HumidiProbe

Temperature and Humidity Sensor

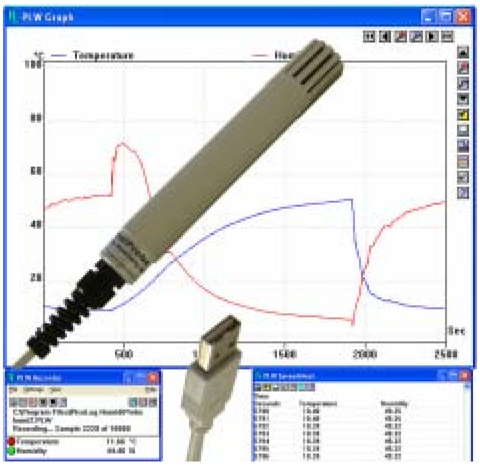
User's Guide

humidiprobe.en-5

Contents

1 Introduction

1.1 About HumidiProbe



HumidiProbe is a complete, self-contained temperature and humidity measuring device. It is calibrated to give highly accurate readings, which it takes every two seconds from a dual-purpose humidity and temperature sensor. Its USB connector is compatible with standard PCs and laptops

The PicoLog data logging program will collect data supplied by the HumidiProbe, or you can use the HumidiProbe driver software to custom-build your own programs that take advantage of its features.

PicoLog supports up to 20 HumidiProbe units.

1.2 Intended use

HumidiProbe is suitable for measuring temperature and relative humidity in benign environments, particularly indoors. You must not expose the unit to temperatures outside the maximum range (see Specifications).

The driver software, which runs under Microsoft Windows XP (SP2 or later), Windows Vista or Windows 7, contains everything necessary to convert the sensor readings into temperature and relative humidity.

1.3 This document

This document describes the physical and electrical properties of the HumidiProbe, and explains how to use the software drivers.

For more information about using HumidiProbe with PicoLog, please consult the following manual:

- PicoLog User's Guide

2 Legal notices

2.1 CE notice

The HumidiProbe meets the intent of EMC directive 89/336/EEC and meets the EN61326-1 (1997) Class A Emissions and Immunity standard.

The HumidiProbe also meets the intent of the Low Voltage Directive and meets the BS EN 61010-1:2001 IEC 61010-1:2001 (safety requirements for electrical equipment, control, and laboratory use) standard.

2.2 FCC notice

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his or her own expense.

Repair information

2.3 License conditions

Grant of license. The material contained in this release is licensed, not sold. Pico Technology Limited ('Pico') 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 and agree to abide by these conditions.

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

Copyright. Pico claims the copyright of and reserves the rights to all material (software, documents etc) contained in this release.

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

Fitness for purpose. No two applications are the same, so Pico cannot guarantee that its equipment or software is suitable for a given application. It is therefore the user's responsibility to ensure that the product is suitable for the user's application.

Mission-critical applications. Because the software runs on a computer that may be running other software products, and may be subject to interference from these other products, this license specifically excludes usage in 'mission-critical' applications, for example life-support systems.

Viruses. This software was continuously monitored for viruses during production. However, the user is responsible for virus checking the software once it is installed.

Support. No software is ever error-free, but if you are dissatisfied with the performance of this software, please contact our technical support staff.

2.4 Warranty

Pico Technology warrants upon delivery, and for a period of 24 months unless otherwise stated from the date of delivery, that the Goods will be free from defects in material and workmanship.

Pico Technology shall not be liable for a breach of the warranty if the defect has been caused by fair wear and tear, wilful damage, negligence, abnormal working conditions or failure to follow Pico Technology's spoken or written advice on the storage, installation, commissioning, use or maintenance of the Goods or (if no advice has been given) good trade practice; or if the Customer alters or repairs such Goods without the written consent of Pico Technology.

2.5 Repairs

The unit contains no user-serviceable parts. Repair or calibration of the unit requires specialized test equipment and must be performed by Pico Technology Limited or its authorized distributors.

