# Technical Note

# Communications Protocol HygroLog NT

**Contents**

## Foreword

This technical note was prepared for customers who wish to use the HygroLog NT data logger on a network (consisting of one or more instruments) independently of the ROTRONIC HW4 software. This note is provided as information only and ROTRONIC shall not be liable for any direct or indirect damage or loss resulting from making use of the information provided here. In general, ROTRONIC can only provide limited technical support to customers who intend to use instruments on a network that is not based on the HW4 software.

## COM Port Settings

Baud rate: 57600
Data bits: 8
Stop bits: 1
Parity: none
Flow Control: none

## General Information

1) Network address: this is the address used by the ROTRONIC HW4 software to communicate with the different instruments on a network. This address should not to be confused with the IP address of any Device Server to which the instrument may be connected on an Ethernet network. When using HW4, each instrument should have a unique network address. Similarly, when several instruments are connected to a Device Server, each instrument within the multi-drop should have a unique network address. In principle, single instruments that are individually connected to a Device Server can share the same network address as long as HW4 is not being used.

2) In a dedicated HW4 network, the Master is automatically the instrument that is directly connected to the PC (RS-232 or USB). This is also defined as the first instrument of the network (and possibly the only instrument). RS-485 is used to connect the second instrument to the first, the third instrument to the second, etc. (multi-drop). In an open network (Ethernet), the instrument that is directly connected to a Device Server (RS-232) is the Master for that particular multi-drop.

3) During communication, the byte groups that constitute a command or a response are sent in the order indicated in this note. For example, when a Header is sent or received, the single byte corresponding to ESC is first and the single byte corresponding to CRC8 is last.

The following describes in which order the bytes within a group, and the bits within a byte, are sent or received:

a) Within a byte, bits are numbered from 0 to 7, counting from the right to the left. Bit 0 is the LSB and bit 7 is the MSB. The LSB (bit 0) is always sent or received first
b) When two or more bytes are part of a group, the low byte is always sent or received first. Examples of byte groups: bytes used to declare the data length in the header (2-byte group), bytes used to pass data in commands 4, 15, 20 and 30, bytes used for the measurement data from the HygroLog NT, bytes used for CRC16 (2-byte group).

4) Both the instrument network address and the type of command use a single byte. The most significant bit of these bytes is reserved for a flag. This leaves 7 bits that can be used for different addresses and commands. Both the maximum number of addresses and the maximum number of command types are limited to the decimal range of 0 to 127 (binary 0000000 to 1111111).

Bit 7 of the address byte is used as a forward flag. To address an instrument that is part of a multi-drop and that is not the Master, bit 7 of the address should be set to 1 (forward flag). This tells the Master to pass the command further.

Each response repeats the type of command that is being answered. Bit 7 of the command type byte is used as a response flag. This bit is set to 0 for a command and to 1 for the response. This is used to differentiate between a command and a response.

5) Network address 255 (binary 11111111) is reserved. This address is used to send commands directly to the Master (first instrument connected to the COM port of the PC or to the serial port of a Device Server). Address 255 should not be used in the situation of an open Ethernet network with several Device Servers.

## Communications Protocol Structure

The general structure of the HygroLog NT communications protocol is as follows:

**Header + Header Checksum + Data + Data Checksum**

Header : consists primarily of the instrument network address, command and the length of any data sent to the instrument or returned by the instrument.
Data : data sent to the instrument or returned by the instrument

Checksums are used to verify integrity of the transmission.

Commands sent to the HygroLog NT as well as the response string always include a header and a header checksum. Data and the data checksum may or may not be present.

**Header (length: 7 bytes)**

ESC + Instrument Type ID + Network Address + Command + Command Parameter + Data length

ESC: 1 byte, first character of the header (ASCII 27 or 0x1B)

Type ID: 1 byte, instrument type ID (for the HygroLogNT: „L“ or ASCII 76 or 0x4C)

Address: 1 byte, instrument network address 0..127.

To address an instrument that is part of a multi-drop and that is not the Master, bit 7 of the address byte should be set to 1 (forward flag).
(0 = Master, 1 = Slave)

Address 255 causes the master answer regardless of the actual instrument network address. Use of address 255 should be reserved for communicating with a single instrument of unknown network address.

