



PT-104

Thermistor Converter

User's Guide

Contents

1 Introduction	1
1 Overview	1
2 Safety warning	2
3 Legal information	3
2 Product information	4
1 Specifications	4
2 Installing the driver	4
3 Connection	5
4 Background on PRTs	8
3 Driver information	10
1 Introduction	10
2 pt104_open_unit	11
3 pt104_close_unit	11
4 pt104_poll_driver	11
5 pt104_get_cycle	12
6 pt104_set_channel	12
7 pt104_set_mains	13
8 pt104_get_value	13
9 pt104_get_version	14
10 pt104_get_unit_info	14
11 pt104_get_driver_version	14
12 pt104_labview	15
13 Windows XP (SP2)/Vista	15
14 Linux	15
4 Writing your own programs	16
1 C / C++	16
2 Delphi	17
3 Excel	17
4 LabVIEW	17
5 Visual Basic	18
6 Agilent VEE	18
5 Technical reference	19
1 Serial port settings	19
2 Serial port connections	19
3 Protocol information	20
4 Modem operation	22
5 Lookup table	22
Index.....	27

1 Introduction

1.1 Overview



The PT-104 is a four-channel, high-resolution temperature converter for use with PT100 and PT1000 type sensors. It can be used to measure temperature, resistance and voltage.

In PT100/PT1000/resistance mode, the unit uses a four-wire circuit.

In voltage mode, the input connector can be treated as a differential input with ground, or two single-ended inputs. Both inputs must be 0 V or above, though it does not matter which input has the higher voltage. For the 115 mV range, the accuracy may vary by 2%, and the temperature coefficient will be 100 ppm/°C.

1.2 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 PT-104 is designed to measure voltages in the range 0 to +2.5 V. Any voltages in excess of ± 30 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.

1.3 Legal information

The material contained in this release is licensed, not sold. Pico Technology Limited grants a licence 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. However, you are responsible for virus-checking the software once it is installed.

Support

If you are dissatisfied 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 14 days of purchase for a full refund.

Upgrades

We provide upgrades, free of charge, from our web site at 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, Delphi.

2 Product information

2.1 Specifications

	Temperature	Resistance	Voltage
Sensor	PT100*, PT1000	N/A	N/A
Range	-200 to 800°C	0 to 375 Ω *; 0 to 10 k Ω	0 to 115 mV; 0 to 2.5 V*
Linearity	10 ppm	10 ppm	10 ppm
Accuracy @25°C	0.01°C*	20 ppm*	0.2%*
Temperature coefficient	3 ppm/°C	3 ppm/°C	100 ppm/°C
RMS Noise (using filter)	0.01°C	10 ppm	10 ppm
Resolution	0.001°C	1 m Ω	0.156 mV
Conversion Time Per Channel	720 ms**	720 ms**	180 ms
Number of inputs	4		
Connectors	4-pin mini-DIN		
Input impedance	>> 1 M Ω		
Overvoltage protection	\pm 30 V		
Output	RS-232, DE9 female		
Environmental	20 to 30°C for stated accuracy; 0 to 70°C overall; 20 to 90% RH		
Software	PicoLog data logging software; Drivers for Windows XP SP2/Vista; Examples for C/C++, Delphi, Excel, LabVIEW, Visual Basic, Agilent-VEE.		
Quoted accuracy is for options marked *			
For four-wire temperature and resistance measurement**			

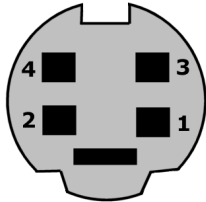
2.2 Installing the driver

The driver is installed automatically when you install the PicoLog software. Alternatively, you can download the driver from our website at <http://www.picotech.com>.

2.3 Connection

To use the PT-104, you should connect its D-connector to your computer's serial port using the cable provided. Next, connect a PT100 or PT1000 PRT to one of the input connectors.

Pin Connections to the PT-104 Mini-DIN socket



Pin	PT100, PT1000 4 Wire	PT100, PT1000 3 Wire	PT100, PT1000 2 Wire	Differential voltage
1	White	Connect to pin 3	Connect to pin 3	Do not connect
2	Red	Red	Red	V -
3	White	White	White	V +
4	Red	Red	Connect to pin 2	Gnd

Single-ended voltage connection

Single-ended mode allows you to double the number of channels from 4 to 8. It is supported by the driver, so you can use it in your own applications. It is not, however, supported by PicoLog.