2.6 Trademarks

Borland, Delphi and Turbo Pascal are trade marks or registered trade marks of Borland International, Inc.

Windows, Excel and Visual Basic are registered trademarks or trademarks of Microsoft Corporation in the USA and other countries.

National Instruments and LabVIEW are trademarks of National Instruments Corporation, registered in the United States and other countries.

Agilent VEE is a registered trade mark of Agilent Technologies.

Pico Technology Limited, PicoLog and HumidiProbe are trademarks of Pico Technology Limited, registered in the United Kingdom and other countries.

3 Using HumidiProbe

3.1 Specifications

Humidity	
Range	0 %RH to 100 %RH (non-condensing)
Accuracy	10 %RH to 90 %RH ±2 % 0 %RH to 100 %RH ±4 %
Resolution	0.05 %RH
Response time	8 seconds
Temperature	
Range	0 to +70 °C
Accuracy	10 to 40 °C ±0.3 °C 0 to 70 °C ±1 °C
Resolution	0.01 °C
Response time	5 to 30 seconds
Electrical	
Conversion time	2 seconds (humidity and temperature)
Interface	USB 1.1 (compatible with USB 2.0)
Connector	Integrated USB cable and connector
Enclosure	
Dimensions	Ø 22 mm, length 170 mm (approx. 4.5 m including cable)
Material	Grey ABS
Protection	NOT waterproof

3.2 Connecting HumidiProbe (quick guide)

Important note:
Always install the PicoLog software BEFORE
connecting your HumidiProbe to the computer for the first time.

Once the software is installed, connect the HumidiProbe to the USB port on your computer using the integrated cable and connector.

To set up the unit with PicoLog, do the following:

1. Open PicoLog Recorder.
2. Select New settings from the File menu.
3. In the Recording dialog box, click OK.
4. In the Sampling Rate dialog box, click OK.
5. In the Converter details dialog box, select HumidiProbe from the drop-down box and click OK.
6. In the HumidiProbe Channels dialog box, double-click on Temp unused.
7. In the Edit HumidiProbe Channel dialog box, click OK and PicoLog will activate the Temperature parameter.
8. In the HumidiProbe Channels dialog box, double-click on Humidity unused.
9. In the Edit HumidiProbe Channel dialog box, click OK and PicoLog will activate the Humidity parameter.
10. In the HumidiProbe Channels dialog box, click OK.
11. PicoLog Recorder should now display the temperature and humidity.

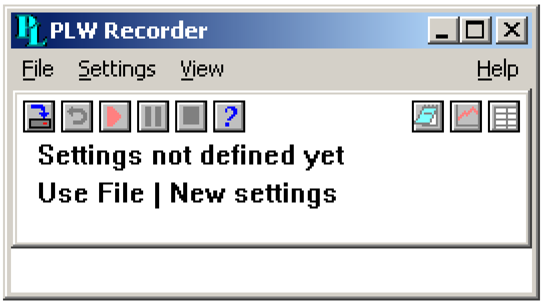
3.3 Connecting HumidiProbe (detailed guide)

Important note:
Always install the PicoLog software BEFORE
connecting your HumidiProbe to the computer for the first time.

Once the software is installed, connect the HumidiProbe to the USB port on your computer using the integrated cable and connector.

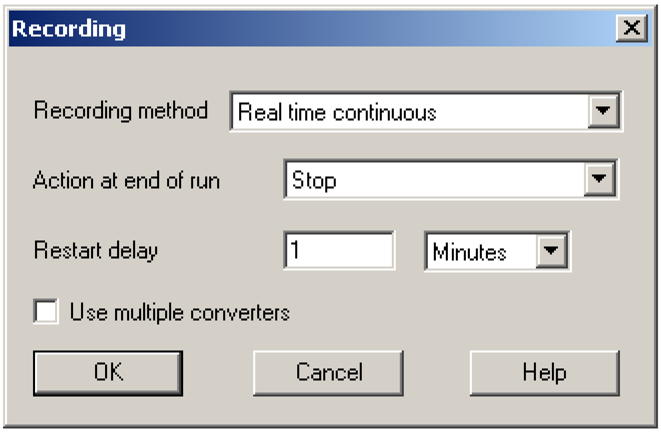
To set up the unit with PicoLog, do the following:

