HumidiProbe

Temperature and Humidity Sensor

User's Guide
## Contents

1 Introduction ........................................................................................................................................1
   1 About HumidiProbe .................................................................1
   2 Intended use .............................................................................1
   3 This document ........................................................................1

2 Legal notices ..................................................................................................................................2
   1 CE notice ..................................................................................2
   2 FCC notice ................................................................................2
   3 License conditions .................................................................2
   4 Warranty ....................................................................................3
   5 Repairs .......................................................................................3
   6 Trademarks ................................................................................3

3 Using HumidiProbe .........................................................................................................................4
   1 Specifications .............................................................................4
   2 Connecting HumidiProbe (quick guide) ......................................6
   3 Connecting HumidiProbe (detailed guide) .................................7

4 Driver software ................................................................................................................................10
   1 Introduction ..............................................................................10
   2 HumidiProbeOpenUnit ...........................................................11
   3 HumidiProbeOpenUnitAsync ....................................................12
   4 HumidiProbeOpenUnitProgress ...............................................13
   5 HumidiProbeGetSingleValue .................................................14
   6 HumidiProbeGetUnitInfo .........................................................15
      1 HumidiProbeGetUnitInfo error codes ....................................16
   7 HumidiProbeCloseUnit ............................................................17

5 Writing your own programs .............................................................................................................18
   1 C / C++ .......................................................................................18
   2 Delphi .........................................................................................18
   3 Excel .........................................................................................18
   4 LabVIEW .....................................................................................19
   5 Visual Basic ..............................................................................19

Index .................................................................................................................................................20
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, Windows 7, and Windows 8* contains everything necessary to convert the sensor readings into temperature and relative humidity.

*Not Windows RT

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 PicoLog User’s Guide.
2 Legal notices

2.1 CE notice


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.

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.

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

## 3.1 Specifications

<table>
<thead>
<tr>
<th>Humidity</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Range</strong></td>
<td>0 %RH to 100 %RH (non-condensing)</td>
</tr>
<tr>
<td><strong>Accuracy</strong></td>
<td>Typical ±2 %RH @ 25 °C (see Figure 1)</td>
</tr>
<tr>
<td><strong>Resolution</strong></td>
<td>0.05 %RH</td>
</tr>
<tr>
<td><strong>Sensor response time</strong></td>
<td>8 seconds</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Temperature</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Range</strong></td>
<td>0 to +70 °C</td>
</tr>
<tr>
<td><strong>Accuracy</strong></td>
<td>Typical ±0.5 °C (see Figure 2)</td>
</tr>
<tr>
<td><strong>Resolution</strong></td>
<td>0.01 °C</td>
</tr>
<tr>
<td><strong>Sensor response time</strong></td>
<td>5 to 30 seconds</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Electrical</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Conversion time</strong></td>
<td>2 seconds (humidity and temperature)</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td>USB 1.1 (compatible with USB 2.0/3.0)</td>
</tr>
<tr>
<td><strong>Connector</strong></td>
<td>Integrated USB cable and connector</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enclosure</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dimensions</strong></td>
<td>Ø 22 mm, length 170 mm (approx. 4.5 m including cable)</td>
</tr>
<tr>
<td><strong>Material</strong></td>
<td>Grey ABS</td>
</tr>
<tr>
<td><strong>Protection</strong></td>
<td>NOT waterproof</td>
</tr>
</tbody>
</table>

*Time to reach 63% of a step function, valid at 25 °C and 1 m/s airflow.  
**Response time depends on air flow and operating conditions.

![Figure 1 - Maximum RH accuracy at 25 °C](image1)

![Figure 2 - Maximum temperature accuracy](image2)

### Operating conditions

Long term exposure to conditions outside normal range may temporarily offset the RH signal (+3 %RH after 60h). After return to normal range it will slowly return towards the calibration state by itself. Prolonged exposure to extreme conditions may accelerate ageing.
The sensor shall not get in close contact with volatile chemicals such as solvents or other organic compounds. Especially high concentration and long exposure must be avoided. Ketenes, acetone, ethanol, isopropyl alcohol, toluene, etc. are known to cause drift of the humidity reading – irreversibly in most cases. Some plastics can cause outgassing, so do not store the sensor in a plastic bag, in bubblewrap or similar.

Application of HumidiProbe in harsh environments must be carefully tested and qualified. Pico Technology qualifies its humidity and temperature sensors to work properly within ambient clean air – qualification for use in harsh environments is the duty of the user of the product.

**Storage conditions and Handling instructions**

Chemical vapors at high concentration in combination with long exposure times may offset the sensor reading.

For these reasons it is recommended to store the Humidiprobe in original packaging or a cardboard box, in the following conditions: Temperature shall be in the range of 10 °C to 50 °C and humidity at 20 %RH to 60 %RH.
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:

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:

   The application opens at the recorder view:

   ![PLW Recorder](image)

   Settings not defined yet
   Use File | New settings

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

   ![Recording dialog box](image)

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

   OK Cancel Help

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

   ![Sampling Rate dialog box](image)

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

   OK Cancel Help

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

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

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:

![PLW Recorder](image)

**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
- Microsoft Windows 8*

*Note: Windows RT not supported.