Connector	Channel	Pin
1	1	3
	5	2
2	2	3
	6	2
3	3	3
	7	2
4	4	3
	8	2

Note on Differential Voltage Mode

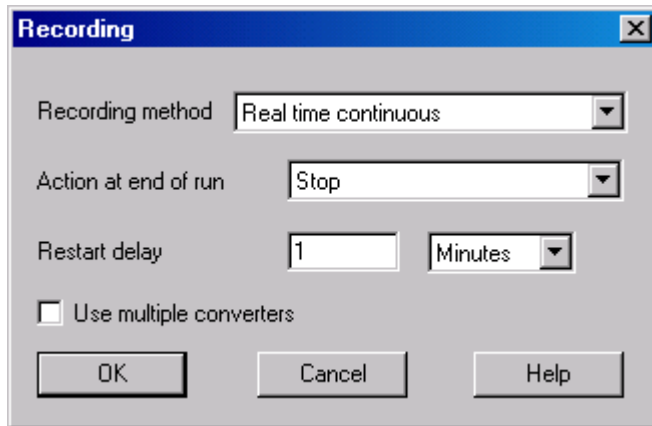
The maximum input voltage range of the PT-104 is 2.5 V. Any voltage in excess of +/-30 V on any input pin may cause permanent damage to the unit.

In Differential Voltage Mode, the input connector should be treated as a differential input with reference to ground. Both inputs (V+ and V-) must be zero volts or above (it does not matter which input has the higher voltage) and must remain within the input range. A ground reference connection is also required for correct operation. The ground connection of each MiniDIN socket consists of a 100 Ω resistor to mains earth/ground via the serial cable outer braiding and the PC chassis.

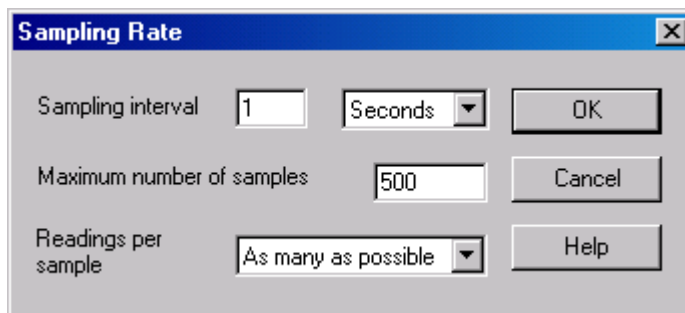
Setting up

To set up the unit, do the following:

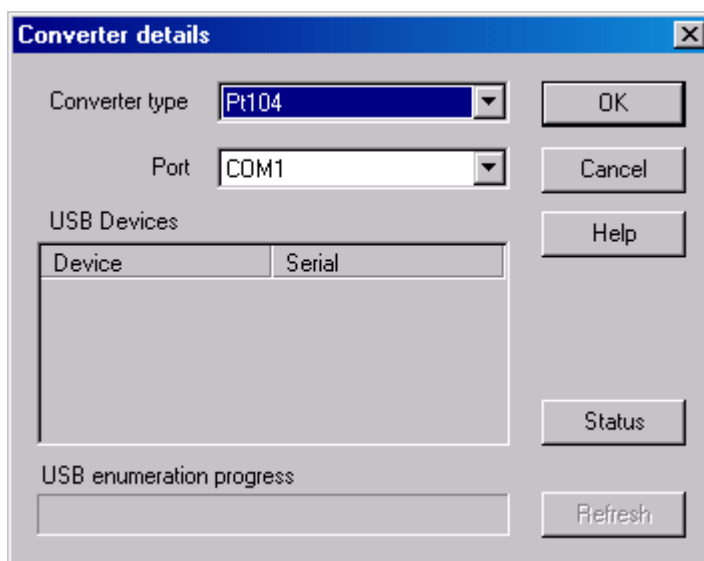
1. From the PicoLog Recorder File menu, select New settings. The Recording dialog box appears:



