of temperature.

Any circuit made up of two or more metals must involve at least two such junctions: if they are all at the same temperature, the voltages are the same at each junction, so they cancel out.

With a circuit made of two metals and two junctions, if the two junctions are at different temperatures, it is possible to measure the difference between the voltages generated by the two junctions.

For practical temperature measurement, we place the 'Hot' junction where we wish to measure the temperature, then measure the voltage. In order to calculate the 'Hot' temperature, we must either place the 'Cold' junction at a known temperature, (for example, in melting ice), or first measure its temperature and then compensate for its temperature. The TC-08 contains a circuit to measure the temperature of the cold junction, and the software driver carries out the calculations necessary for cold junction compensation.

The voltage between A and B depends on the type of metal. There are two main groups of commercially available thermocouple: base metal and noble metal. Base metal thermocouples, for example type J and K, are made of alloys which are optimised to give a relatively high voltage, but which deteriorate and eventually fail at high temperature. Noble metal thermocouples, for example R and S, produce a much smaller voltage- virtually zero at room temperature- but it is possible to buy units that are suitable for extended use at temperatures up to 1800 C.

A type K thermocouple produces a voltage which changes by about 40µV per degree Celsius. The TC-08 first amplifies this signal, then feeds it into a 16-bit analog to digital converter. The amplifier and ADC are set up to give an input sensitivity of approx 1µV per LSB: the resolution for a type K thermocouple is therefore 1/40 C. The ADC can take a measurement for a channel every 800ms.

The software continuously takes readings from the selected channels and from the cold junction compensation circuit. For each reading, it updates a low pass filter. You can use either the measured or the filtered value: the filtered value is much less prone to electrical noise, but it tends to lag behind if the measured value changes quickly.

The filter works by adjusting the filtered value by a proportion of the difference between the measured and filtered values. This proportion is controlled by the filter factor. A high filter factor means that only a small proportion of the difference is added each time, so the filtered value changes very slowly. The filter factor is fixed at 10 for PicoLog, and with the drivers it can be adjusted between 1 (no filter) and 100 (very slow filter).

Note that the filter time constant is also affected by the number of channels that are in use. The more channels selected, the slower the filter.

(More information on choosing and using Thermocouples can be found in our collection of application notes at http://www.picotech.com)

Specifications

Thermocouple types	B,E,J,K,N,R,S,T	
Voltage mode (type X)	±60mV	
Number of input channels	8	
	Version 1	Version 2
Conversion time -per active channel	800ms	200ms
Conversion time-cold junction compensation	1500ms	200ms
Cold junction compensation	carried out by driver	
Uncalibrated accuracy	the sum of ±0.3% and ±0.5 C	
Common mode range	±4V	
Overvoltage protection	±10V	
Input impedance	2M Ohm	
Input connectors	8 x miniature thermocouple	
Output connector	9 way female D-type to computer serial port	
Power requirements	No power supply required	
Environmental conditions	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 C and 0.025 C can be achieved. Actual resolution is typically better than these figures at higher temperatures.

Thermocouple type	Overall Range (C)	0.1 C Resolution	0.025 C Resolution
B	100..1800	1030..1800	-
E	-270..790	-240..790	-140..790
J	-210..1050	-210..1050	-120..1050
K	-270..1370	-220..1370	-20..1150
N	-260..1300	-210..1300	340..1260
R	-50..1760	330..1760	-
S	-50..1760	250..1760	-
T	-270..400	-230..400	-20..400

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

Drivers

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

DOS
Windows 3.x
Windows 95/98
Windows NT/2000

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 object files for DOS and protected mode, and 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 and Pascal 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	**Function**
`tc08_open_unit`	Open the driver to use a specified serial port(s)
`tc08_close_unit`	Close the port (ALWAYS DO THIS!)
`tc08_poll_driver`	Poll the driver (not usually necessary)
`tc08_get_cycle`	Find out when the driver has taken a new set of readings
`tc08_set_resolution`	Specify the resolution (and speed) for conversions
`tc08_set_channel`	Specify the thermocouple type and filtering for a channel
`tc08_get_temp`	Get the most recent temperature reading from a channel
`tc08_get_cold_junction`	Get the cold junction temperature for a TC08
`tc08_get_version`	get the version number of this TC-08

The normal calling sequence for these routines is as follows:

Open driver
Set Channels
While you want to measure temperatures,
  Get temperature
End While
Close Driver

tc08_open_unit

DOS version:

```
unsigned short tc08_open_unit (
            unsigned short    port,
            unsigned short    base,
            unsigned short    irq);
```