The 32-bit Windows driver is supplied as a DLL, HumidiProbe.dll. This can be used with C, Delphi, LabVIEW 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:

<table>
<thead>
<tr>
<th>Routine</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>HumidiProbeOpenUnit</td>
<td>Open a HumidiProbe unit</td>
</tr>
<tr>
<td>HumidiProbeOpenUnitAsync</td>
<td>Open a HumidiProbe unit without blocking the calling thread</td>
</tr>
<tr>
<td>HumidiProbeOpenUnitProgress</td>
<td>Check the progress of an asynchronous open operation</td>
</tr>
<tr>
<td>HumidiProbeGetUnitInfo</td>
<td>Obtain unit information in a character string</td>
</tr>
<tr>
<td>HumidiProbeGetSingleValue</td>
<td>Get the most recent temperature and relative humidity readings</td>
</tr>
<tr>
<td>HumidiProbeCloseUnit</td>
<td>Shut down a HumidiProbe unit</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Arguments:</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns:</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>&gt; 0</td>
</tr>
</tbody>
</table>
4.3 **HumidiProbeOpenUnitAsync**

    short HumidiProbeOpenUnitAsync (void);

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

<table>
<thead>
<tr>
<th>Arguments:</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns:</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>
4.4 HumidiProbeOpenUnitProgress

```c
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

```c
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.

<table>
<thead>
<tr>
<th>Arguments:</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>handle</td>
<td>The handle returned by <code>HumidiProbeOpenUnit</code>.</td>
</tr>
<tr>
<td>temp</td>
<td>A pointer to the temperature from the driver in degrees Celsius. If <code>temp</code> is a null pointer, it is ignored.</td>
</tr>
<tr>
<td>filterTemp</td>
<td>If 0, <code>temp</code> is unfiltered; if non-zero, <code>temp</code> is the median of a number of readings.</td>
</tr>
<tr>
<td>humidity</td>
<td>A pointer to the relative humidity from the driver in percent. If <code>humidity</code> is a null pointer, it is ignored.</td>
</tr>
<tr>
<td>filterHumidity</td>
<td>If 0, <code>humidity</code> is unfiltered; if non-zero, <code>humidity</code> is the median of a number of readings.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Returns:</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HUMIDIPROBE_CR_NONE (0)</td>
<td>if no new readings were taken;</td>
</tr>
<tr>
<td>HUMIDIPROBE_CR_OLD_READING (1)</td>
<td>if the temperature and humidity returned are old readings because the previous readings were less than 2 seconds old;</td>
</tr>
<tr>
<td>HUMIDIPROBE_CR_NEW_READING (2)</td>
<td>if the temperature and humidity are new readings.</td>
</tr>
</tbody>
</table>
4.6  **HumidiProbeGetUnitInfo**

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

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

**Arguments:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>handle</td>
<td>The handle of the device for which information is required. If an invalid</td>
</tr>
<tr>
<td></td>
<td>handle is passed, the <strong>error code</strong> from the last unit that failed to open</td>
</tr>
<tr>
<td></td>
<td>is returned when info = HUMIDIPROBE_ERROR (7).</td>
</tr>
<tr>
<td>string</td>
<td>A pointer to the character string buffer in the calling function where the</td>
</tr>
<tr>
<td></td>
<td>unit information string (selected with info) will be stored. If a null</td>
</tr>
<tr>
<td></td>
<td>pointer is passed, no information will be written.</td>
</tr>
<tr>
<td>stringLength</td>
<td>The length of the character string buffer. If the string is not long enough</td>
</tr>
<tr>
<td></td>
<td>to accept all of the information, only the first stringLength characters</td>
</tr>
<tr>
<td></td>
<td>are returned.</td>
</tr>
<tr>
<td>info</td>
<td>An enumerated type specifying what information is required from the driver.</td>
</tr>
</tbody>
</table>

**Returns:**

<table>
<thead>
<tr>
<th>Return</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>If one or more parameters is out of range, or a</td>
</tr>
<tr>
<td></td>
<td>null pointer is passed for string.</td>
</tr>
<tr>
<td>&gt; 0</td>
<td>The length of the string written to the</td>
</tr>
<tr>
<td></td>
<td>character string buffer, string.</td>
</tr>
</tbody>
</table>

**Allowed values for info argument:**

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

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

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

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 SDK 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`
Index

A
Access  2
Accuracy  4

C
C / C++  18
CE notice  2
Connecting HumidiProbe (detailed guide)  7
Connecting HumidiProbe (quick guide)  6
Connector  4
Conversion time  4, 14
Copyright  2

D
Delphi  18
Dimensions  4
DLL  10
Driver  10

E
Enclosure  4
Error codes  16
Excel  18

F
FCC notice  2
Filtering  14
Fitness for purpose  2

H
Help files  1
HumidiProbeCloseUnit  17
HumidiProbeGetSingleValue  14
HumidiProbeGetUnitInfo  15, 16
HumidiProbeOpenUnit  11
HumidiProbeOpenUnitAsync  12
HumidiProbeOpenUnitProgress  13
Humidity range  4

I
Installation  6, 7
Intended use  1
Interface  4

L
LabVIEW  19
Liability  2
Licence conditions  2

M
Microsoft Windows  10
Mission-critical applications  2

O
Overview  1

P
PicoLog  1, 6, 7
Protection  4

R
Repairs  3
Resolution  4
Response times  4

S
SDK  10
Specifications  4

T
Temperature range  4
Trademarks  3

U
Usage  2
USB  4

V
Visual Basic  19

W
Warranty  3
Windows (Microsoft)  10