2. Click OK. The Sampling Rate dialog box appears:

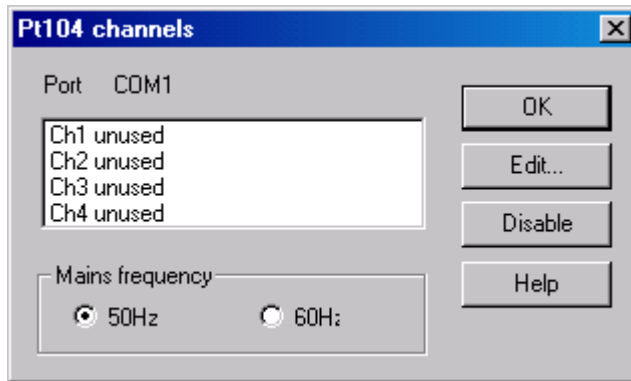


3. Click OK. The Converter details dialog box appears:



4. From the Converter type drop-down list, select PT-104.

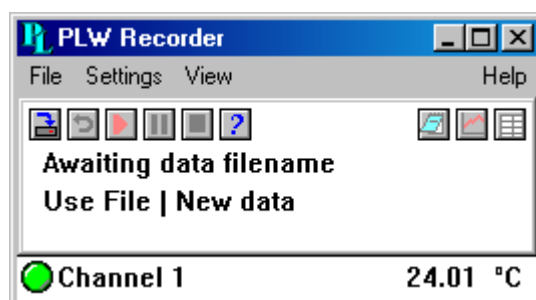
- From the Port drop-down list, select the port and click OK. After a short while, the PT-104 channels window appears:



- In the PT-104 channels window, double-click on Ch1 unused. The Edit PT-104 Channel dialog box appears:



- Type in a name for the channel, if required
- Select the data type required i.e temperature, resistance or voltage
- Select circuit - for PT100 and PT1000
- Now click OK. Readings from the PT-104 should appear in the monitor window:



2.4 Background on PRTs

PRTs (Platinum Resistance Thermometers) offer excellent accuracy over a wide temperature range (from -200 to 850°C). Sensors are interchangeable between different manufacturers, and are available in various accuracy ratings in packages to suit most applications. Unlike thermocouples, it is not necessary to use special cables to connect to the sensor.

The principle of operation is to measure the resistance of a platinum element. The most common type (PT100) has a resistance of 100 Ω at 0°C and 138.4 Ω at 100°C.

The relationship between temperature and resistance is approximately linear over a small temperature range: for example, if you assume that it is linear over the 0 to 100°C range, the error at 50°C is 0.4°C. For precision measurement, it is necessary to linearise the resistance to give an accurate temperature. The most recent definition of the relationship between resistance and temperature is International Temperature Standard 90 (ITS-90). This linearisation is done automatically with software.

The linearisation equation is:

$$R_t = R_0 * (1 + A * t + B * t^2 + C * (t - 100) * t^3)$$

$$A = 3.9083 \text{ E-3}$$

$$B = -5.775 \text{ E-7}$$

$$C = (\text{below } 0^\circ\text{C}) -4.183 \text{ E-12}$$

$$(\text{Above } 0^\circ\text{C}) \text{ zero}$$

For a PT100 sensor, a 1°C temperature change will cause a 0.384 Ω change in resistance, so even a small error in measurement of the resistance (for example, the resistance of the wires leading to the sensor) can cause a large error in the measurement of the temperature. For precision work, sensors have four wires - two to carry the sense current, and two to measure the voltage across the sensor element. It is also possible to obtain three-wire sensors, although these operate on the (not necessarily valid) assumption that the resistance of each of the three wires is the same.

The current through the sensor will cause some heating. For example, a sense current of 245 μA through a 100 Ω resistor will generate 6 μW of heat. If the sensor element is unable to dissipate this heat, it will report an artificially high temperature. This effect can be reduced by either using a large sensor element, or by making sure that it is in good thermal contact with its environment.

Using a 1 mA sense current will give a signal of only 100 mV. Because the change in resistance for a degree Celsius is very small, even a small error in the measurement of the voltage across the sensor will produce a large error in the temperature measurement. For example, a 100 μV voltage measurement error will give a 0.4°C error in the temperature reading. Similarly, a 1 μA error in the sense current will give 0.4°C temperature error.

Because of the low signal levels, it is important to keep any cables away from electric cables, motors, switchgear and other devices that may emit electrical noise. Using screened cable, with the screen grounded at one end, may help to reduce interference. When using long cables, it is necessary to check that the measuring equipment is capable of handling the resistance of the cables. Most equipment can cope with up to 100 Ω per core.

The type of probe and cable should be chosen carefully to suit the application. The main issues are the temperature range and exposure to fluids (corrosive or conductive) or metals. Clearly, normal solder junctions on cables should not be used at temperatures above about 170°C.

Sensor manufacturers offer a wide range of sensors that comply with BS1904 class B (DIN 43760). These sensors offer an accuracy of $\pm 0.3^\circ\text{C}$ at 0°C. For increased accuracy, you can use BS1904 class A ($\pm 0.15^\circ\text{C}$) or tenth-DIN sensors ($\pm 0.03^\circ\text{C}$). Companies like Isotech can provide standards with 0.001°C accuracy. Please note that these accuracy specifications relate to the SENSOR ONLY: it is necessary to add on any error in the measuring system as well.

Related standards are IEC751 and JISC1604-1989. IEC751 also defines the colour coding for PRT sensor cables: the one or two wires attached to one end of the sensor are red, and the one or two wires at the other end are white.