Windows version:

```
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.

Under DOS, this routine has extra parameters to specify the base address an interrupt number for the COM port. These can be set to zero for the default base address and IRQ. Under Windows, 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.

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.

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.

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.

tc08_set_resolution

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

The TC-08 normally works at 16-bit resolution, which takes 200ms 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.

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.

`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 ±60mV input. You can also set it to blank to de-activate a channel that you have already been using.

The `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 ±1%. Note that the slope and offset have no effect when using type X (60mV 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.

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 et cetera.
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.

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.

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.

DOS Driver

The DOS driver is supplied in two object files, `tc08drv.obj` and `commdrv.obj`. It can be used in both C and Pascal programs.

Windows 3.x Driver

The windows 16-bit driver is the file `TC0816.DLL`: it is installed in `the drivers\win` directory. If an application is unable to find the DLL, try moving the DLL to `\windows\system`.

The 16-bit driver is intended for use with all applications running under Windows 3.11 and for 16-bit applications running under Windows 95.

Windows 95/98

Windows 95 and 98 can run both 16-bit and 32-bit applications. For 16-bit applications, see Windows 3.1.

The windows 32-bit driver is the file `TC0832.DLL`: it is installed in `the drivers\win32` directory. If an application is unable to find the DLL, try moving the DLL to `\windows\system`.

Windows NT and 2000

Most applications running under Windows NT and 2000 are 32-bit applications. The windows 32-bit driver is the file `TC0832.DLL`: it is installed in `the drivers\win32` directory. If an application is unable to find the DLL, try moving the DLL to `\windows\system`.

C (DOS)

To link the driver into you program, you should take the following steps:

add #define DOS at the top of your program

`#include` the header file `tc08.h` into your program

If you are using an IDE, include the file tc08`drv.obj and commdrv.obj` in you project.

If you are using a command-line compiler, include the file `tc08drv.obj and commdrv.obj` in you linkfile.

C (Windows)

The C example program is a generic windows application- ie it does not use Borland AppExpert or Microsoft AppWizard. To compile the program, create a new project for an Application containing the following files:

tc08tes.c
tc08tes.rc
either tc0816.lib (All 16-bit applications)
or tc0832.lib (Borland 32-bit applications)
or tc08ms.lib (Microsoft Visual C 32-bit applications)

The following files must be in the same directory:

tc08tes.rch
tc08w.h
either tc0816.dll (All 16-bit applications)
or tc0832.dll (All 32-bit applications)

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-mangling (or decoration, as Microsoft call it), and enables C++ routines to make calls to the driver routines using C headers.

Pascal

The program `tc08pas.pas` can be compiled either as a stand-alone program `{$DEFINE MAIN}` or as a unit which can be linked into other programs `{$UNDEF MAIN}`.

`tc08pas.pas` includes the driver using the `{$L tc08drv.obj}` and `{$L commdrv.obj}` commands: it also provides pascal prototypes for each of the routine in the driver.

This program has been tested with Borland Turbo Pascal V6.0.

BASIC

The TC08 is not supported under DOS basic.

Delphi

The `WIN` sub-directory 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.

Excel

The easiest way to get data into Excel is to PicoLog for Windows.

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 5

The example `TC0816.XLS` reads in 20 values of the cold junction temperature and channel 1 temperature, one per second, and assigns them to cells A1..B20.

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.

Visual Basic

Version 3 (16 bits)

The DRIVERS\WIN16 sub-directory contains a simple Visual Basic program, `TC0816.mak`.

TC0816.MAK
TC0816.FRM

Note that it is usually necessary to copy the .DLL file to your \windows\system directory.

Version 4 and 5 (32 bits)

The DRIVERS\WIN32 sub-directory contains the following files:

TC0832.VBP
TC0832.BAS
TC0832.FRM

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.llb library in the `DRIVERS\WIN32` sub-directory shows how to access this routine.

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

You will then 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.

HP-Vee

The routine described here was tested using HP-Vee version 5 under Windows 95.

The example program tc08.vee shows how to collect a block of data from the tc-08.

LINUX

See the tc08.tar file for more information.

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 that, 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.

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 RS232 data lines (pin connections as onTC-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 (>7V) to power the TC-08
5	GND	0V line
4	DTR	Held at a negative voltage (<-7V) 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.

Protocol

About a second after powering on the TC-08, the driver can communicate with the TC-08 as a normal RS232 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 +/-59524 microvolts are represented by +/-65535.

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 (degC)	Thermocouple output (uV)
0	0
10	397
20	798
30	1065
40	1340
50	1619
60	1902

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

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 150ms each way, 300ms total, the following text should be added to Win.INI so that the driver waits longer for each response.

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