1. Open PicoLog Recorder.
   The application opens at the recorder view:



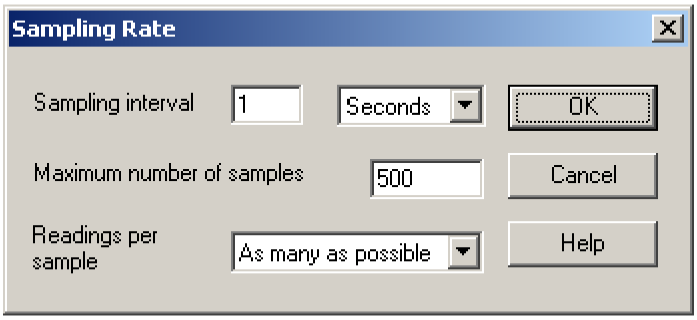
Settings not defined yet
Use File | New settings

2. Select New settings from the File menu.
   PicoLog displays the Recording dialog box:



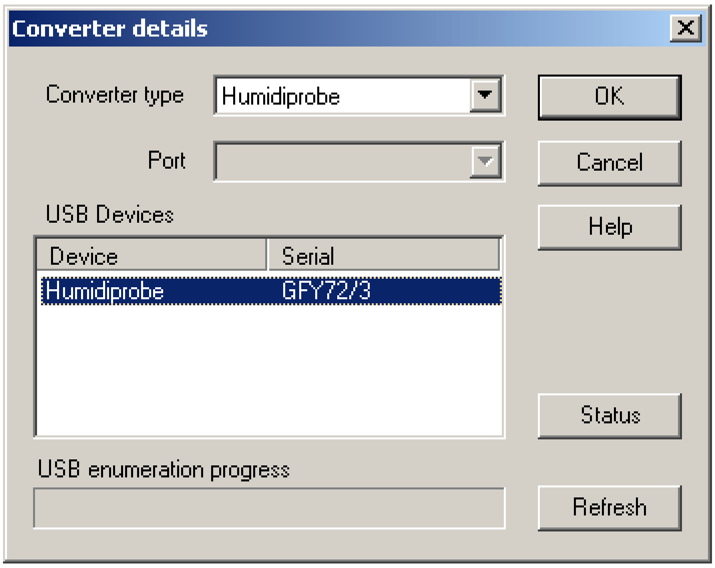
Recording method: Real time continuous
Action at end of run: Stop
Restart delay: 1 Minutes
□ Use multiple converters

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



Sampling interval: 1 Seconds
Maximum number of samples: 500
Readings per sample: As many as possible

4. Click OK.
   PicoLog displays the Converter details dialog box:



Converter type: Humidiprobe

Port:

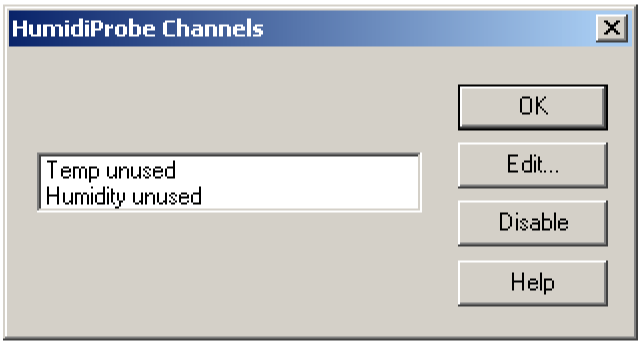
USB Devices

Device	Serial
Humidiprobe	GFY72/3

5. From the Converter type drop-down list, select Humidiprobe.

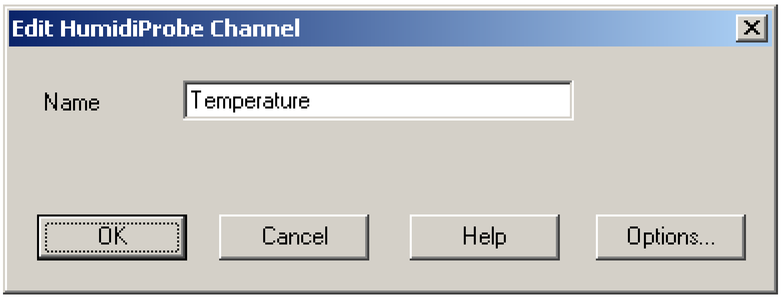
6. After a few seconds, the program will detect the HumidiProbe and list it under "USB Devices".

7. Click OK.
   The HumidiProbe Channels dialog box appears:



Temp unused
Humidity unused

8. Double-click on Temp unused.
   The Edit HumidiProbe Channel dialog box appears



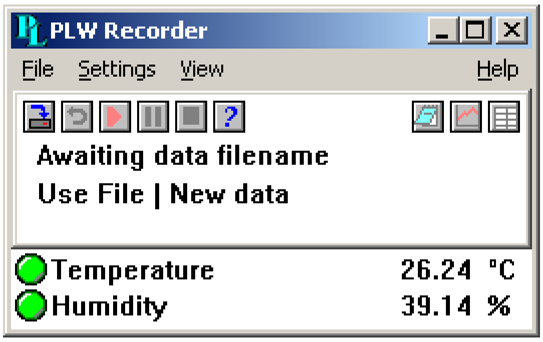
Name: Temperature

9. Click OK.
   PicoLog activates the Temperature parameter.

10. Double-click on Humidity unused.
    The Edit HumidiProbe Channel dialog box appears

11. Click OK.
    PicoLog activates the Humidity parameter.

12. At the HumidiProbe Channels dialog box, click OK.
    The recorder view should now display the temperature and humidity:



Awaiting data filename

Use File | New data

Temperature 26.24 °C

Humidity 39.14 %

4 Driver software

4.1 Introduction

To allow you to write your own programs for the HumidiProbe, we supply a Software Development Kit (SDK) containing drivers and example code. The drivers are for use with the following operating systems:

- Microsoft Windows XP (SP2 or later)
- Microsoft Windows Vista
- Microsoft Windows 7

The 32-bit Windows driver is supplied as a DLL, `HumidiProbe.dll`. This can be used with C, Delphi, LabVIEW, Agilent VEE 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 driver exports the following routines:

Routine	Function
`HumidiProbeOpenUnit`	Open a HumidiProbe unit
`HumidiProbeOpenUnitAsync`	Open a HumidiProbe unit without blocking the calling thread
`HumidiProbeOpenUnitProgress`	Check the progress of an asynchronous open operation
`HumidiProbeGetUnitInfo`	Obtain unit information in a character string
`HumidiProbeGetSingleValue`	Get the most recent temperature and relative humidity readings
`HumidiProbeCloseUnit`	Shut down a HumidiProbe unit

All routines are C functions using the standard call naming convention (`__stdcall`) and are exported with both decorated and undecorated names.

The normal calling sequence for these routines is as follows:

```
Open driver
While you want to measure temperatures:
   Get temperature and humidity
End while
Close driver
```

4.2 HumidiProbeOpenUnit

```
short HumidiProbeOpenUnit (void);
```

This routine opens a HumidiProbe unit. The driver can support up to 64 units. If you wish to use more than one HumidiProbe, call the routine once for each unit.

Arguments:	None	
Returns:	-1	If the unit fails to open
	0	If no unit is found
	> 0	Handle to the device opened

4.3 HumidiProbeOpenUnitAsync

```
short HumidiProbeOpenUnitAsync (void);
```

This routine opens a HumidiProbe unit without blocking the calling function.