3 Driver information

3.1 Introduction

The PT-104 is supplied with driver routines that you can build into your own programs.

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 PDF file.

The driver routine is supplied as a Windows DLL.

The Windows DLL can be used with C, 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.

These are the routines in the driver:

- | | |
|--|---|
| ● pt104_open_unit | Open the device on a specified serial port. |
| ● pt104_close_unit | Close the port (do this each time you finish using the device!) |
| ● pt104_poll_driver | Poll the driver (not usually necessary). |
| ● pt104_get_cycle | Find out when the driver has taken a new set of readings. |
| ● pt104_set_channel | Specify the sensor type and filtering for a channel. |
| ● pt104_set_mains | Specify the mains setting 50 or 60 Hz. This enables the product to reject mains-frequency interference signals. |
| ● pt104_get_value | Get the most recent data reading from a channel. |
| ● pt104_get_version | Get the version number of this PT-104. |
| ● pt104_get_unit_info | Get the version number, batch number and calibration number and date of this PT-104. |
| ● pt104_get_driver_version | Get the version number of the driver. |
| ● pt104_labview | Easy to use interface for LabVIEW. |

The normal calling sequence for these routines is as follows:

1. Open driver
2. Set Channels
3. While you want to read data
4. Get data
5. End While
6. Close Unit
7. Close Driver

3.2 pt104_open_unit

```
short pt104_open_unit ( short port );
```

This routine specifies the serial port number with a PT-104 unit. If you wish to use more than one PT-104, you should call the routine once for each PT-104.

Arguments:	
<code>port</code>	- the port must be 1 for COM1, 2 for COM2, and so on
Returns:	
<code>TRUE</code>	- if the driver successfully opens the PT-104

3.3 pt104_close_unit

```
void pt104_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 `pt104_close_unit` for each port before you exit from your program. If you do not, your computer may misbehave until you next reboot it.

Arguments:	
<code>port</code>	- the port number. The port must be 1 for COM1, 2 for COM2, and so on
Returns:	
<code>none</code>	

3.4 pt104_poll_driver

```
void pt104_poll_driver (void);
```

It is not normally necessary to call this routine, as the driver uses the timer to poll the PT-104. Some programs, like Excel, appear to block the timer, so it is necessary to poll the driver periodically whilst waiting for data.

Arguments:	
<code>none</code>	
Returns:	
<code>none</code>	

3.5 pt104_get_cycle

```
unsigned short pt104_get_cycle(
    unsigned long far * cycle,
    unsigned short    port);
```

This routine returns the number of complete cycles of readings taken from a specified PT-104.

When you call [pt104_get_value](#), 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 [pt104_get_value](#) only when a cycle has completed.

Note: Each PT-104 is polled independently, so the cycle numbers for multiple PT-104s may not keep in step.

Arguments:	
<code>cycle</code>	- a pointer to an array where the cycle numbers are stored
<code>port</code>	- the port number. The port must be 1 for COM1, 2 for COM2, and so on
Returns:	
	- a value representing the number of complete cycles of readings taken from a particular PT-104

3.6 pt104_set_channel

```
short pt104_set_channel (
    unsigned short port,
    unsigned short channel,
    unsigned short data_type
    unsigned short no_of_wires);
```

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 [pt104_open_unit](#).

The fewer channels are selected, the more frequently these channels will be updated. It takes about 1 second per active channel.

If a call to `pt104_set_channel` has a `data_type` of single ended, then the channel specified's 'sister' channel is also enabled eg, enabling 3 will also enable 7.

Arguments:	
<code>channel</code>	- specifies which channel you want to set the details for. It should be between 1 and 4 if using single ended inputs in voltage mode.
<code>data_type</code>	- set to the type of reading you require (1 for PT100, 2 for PT1000, 3 for resistance 0 to 500 R, 4 for resistance 0 to 10 k, 5 for differential voltage 0 to 100 mV, 6 for differential voltage 0 to 2.5 V, 7 for single ended voltage 0 to 100 mv and 8 for single-ended voltage 0 to 2.5 V
<code>no_of_wires</code>	- specifies how many wires the PT100 or PT1000 has (set to 2, 3 or 4)

Returns:	
TRUE	- if the channel was successfully set

3.7 pt104_set_mains

```
void pt104_set_mains (
    unsigned short sixty_hertz);
```

This routine is used to select the mains frequency of 50 or 60 Hz.

Choosing the incorrect frequency may increase susceptibility to electrical noise.

Arguments:	
sixty_hertz	- for 50 Hz set this to 0
z	- for 60 Hz set this to 1
Returns:	
none	

3.8 pt104_get_value