Command: 1 byte, command (0..127)

Bit 7 of the command byte is used as a response flag
(0 = command, 1 = response)

Parameter: 1 byte, command code. Most commands do not use any parameter and in that case byte 0x00 is used.

Data length: 2 bytes, low byte first. Number of data bytes sent to or returned by the instrument When no data is being sent, two 0x00 bytes are used.

**Header Checksum**: CRC8 (length: 1 byte)

CRC8 is used to verify the integrity of the header transmission.

**Data** (variable length, low byte first)

Data should be formatted as a Byte array. For the transmission of data from the PC to the instrument, the size of a data packet is limited to 246 bytes. Larger amounts of data should be divided into packets prior to being sent. For the transmission of data from the instrument to the PC, the size of a data packet is limited to 65535 bytes.

**Data Checksum**: CRC16 (length: 2 bytes, low byte first)

This checksum is used only when data is being sent. The data checksum consists of two 8-bit bytes and is computed based on the data that is being sent. During communication, the LSB is sent first and the MSB last. CRC16 is used to verify the integrity of the data transmission.

## *Examples*

Note: in the following examples, the value of the different checksums has left not been computed.

<u>Example 1:</u>

Send command “1 “and the number 321 (data) to a HygroLog NT with an unknown network address. Reserved address 255 is used to talk to the instrument:

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x1B | 0x4C | 0xFF | 0x01 | 0x00 | 0x03 | 0x00 | 0x?? | 0x01 | 0x02 | 0x03 | 0x?? | 0x?? |
| ESC | „L“ | Addr | Cmd | Para | Length | | CRC8 | 1 | 2 | 3 | CRC16 | |

Instrument response (the instrument network address is 0) :

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0x1B | 0x4C | 0x00 | 0x81 | 0x00 | 0x00 | 0x00 | 0x?? |
| ESC | „L“ | Addr | Cmd | Para | Length | | CRC8 |

Notes:

In the answer, the command byte is repeated after adding decimal 128 to the decimal value of the command (answer flag set to 1). Decimal 128 + 1 = 0x81

In both the command and the answer, the byte-groups Length, Data and CRC16 are sent / received starting with the low byte.

<u>Example 2:</u>

Send command “2” to read data from the instrument

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0x1B | 0x4C | 0xFF | 0x02 | 0x00 | 0x00 | 0x00 | 0x?? |
| ESC | „L“ | Addr | Cmd | Para | Length | | CRC8 |

Instrument Response (the data is assumed to be the number 321):

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x1B | 0x4C | 0x00 | 0x82 | 0x00 | 0x03 | 0x00 | 0x?? | 0x01 | 0x02 | 0x03 | 0x?? | 0x?? |
| ESC | „L“ | Addr | Cmd | Para | Length | | CRC8 | 1 | 2 | 3 | CRC16 | |

Example 3:

Send command “1” and the number 321 (data) to a HygroLog NT with address 7 (not the Master) in a multi-drop.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x1B | 0x4C | 0x87 | 0x01 | 0x00 | 0x03 | 0x00 | 0x?? | 0x01 | 0x02 | 0x03 | 0x?? | 0x?? |
| ESC | „L“ | Addr | Com | Para | Length | | CRC8 | 1 | 2 | 3 | CRC16 | |

Note: address 7 corresponds to binary 00000111. Since the instrument is not the master in the multi-drop, bit 7 of the address (forward flag) is set to 1 and the address is sent as 10000111 which is the same as decimal 135 (128 + 7) or 0x87.

Instrument Response:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0x1B | 0x4C | 0x07 | 0x81 | 0x00 | 0x00 | 0x00 | 0x?? |
| ESC | „L“ | Addr | Cmd | Para. | Length | | CRC8 |

Notes:

- The instrument network address is always included in the instrument response, even when address 255 was used to talk to the instrument.

- The instrument response always repeats the command that was sent to the instrument. In order to differentiate between a request and a response, bit 7of the command is set to 1 in the response.

- When no data is being sent (data length zero), the CRC16 checksum is omitted.

# Commands

## *Command 2: request current data (standard inputs)*

This command is used to read the current measured and calculated values for all standard probe inputs of the HygroLog NT. The response includes all of the information that is required to process the data: probe number, calculation type, engineering units, status, serial number and name of the instrument, etc.