Arguments:	None	
Returns:	0	If there is a previous open operation in progress.
	1	If the call has successfully initiated an open operation.

4.4 HumidiProbeOpenUnitProgress

```
short HumidiProbeOpenUnitProgress (
  short * handle,
  short * progress );
```

This routine checks the progress of an asynchronous open operation initiated by HumidiProbeOpenUnitAsync.

Arguments:	handle	A pointer to a short where the unit handle is to be written. On output: -1 if the unit fails to open 0 if no unit is found > 0 handle of device (valid only if function returns non-zero)
	progress	A pointer to a short where the percentage progress is to be written. 100% implies that open operation is complete
Returns:	<>0	If the driver successfully opens the HumidiProbe.
	0	If the operation failed

4.5 HumidiProbeGetSingleValue

```
short HumidiProbeGetSingleValue (
    short   handle,
    float * temp,
    short   filterTemp,
    float * humidity,
    short   filterHumidity);
```

This routine retrieves the temperature and relative humidity from the specified HumidiProbe unit. New readings are available every two seconds. If you call the routine more frequently than this, it will return old readings.

Arguments:	handle	The handle returned by HumidiProbeOpenUnit.
	temp	A pointer to the temperature from the driver in degrees Celsius. If temp is a null pointer, it is ignored.
	filterTemp	If 0, temp is unfiltered; if non-zero, temp is the median of a number of readings.
	humidity	A pointer to the relative humidity from the driver in percent. If humidity is a null pointer, it is ignored.
	filterHumidity	If 0, humidity is unfiltered; if non-zero, humidity is the median of a number of readings.

Returns:	HUMIDIPROBE_CR_NONE (0) if no new readings were taken; HUMIDIPROBE_CR_OLD_READING (1) if the temperature and humidity returned are old readings because the previous readings were less than 2 seconds old; HUMIDIPROBE_CR_NEW_READING (2) if the temperature and humidity are new readings.

4.6 HumidiProbeGetUnitInfo

```
short HumidiProbeGetUnitInfo (
   short   handle,
   char  * string,
   short   stringLength,
   short   info );
```

This routine obtains information from the driver about the specified HumidiProbe unit.

Arguments:	handle	The handle of the device for which information is required. If an invalid handle is passed, the error code from the last unit that failed to open is returned when info = HUMIDIPROBE_ERROR (7).
	string	A pointer to the character string buffer in the calling function where the unit information string (selected with info) will be stored. If a null pointer is passed, no information will be written.
	stringLength	The length of the character string buffer. If the string is not long enough to accept all of the information, only the first stringLength characters are returned.
	info	An enumerated type specifying what information is required from the driver.
Returns:	0	If one or more parameters is out of range, or a null pointer is passed for string.
	> 0	The length of the string written to the character string buffer, string.

Allowed values for info argument:

info	Description	Example
HUMIDIPROBE_DRIVER_VERSION (0)	The version of HumidiProbe.dll. A valid handle is optional when calling the function with info=0.	1.0.0.1
HUMIDIPROBE_USB_VERSION (1)	The type of USB port to which the HumidiProbe is connected.	1.1
HUMIDIPROBE_HARDWARE_VERSION (2)	The hardware version of the HumidiProbe attached.	2
HUMIDIPROBE_VARIANT_INFO (3)	The type of HumidiProbe attached.	1
HUMIDIPROBE_BATCH_AND_SERIAL (4)	The batch and serial number of the unit.	GFY72/3
HUMIDIPROBE_CAL_DATE (5)	The calibration date of the unit.	14Mar05
HUMIDIPROBE_KERNEL_DRIVER_VERSION (6)	The kernel driver version.	1.0
HUMIDIPROBE_ERROR (7)	An error code (see list of error codes).	4

4.6.1 HumidiProbeGetUnitInfo error codes

HumidiProbeGetUnitInfo will return one of the following error codes when called with info = HUMIDIPROBE_ERROR (7):

