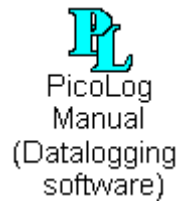


RH-02 Manual v1.0

1. Introduction
2. Connecting to PC
3. Principles of Operation
4. Specifications
5. Technical Information
6. Using USB ports



Writing your own software overview



Other Information

Contact



Pico

Technical



Support

Signal



Conditioners



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)

Legal information

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 before it is installed, but you may 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.

Fitness for purpose

No two applications are the same, so Pico Technology 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.

Because the software runs on a computer that may be running other software products, and may be subject to interference from these other applications, this license specifically excludes usage in 'mission critical' applications, for example medical or safety related applications

Trade Marks

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

Microsoft, MS-DOS, Windows, Windows NT, Visual C++, 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, PicoScope, PicoLog and DrDAQ are trademarks of Pico Technology Limited, registered in the United Kingdom and other countries.

Introduction

The PICO RH-02 is a complete temperature and humidity input device for use with IBM compatible computers. It connects to the serial port of the computer, and has built in sensors that are calibrated to give highly accurate readings.

It can be used with the PicoLog data logging program: alternatively, you can use the RH-02 driver software to develop your own programs to collect and analyse data from the unit.

PicoLog and the drivers support up to four RH-02 units under DOS, and nine RH-02 units under Windows.

This file describes the physical and electrical properties of the RH-02, and explains how to use the software drivers.

For more information about using the RH-02 with PicoLog, please consult the following files:

PicoLog for Windows	PLW044.HLP (Windows help file)
PicoLog for DOS	PL.TXT

Principles of operation

The RH-02 takes a reading every two seconds from each of two sensors- a humidity sensor and a precision thermistor. These sensors are built into the RH-02. The driver software contains everything necessary to convert the sensor readings into temperature and humidity.

Filtering

The RH-02 software includes a filter which eliminates errors caused by electrical noise. 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.

Specifications

Conversion Time		2 seconds (for both humidity & temp)
Resolution		0.0025%
Humidity	Range	0 to 100% (non condensing)
	Accuracy	±2% (5% to 95%)
	Response time	Approx 1 minute
Temperature	Range	0 to 70°C
	Accuracy	±0.2C (0 to 70°C)
		10 Seconds
Connector		D9 female to computer
Enclosure	Dimensions	130x60x30mm
	Material	Black ABS NOT waterproof

Connecting the RH-02

To use the RH-02, you should connect the D-connector on the RH-02 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.

To check that the unit is working:

1. start up PicoLog for Windows
2. Select **File**
3. Select **New Settings**
4. At the Recording dialog, press **OK**
5. At the Sampling dialog, press **OK**
6. Set the converter type to **RH02**
7. Select the COM port that you have connected the RH02 to
8. Press **OK**
9. At the RH02 channels dialog, double-click on **Temp unused**
10. At the Edit RH02 channel dialog, press **OK**
11. Back to the RH02 channels dialog, double_click on **Humidity unused**
12. At the Edit RH02 channel dialog, press **OK**
13. Back to the RH02 channels dialog, press **OK**
14. The recorder view should now display the temperature and humidity.

Driver summary

The RH-02 is supplied with driver routines that you can build short into your own programs. Drivers are supplied for use with the following operating systems:

- DOS
- Windows 3.1
- Windows 95/98
- Windows NT/2000
- Linux

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.

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
<code>rh02_open_unit</code>	Open the driver to use a specified serial port(s)
<code>rh02_close_unit</code>	Close the port (Warning! failure to do this under DOS could cause your computer to malfunction)

<code>rh02_get_cycle</code>	Find out when the driver has taken a new set of readings
<code>rh02_get_value</code>	Get the most recent temperature or humidity reading
<code>rh02_get_version</code>	get the version number of this RH-02

The normal calling sequence for these routines is as follows:

```
Open driver
While you want to measure temperatures,
    Get temperature and/or humidity
End While
Close Driver
```

rh02_open_unit

DOS version:

```
short int rh02_open_unit (
    short int port,
    short int base,
    short int irq);
```

Windows version:

```
short int rh02_open_unit (
    short int port);
```

This routine specifies the serial port number with an RH-02 unit. If you wish to use more than one RH-02, you should call the routine once for each RH-02.

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

Under DOS, this routine has extra parameters to specify the base address and short interrupt number for the COM port. These can be set to zero for the default base address and IRQ. See [Serial port settings](#) for more information

Under Windows, the base address and IRQ information is defined within Windows, so it is not necessary to specify a value.