```
short pt104_get_value (
    long far * data,
    unsigned short port,
    unsigned short channel,
    unsigned short filtered);
```

Once you open the driver and define some channels, the driver begins to take continuous readings from the PT-104. When you call this routine, it immediately sets data to the most recent reading for the specified channel.

Temperatures are returned in thousandths of a degree Celsius, voltages in the 2.5 V range are returned in tens of nanovolts (2.5 V returned as 250 000 000 * 10 nV), voltages in the 115 mV range are returned in nanovolts and resistances in milliohms.

When measuring single-ended voltages:

```
connector 1 pin 2 is channel 1
connector 2 pin 2 is channel 2
...
connector 1 pin 3 is channel 5
connector 2 pin 3 is channel 6
```

Arguments:	
data	- a pointer to an array where the sample values will be stored
port	- the port number. The port must be 1 for COM1, 2 for COM2, and so on
channel	1 - returns a reading from Channel 1 2 - returns a reading from Channel 2 3 - returns a reading from Channel 3 4 - returns a reading from Channel 4
filtered	- if set 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

Returns:	
TRUE FALSE	- if a reading is available - if any of the inputs are out of range. 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

3.9 pt104_get_version

```
unsigned short pt104_get_version (
    unsigned short far * version,
    unsigned short    port);
```

This routine sets version to version number of the specified PT-104.

Arguments:	
version	- driver version. The upper byte of the version is always 104 for a PT-104; the lower byte is the two hex digits of the version and release. It provides a useful check that the link to the PT-104 is working correctly
port	- the port number. The port must be 1 for COM1, 2 for COM2, and so on
Returns:	
TRUE	- if the parameters are in range

3.10 pt104_get_unit_info

```
short pt104_get_unit_info (
    char * str,
    unsigned short line,
    unsigned short port)
```

Call this routine to obtain information on the unit. `str` is set to the information specified by `line`.

Arguments:	
str	- a pointer to a char array that is to receive the information
line	- 0 for version number, 1 for calibration number, 2 for calibration date, 3 for batch number
port	- the port number. The port must be 1 for COM1, 2 for COM2, and so on
Returns:	
	- the number of bytes written to <code>str</code>

3.11 pt104_get_driver_version

```
short pt104_get_driver_version (void)
```

Arguments:	
none	

Returns:	
	- the version number of the current driver

3.12 pt104_labview

```
short pt104_labview (
    float          * result,
    unsigned short port,
    unsigned short channel,
    unsigned short data_type,
    unsigned short filtered,
    unsigned short no_of_wires)
```

This is an easy-to-use interface for LabVIEW. To gain a thorough understanding of the capabilities of this routine, see the example program supplied with the software.

Arguments:	
<code>result</code>	- a pointer to a float that is to receive the result
<code>port</code>	- the port number. The port must be 1 for COM1, 2 for COM2, and so on
<code>channel</code>	- 1 for Channel 1, 2 for Channel 2, 3 for Channel 3, and so on
<code>data_type</code>	- set to the type of reading you require (1 for PT100, 2 for PT1000, 3 for resistance 0 to 500 R, 4 for resistance 0 to 10 k, 5 for differential voltage 0 to 100 mV, 6 for differential voltage 0 to 2.5 V, 7 for single ended voltage 0 to 100 mV and 8 for single ended voltage 0 to 2.5 V)
<code>filtered</code>	- if set to TRUE, the driver returns a low-pass filtered value of the temperature. The time constant of the filter depends on the value of <code>filter_factor</code> for this channel, and on how many channels are active
<code>no_of_wires</code>	- specifies how many wires the PT100 or PT1000 has (set to 2, 3 or 4)
Returns:	
1	- if successful;
0	- if unsuccessful

3.13 Windows XP (SP2)/Vista

The 32-bit Windows driver is the file `pt10432.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`.

3.14 Linux

At time of release, Linux is not supported. Please check <http://www.picotech.com/drivers.html> for updates.

4 Writing your own programs

4.1 C / C++

C

You should find the following files in the directory where your software is installed:

`pt104tes.c` - Demonstration file.
`pt10432.dll` - Windows 32-bit driver.
`pt104.h` - header file

Producing a library file (`.lib`)

For Borland and Watcom C, Microsoft Visual C version 1.5 or lower, use the `implib` program that comes supplied with these compilers.

The command is:

```
Implib pt10432.lib pt10432.dll
```

This command is typed in at the command prompt - make sure you are in the correct directory or the `pt10432.dll` file will not be found. This will create a file called `pt10432.lib`.

The following steps are then required to use the drivers in your program:

1. Include the `pt10432.lib` in your program
2. Include the file `pt104.h` in the source file(s) of your program.

The `pt104tes.c` file can be used to demonstrate using the PT-104 driver program within a Windows application. You will also need to include the resource file `pt104tes.rc`.

For Microsoft Visual C versions 2, 4 and 5, Microsoft no longer supply `implib`. Furthermore, the names used in these versions of C are decorated - there is a prefix which indicates how many bytes are transferred to the routine as parameters. As a result, the C names do not match the names in the DLL. The Microsoft tools to alias decorated to undecorated names do not appear to work, so it is therefore necessary to use ordinal linking- linking by number, rather than name. The file `pt104ms.lib` (supplied with examples) contains all the necessary routines. To use this file:

1. Include the `pt104ms.lib` in your project
2. Include the file `pt104.h` in the C source file(s) of your program.

C++

C++ programs can access all versions of the driver. If `pt104.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.

4.2 Delphi

The `win` subdirectory contains a simple program `pt104.dpr` which opens the drivers and reads temperatures from two channels. You will need the following files to build a complete program:

```
pt104.dpr      - Project file
pt104fm.dfm   - Delphi form file
pt104fm.pas   - Delphi Pascal unit
pt104.inc     - Procedure prototypes for driver routines
```

The file `pt104.inc` contains procedure prototypes for the driver routines: you can include this file in your application.

This example has been tested with Delphi version 2.

4.3 Excel

The easiest way to transfer 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 `pt104xx.dll` to read in a set of data values. The Excel Macro language is similar to Visual Basic.

Excel 7

The example `pt10432.XLS` reads in 20 values from the unit on channel 1 at one per second, and assigns them to cells A4 to A24.

4.4 LabVIEW

The routines described here were tested using LabVIEW under Windows 98 version 4.0.

Although it is possible to access all of the driver routines described earlier, it is easier to use the special LabVIEW access routine. The `pt104.llb` library in the `DRIVERS` subdirectory shows how to access this routine.

To use this routine, copy `pt104.llb` and `pt10432.dll` to your LabVIEW `user.lib` directory.

You will then use the PT-104 sub-vi, and an example sub-vi which demonstrates 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 4) the PRT type (1 for PT100, 2 for PT1000). The sub-vi returns a temperature for the PRT.

4.5 Visual Basic

The `DRIVERS` subdirectory contains the following files:

- `pt10432.VBP` - Visual basic project file
- `pt10432.BAS` - Contains procedure prototypes for driver routines
- `pt10432.FRM` - Visual basic form

4.6 Agilent VEE

The routine described here was tested using Agilent-VEE version 5 under Windows 98.

The `DRIVERS` subdirectory contains the following files:

- `pt10432.dll` - 32-bit driver file
- `pt104.vee` - Agilent-VEE file
- `pt104.vh` - Procedure prototypes for driver

To use the example, open the file `pt104.vee` in Agilent-VEE and edit the import library icon so:

- File Name path is set to the location of `pt10432.dll`
- Definition File path is set to the location of `pt104.vh`

The example program collects 1000 readings from the PT-104 and displays them on an x-y trace.

5 Technical reference

5.1 Serial port settings

2400 baud, 8 data, 1 stop, no parity.

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	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.

5.2 Serial port connections

The information presented here is necessary only if you wish to connect the PT-104 to the PC in some unusual way (for example, via a radio modem).

The PT-104 uses the following RS232 data lines (pin connections as on PT-104).

Pin	Name	Usage
3	TX	Data from the PC to the PT-104
2	RX	Data from the PT-104 to the PC
7	RTS	Held at a positive voltage (>7V) to power the PT-104
5	GND	0 V line
4	DTR	Held at a negative voltage (<-7V) to power the PT-104

The driver powers up the PT-104 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 PT-104.

When the PT-104 is first powered up, it sends a version response (see [Protocol information](#)).

5.3 Protocol information

Requests

The computer can send the following request to the PT-104.

Command	Data bytes	Function
0x00	-	Get Version
0x01	-	Read EEPROM
0x02	one byte, bit 0 is LSB	Start Converting
	bit 0: enable channel 1	0 = off, 1 = on
	bit 1: enable channel 2	ditto
	bit 2: enable channel 3	ditto
	bit 3: enable channel 4	ditto
	bit 4: channel 1 gain	0 = x1, 1 = x21
	bit 5: channel 2 gain	ditto
	bit 6: channel 3 gain	ditto
	bit 7: channel 4 gain	ditto
0x03	bit 0: select 50/60 Hz mains	0 = 50 Hz, 1 = 60 Hz

Responses

At start-up, and on receipt of a version request, the PT-104 sends a version response.

Byte	Value	
1	xFF	
2	xAA	
3	x55	
4	x68	Product type
5	x10	Version

On receipt of a Start Converting request, the PT-104 starts sending conversion responses - approx one every 180 milliseconds. There are four measurement points for each channel, and the PT-104 automatically cycles through each of the measurements for each of the active channels. Each response is as follows:

Byte	Value
1	bits 0-1: measurement no. (0 to 3)
	bits 2-3: channel no. (0 to 3)
	bits 4-7: always zero
2-5	reading data: byte 2 is MSB
	0x20000000 = scaled minimum
	0xE0000000 = scaled maximum
	The result will always be in the range 0x20000000 to 0xE0000000 inclusive

On receipt of a Read EEPROM request, the unit returns 64 bytes of EEPROM data. This data contains the following:

Byte	Size
1	Checksum (always 0x55AB)
3	Calibration version (1 = current)

4	Spare
5	Calibration date (ddmmyy followed by NULL char)
13	Batch number
19	Calibration for channel 1 (resistance x 1E6)
23	Calibration for channel 2
27	Calibration for channel 3
31	Calibration for channel 4

Examples