Error code	Symbol, Description
0	HUMIDIPROBE_OK The HumidiProbe is functioning correctly.
1	HUMIDIPROBE_KERNEL_DRIVER The kernel driver (picopp.sys) does not support this product.
2	HUMIDIPROBE_NOT_FOUND No HumidiProbe could be found.
3	HUMIDIPROBE_CONFIG_FAIL Unable to download firmware.
4	HUMIDIPROBE_INVALID_PARAMETERS Temperature and Humidity parameters are both null.
5	HUMIDIPROBE_CONVERSION_FAILED The unit tried to get a new temperature and humidity reading and failed.
6	HUMIDIPROBE_FW_FAIL The firmware could not be loaded.
7	HUMIDIPROBE_ERROR_OS_NOT_SUPPORTED The driver does not support this operating system.

4.7 HumidiProbeCloseUnit

```
void HumidiProbeCloseUnit (short handle);
```

This routine shuts down the specified HumidiProbe unit. You do not need to call this routine before exiting the application, as the driver will automatically shut down the unit. However, you can call it if you need to shut down the unit without exiting the application.

Arguments:	handle	The handle, returned by HumidiProbeOpenUnit, of the HumidiProbe to be closed.

Returns:	1	If a valid handle is passed.
	0	If not.

5 Writing your own programs

5.1 C / C++

C

The C program HProbe.c, provided as an example with the driver, is a generic Windows application - meaning it does not use Borland AppExpert or Microsoft AppWizard. To compile the program, create a new project for an application containing the following files:

- HPROBE.C
- Hprobe.rc

and either

- humidiprobebc.lib (Borland 32-bit applications)

or

- HumidiProbe.lib (Microsoft Visual C 32-bit applications)

The following files must be in the same directory:

- HumidiProbeApi.h
- HumidiProbe.dll (All 32-bit applications)

C++

C++ programs can access all versions of the driver. If HumidiProbeApi.h is included in a C++ program, the PREF1 macro expands to extern "C"; this disables "name decoration", as Microsoft calls it, and enables C++ routines to make calls to the driver routines using C headers.

5.2 Delphi

The Examples\HumidiProbe subdirectory contains HPROBE.DPR, a simple program which opens the drivers and reads temperature and humidity. You will need the following files to build a complete program.

- HPROBEFM.dfm
- HPROBEFM.pas
- hprobe.inc

hprobe.inc contains procedure prototypes for the driver routines. You can include this file in your application.

This example has been tested with Delphi version 3.

5.3 Excel

The easiest way to transfer data to 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 the driver to read in a set of data values. The Excel macro language is similar to Visual Basic.

HPROBE.XLS, an example file, reads in 20 values of the channel 1 temperature, one per second, and assigns them to cells A1..A20.

5.4 LabVIEW

The `Hprobe.vi` example shows how to access the driver functions using LabVIEW. To use the example, copy these files to your LabVIEW directory:

- `Hprobe.vi`
- `HumidiProbe.dll`

The example program displays the temperature in degrees Celsius and the relative humidity in percent.

5.5 Visual Basic

Use the following files from the SDK:

- `HPROBE.VBP`
- `HPROBE.BAS`
- `HPROBE.FRM`

5.6 Agilent VEE

Two example programs are supplied in the drivers directory: one for VEE 6, the other for VEE 7. They show how to collect readings continuously from the HumidiProbe and log the data to disk.

VEE 6

The following files are supplied:

- `HProbe6.vee`
- `humidiprobe.vh`

The code was tested using Agilent VEE version 6 under Windows.

VEE 7

The following files are supplied:

- `HProbe7.vee`
- `HumidiProbe.vh`

The code was tested using Agilent VEE version 7 under Windows.

Index

Pico Technology

James House
Colmworth Business Park
ST. NEOTS
Cambridgeshire
PE19 8YP
United Kingdom
Tel: +44 (0) 1480 396 395
Fax: +44 (0) 1480 396 296
www.picotech.com

humidiprobe.en-5

18.06.12