This routine returns TRUE if the driver successfully opens the RH-02.

rh02_close_unit

```
void rh02_close_unit (unsigned short int port);
```

This routine disconnects the driver from the specified serial port.

If you successfully open any serial ports, you **MUST** call `rh02_close_unit` for each port before you exit from your program. If you do not, your computer may misbehave until you next reboot it.

rh02_get_cycle

```
short int rh02_get_cycle
    (long * cycle,
    short int port);
```

This routine returns the number of complete cycles of readings taken from a particular RH-02.

When you call `rh02_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 `rh02_get_value` only when a cycle has completed.

Note: each RH-02 is polled independently, so the cycle numbers for multiple RH-02s may not keep in step.

rh02_get_value

```
short int rh02_get_value (  
    long *      value,  
    short int   port,  
    short int   channel,  
    short int   filtered);
```

Once you open the driver, the driver constantly takes readings from the RH-02. When you call this routine, it immediately sets value to the most recent reading for the specified channel. The temperature is in hundredths of a degree Celsius, and the humidity in hundredths of a percent.

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 temperature, 2 for humidity.

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.

rh02_get_version

```
short int rh02_get_version (  
    short int * version,  
    short int   port);
```

This routine sets `version` to version number of the specified RH-02.

The upper byte of the version is always 8 for a RH-02: the lower byte is the two hex digits of the version and release. It provides a useful check that the link to the RH-02 is working correctly.

DOS Driver

The DOS driver is supplied in two object files, `rh02drv.obj` and `commdrv.obj`. It can be used in both `C` and `Pascal` programs.

Windows 16-bit Driver

The windows 16-bit driver is the file `RH0216.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`.

It has been tested under Windows 3.11 and under Windows 95 with the following 16-bit applications:

- Borland C 4.5
- Delphi 1
- Visual Basic 3
- Excel 5

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 RH0232.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/2000

Most applications running under Windows NT are 32-bit applications. The windows 32-bit driver is the file RH0232.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`.

Linux

A Linux driver is under development: please check the drivers section of the Pico Technology web site (<http://www.picotech.com/drivers.html>) for availability.

C

DOS

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

#include the header file `rh02.h` into your program

If you are using an IDE, include the file `rh02drv.obj` and `commdrv.obj` in you project.

If you are using a command-line compiler, include the file `rh02drv.obj` and `commdrv.obj` in you linkfile.

C / 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:

`rh02tes.c`

`rh02tes.rc`

either `rh0216.lib` (All 16-bit applications)

or `rh0232.lib` (Borland 32-bit applications)

or `rh02ms.lib` (Microsoft Visual C 32-bit applications)

The following files must be in the same directory:

`rh02tes.rch`

`rh02w.h`

either `rh0216.dll` (All 16-bit applications)

or `rh0232.dll` (All 32-bit applicaitons)

C++

C++ programs can access all versions of the driver. If `adc11.h` or `adc11w.h` are 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 `rh02pas.pas` can be compiled either as a stand-alone program `{ $DEFINE MAIN }` or as a unit which can be linked short into other programs `{ $UNDEF MAIN }`.

`rh02pas.pas` includes the driver using the `{ $L rh02drv.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.

Delphi

The WIN sub-directory contains a simple program `rh02.dpr` which opens the drivers and reads temperatures from the three channels. You will need the following files to build a complete program.

- `rh02fm.dfm`
- `rh02fm.pas`
- `rh02.inc`

The file `RH02.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 use PicoLog for Windows.

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.

Excel 5

The example `RH0216.XLS` reads in 20 values of the channel 1 temperature, one per second, and assigns them to cells A1..A20.

Excel 7

The example `RH0232.XLS` reads in 20 values of the channel 1 temperature, one per second, and assigns them to cells A1..A20.