```
GAIN = 21
MAXINPUT = 2500000
FSD = 0x10000000
SCALED_MIN = 0x20000000
```

To obtain a single ended voltage reading 0 - 115 mV
0x0A in byte 1 is measurement no 2 on channel 2.

Reading data in bytes 2-5 gives a value of 0x50000000.

Then:

$$\begin{aligned} \text{result} &= ((0x50000000 - \text{SCALED_MIN}) * \text{MAXINPUT}) / (\text{GAIN} * \text{FSD}) \\ &= 357142.857142857142857142857142857 \end{aligned}$$

where

MAXINPUT = 2,500,000 is the maximum input voltage in microvolts,
GAIN is a fixed gain factor,
FSD = 0x10000000 for a 28-bit ADC

To convert the result to mV divide by 10,000.0:

$$= 35.7142$$

To obtain a single ended voltage reading 0 - 2.5 V
using the same reading.

0x0A in byte 1 is measurement no 2 on channel 2.

Reading data in bytes 2-5 giving a value of 0x50000000.

To convert to voltage:

$$\text{result} = (0x50000000 - \text{SCALED_MIN}) * \text{MAXINPUT} / \text{FSD}$$

where **MAXINPUT** is the maximum input voltage in microvolts.

To convert result to V divide result by 10,000,000.0

To obtain a differential reading. 0 - 2.5V

Take two measurements on a channel i.e measurement 3 and 2 on channel 2.

Then:

$$((\text{measurement 3} - \text{measurement 2}) * \text{MAXINPUT}) / \text{FSD}$$

To convert result to V divide answer by 10,000,000.0

To calculate a resistance

Read the EEPROM to obtain the calibration information for the channels.

Take measurement 0, 1, 2 and 3 on a channel.

Then:

$$\text{result} = (\text{channel calibration} * (\text{measurement 3} - \text{measurement 2})) / (\text{measurement 1} - \text{measurement 0})$$

To convert to a resistance divide result by 1,000,000.0

To calculate a temperature

Carry out resistance measurement

Use a [lookup table](#) to convert the resistance to a temperature.

5.4 Modem operation

The PT-104 is normally connected directly to the computer, but it is also possible to access the PT-104 via a modem using the Windows driver.

It is necessary to provide power to the PT-104, either by instructing the modem to provide power or by connecting a power supply directly to the PT-104. See [Serial port connections](#) for information.

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

5.5 Lookup table

Temp	Resistance
-50	80.306282
-49	80.703340
-48	81.100257
-47	81.497036
-46	81.893677
-45	82.290179
-44	82.686545
-43	83.082774
-42	83.478868
-41	83.874827
-40	84.270652
-39	84.666343
-38	85.061901
-37	85.457327
-36	85.852622
-35	86.247785
-34	86.642818
-33	87.037721
-32	87.432495
-31	87.827140

-30	88.221657
-29	88.616046
-28	89.010309
-27	89.404445
-26	89.798455
-25	90.192339
-24	90.586099
-23	90.979734
-22	91.373246
-21	91.766634
-20	92.159898
-19	92.553041
-18	92.946061
-17	93.338960
-16	93.731737
-15	94.124394
-14	94.516930
-13	94.909346
-12	95.301643
-11	95.693820
-10	96.085879
-9	96.477819
-8	96.869641
-7	97.261345
-6	97.652931
-5	98.044401
-4	98.435753
-3	98.826989
-2	99.218109
-1	99.609112
0	100.000000
1	100.390772
2	100.781429
3	101.171970
4	101.562396
5	101.952706
6	102.342901
7	102.732980
8	103.122944
9	103.512792
10	103.902525
11	104.292142
12	104.681644
13	105.071030
14	105.460301
15	105.849456
16	106.238496
17	106.627420
18	107.016229
19	107.404922
20	107.793500
21	108.181962
22	108.570309
23	108.958540
24	109.346656
25	109.734656
26	110.122541
27	110.510310

28	110.897964
29	111.285502
30	111.672925
31	112.060232
32	112.447424
33	112.834500
34	113.221461