Visual Basic

Version 3 (16 bits)

The DRIVERS\WIN16 sub-directory contains a simple Visual Basic program, `rh0216.mak`.

```
rh0216.MAK
rh0216.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:

rh0232.VBP
rh0232.BAS
rh0232.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 rh02.llb library in the DRIVERS\WIN32 sub-directory shows how to access this routine.

To use this routine, copy rh02.llb and rh0232.dll to your labview user.lib directory.

You the rh02 sub-vi will now appear in the user libraries box. You can load the rh02_example sub-vi which demonstrate how to use it. The sub-vi accepts the port (1 for COM1) and, optionally, whether to filter the data. It returns the temperature in degrees Celsius and the relative humidity in percent.

HP-Vee

The example program rh02.vee is in the drivers\win32 directory. It was tested using HP-Vee version 5 under Windows 95.

The example shows how to collect readings continuously from the rh02.

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 RH02 to the PC in some unusual way (for example, via a radio modem).

The RH-02 uses the following RS232 data lines (pin connections as on RH-02)

Pin	Name	Usage
3	TX	Data from the PC to the RH-02
2	RX	Data from the RH-02 to the PC
7	RTS	Held at a positive voltage (>7V) to power the RH-02
5	GND	0V line
4	DTR	Held at a negative voltage (<-7V) to power the RH-02

The driver powers up the RH-02 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 RH-02.

Protocol

The RH-02 operates at 2400 baud with 1 stop bit and no parity.

The driver controls the RH-02 using the following sequence

1. Switch RTS on and DTR off to provide power.
2. Wait for more than 1 second for the RH02 to settle
3. Send an single control byte to the RH02
4. Wait for a response from the RH02

Steps 3 and 4 are repeated for each measurement.

The RH02 signals the end of conversion by sending a block of bytes. No data should be sent to the RH02 during the conversion, as it may be lost or corrupted.

The RH02 accepts the following commands:

Command	Function	Response length	Response
0x01		2	RH02 Version
0x02		32	RH02_INFO (see below)
0x04		2	Temperature reading
0x05		2	Humidity reading

The RH02 version response is made up of a 2 (for RH02) followed by the version number (currently 1).

The RH02_INFO block contains the following information:

```
typedef struct
{
    unsigned short checksum;
```

```
unsigned char version;
unsigned char spare1;
unsigned short ref_low;
unsigned short ref_high;
char batch [6];
unsigned short unit;
char cal_date [8];
char spare2 [8];
} RH02_INFO;
```

The Temperature reading is converted to a temperature using the following table:

```
static unsigned short humtemp_table [] =
{
/* 0000 */ 14522U,
/* 0001 */ 15142U,
/* 0002 */ 15776U,
/* 0003 */ 16425U,
/* 0004 */ 17087U,
/* 0005 */ 17754U,
/* 0006 */ 18441U,
/* 0007 */ 19139U,
/* 0008 */ 19849U,
/* 0009 */ 20569U,
/* 0010 */ 21290U,
/* 0011 */ 22028U,
/* 0012 */ 22774U,
/* 0013 */ 23527U,
/* 0014 */ 24287U,
/* 0015 */ 25042U,
/* 0016 */ 25811U,
/* 0017 */ 26584U,
/* 0018 */ 27360U,
/* 0019 */ 28138U,
/* 0020 */ 28905U,
/* 0021 */ 29684U,
/* 0022 */ 30461U,
/* 0023 */ 31236U,
/* 0024 */ 32009U,
/* 0025 */ 32768U,
/* 0026 */ 33532U,
/* 0027 */ 34292U,
/* 0028 */ 35046U,
/* 0029 */ 35794U,
/* 0030 */ 36523U,
/* 0031 */ 37255U,
/* 0032 */ 37979U,
/* 0033 */ 38693U,
/* 0034 */ 39398U,
/* 0035 */ 40082U,
/* 0036 */ 40766U,
/* 0037 */ 41439U,
/* 0038 */ 42101U,
/* 0039 */ 42751U,
/* 0040 */ 43379U,
/* 0041 */ 44005U,
/* 0042 */ 44618U,
/* 0043 */ 45219U,
/* 0044 */ 45808U,
/* 0045 */ 46373U,
/* 0046 */ 46936U,
/* 0047 */ 47485U,
/* 0048 */ 48022U,
/* 0049 */ 48545U,
/* 0050 */ 49047U
};
```

The humidity reading is converted to a relative humidity using the following procedure:

```
humidity = (value - rh02_info.ref_low) * 100 / (rh02_info.ref_high - rh02_info.ref_low);  
temp_correction = (temp - 25) * 0.00216;  
humidity = humidity / (1 - temp_correction);
```

Modem operation

The RH-02 is normally connected directly to the computer, but it is also possible to access the RH-02 via a modem using the Windows driver.

It is necessary to provide power to the RH-02, either by instructing the modem to provide power or by connecting a power supply directly to the RH02. 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 RH02. 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.

```
[RH02]  
Turnround=300
```