35	113.608306
36	113.995036
37	114.381650
38	114.768149
39	115.154532
40	115.540800
41	115.926952
42	116.312989
43	116.698910
44	117.084716
45	117.470406
46	117.855981
47	118.241440
48	118.626784
49	119.012012
50	119.397125
51	119.782122
52	120.167004
53	120.551770
54	120.936421
55	121.320956
56	121.705376
57	122.089680
58	122.473869
59	122.857942
60	123.241900
61	123.625742
62	124.009469
63	124.393080
64	124.776576
65	125.159956
66	125.543221
67	125.926370
68	126.309404
69	126.692322
70	127.075125
71	127.457812
72	127.840384
73	128.222840
74	128.605181
75	128.987406
76	129.369516
77	129.751510
78	130.133389
79	130.515152
80	130.896800
81	131.278332
82	131.659749
83	132.041050
84	132.422236
85	132.803306

86	133.184261
87	133.565100
88	133.945824
89	134.326432
90	134.706925
91	135.087302
92	135.467564
93	135.847710
94	136.227741
95	136.607656
96	136.987456
97	137.367140
98	137.746709
99	138.126162
100	138.505500
101	138.884722
102	139.263829
103	139.642820
104	140.021696
105	140.400456
106	140.779101
107	141.157630
108	141.536044
109	141.914342
110	142.292525
111	142.670592
112	143.048544
113	143.426380
114	143.804101
115	144.181706
116	144.559196
117	144.936570
118	145.313829
119	145.690972
120	146.068000
121	146.444912
122	146.821709
123	147.198390
124	147.574956
125	147.951406
126	148.327741
127	148.703960
128	149.080064
129	149.456052
130	149.831925
131	150.207682
132	150.583324
133	150.958850
134	151.334261
135	151.709556
136	152.084736
137	152.459800
138	152.834749
139	153.209582
140	153.584300
141	153.958902
142	154.333389
143	154.707760

144	155.082016
145	155.456156
146	155.830181
147	156.204090
148	156.577884
149	156.951562
150	157.325125
151	157.698572
152	158.071904
153	158.445120
154	158.818221
155	159.191206
156	159.564076
157	159.936830
158	160.309469
159	160.681992
160	161.054400
161	161.426692
162	161.798869
163	162.170930
164	162.542876
165	162.914706
166	163.286421
167	163.658020
168	164.029504
169	164.400872
170	164.772125
171	165.143262
172	165.514284
173	165.885190
174	166.255981
175	166.626656
176	166.997216
177	167.367660
178	167.737989
179	168.108202
180	168.478300
181	168.848282
182	169.218149
183	169.587900
184	169.957536
185	170.327056
186	170.696461
187	171.065750
188	171.434924
189	171.803982
190	172.172925
191	172.541752
192	172.910464
193	173.279060
194	173.647541
195	174.015906
196	174.384156
197	174.752290
198	175.120309
199	175.488212
200	175.856000

Index

A

Agilent VEE 18

C

C++ 16

Connecting the PT-104 5

Connections 19

D

Delphi 17

Driver routines

pt104_close_unit 11

pt104_get_cycle 12

pt104_get_driver_version 14

pt104_get_unit_info 14

pt104_get_value 13

pt104_get_version 14

pt104_labview 15

pt104_open_unit 11

pt104_poll_driver 11

pt104_set_channel 12

pt104_set_mains 13

E

Excel 17

I

Installation 4

L

LabVIEW 17

Legal information 3

Linux 15

Lookup table 22

M

Modem operation 22

P

Principles of operation 8

Protocol 20

pt104_close_unit 11

pt104_get_cycle 12

pt104_get_driver_version 14

pt104_get_unit_info 14

pt104_get_value 13

pt104_get_version 14

pt104_labview 15

pt104_open_unit 11

pt104_poll_driver 11

pt104_set_channel 12

pt104_set_mains 13

S

Safety warning 2

Serial port settings 19

Specifications 4

V

Visual Basic 18

W

Windows XP (SP2)/Vista 15

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

pt104.en 2.10.07

Copyright 2005 Pico Technology Limited. All rights reserved.