**Command format**:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ESC | „L“ | Addr | 2 | 0x00 | 0x00 | 0x00 | CRC8 |

**Response format:**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ESC | „L“ | Addr | 2+128 | 0x00 | 172 (2 bytes) | CRC8 | Data (see Table) | CRC16 |

In the response, a total of 172 bytes are used to send the data, using the following structure:

| | | |
|---|---|---|
| 30 | Byte | Probe 1 |
| 30 | Byte | Probe 2 |
| 30 | Byte | Probe 3 |
| 30 | Byte | Reserved |
| 1 | Byte | Temperature unit (see Units Table) |
| 1 | Byte | Humidity unit (% or Aw, see Units Table) |
| 2 | String | Humidity descriptor (2 characters following the % character) |
| 1 | Byte | Pressure unit (see Units Table) |
| 1 | Byte | Enthalpy unit (see Units Table) |
| 1 | Byte | Density per volume unit (see Units Table) |
| 1 | Byte | Density per weight unit (see Units Table) |
| 10 | String | Instrument serial number |
| 30 | String | Instrument name (user text) |
| 3 | Byte | Reserved |
| 1 | Byte | Status of the optional inputs (see Option Status Table) |

**Structure for Probe 1 to Probe 3 (digital probes):**

| | | |
|---|---|---|
| 12 | String | Probe description |
| 4 | Single | Humidity (measured value) see data format below |
| 4 | Single | Temperature (measured value) see data format below |
| 4 | Single | Calculated parameter see data format below |
| 1 | Byte | Humidity status (see Value Status Table) |
| 1 | Byte | Temperature status (see Value Status Table) |
| 1 | Byte | Calculation status (see Value Status Table) |
| 1 | Byte | Calculation type (see Calculated Parameter) |
| 1 | Byte | Reserved |
| 1 | Byte | Probe type (see Probe Type Table) |

**Important:**

Regardless of the configuration of the instrument, both the measured and calculated values are sent in metric units. Relative humidity is always sent in %RH, even if the instrument displays water activity. Converting back the values to the instrument unit system must be done by the PC.

**Data format for humidity, temperature and the calculated parameter:**

Four bytes are used to send the data for each of these parameters. The low byte is sent first. Putting the four bytes back together in the proper order gives a 32-bit variable of the Single type that is formatted according to IEEE standard 754 as a floating-point binary number (scientific binary).

Detailed information on the IEEE standard 754 can be found at the following web sites:

http://research.microsoft.com/~hollasch/cgindex/coding/ieeefloat.html

http://www.randelshofer.ch/fhw/gri/float.html

## *Command 7: request current data (optional inputs from docking station)*

This command is used to read the current measured and calculated values for all the optional inputs of the HygroLog NT docking station. The response includes all of the information that is required to process the data: probe number, calculation type, engineering units, status, serial number and name of the instrument, etc.

**Command format:**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ESC | „L“ | Addr | 7 | 0x00 | 0x00 | 0x00 | CRC8 |

**Response format:**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ESC | „L“ | Addr | 7+128 | 0x00 | 172 (2 bytes) | CRC8 | Data (see Table) | CRC16 |

A total of 152 bytes are used to send the data, using the following structure:

| | | |
|---|---|---|
| 30 | Byte | Probe 4 |
| 30 | Byte | Probe 5 |
| 30 | Byte | Probe 6 |
| 30 | Byte | Probe 7 |
| 14 | Byte | Digital Input 1 |
| 14 | Byte | Digital Input 2 |
| 4 | Byte | Reserved |

**Structure for digital probes (probe type 0 and 7)**

| | | |
|---|---|---|
| 12 | String | Probe description |
| 4 | Single | Humidity (measured value) see data format below |
| 4 | Single | Temperature (measured value) see data format below |
| 4 | Single | Calculated parameter see data format below |
| 1 | Byte | Humidity status (see Value Status Table) |
| 1 | Byte | Temperature status (see Value Status Table) |
| 1 | Byte | Calculation status (see Value Status Table) |
| 1 | Byte | Calculation type (see Calculated Parameter Table) |
| 1 | Byte | Reserved |
| 1 | Byte | Probe type (see Probe Type Table) |

**Structure for analog probes (probe type 2 and 3)**

| | | |
|---|---|---|
| 12 | String | Probe description |
| 4 | Single | Measured value |
| 4 | String | Unit |
| 4 | Byte | Reserved |
| 1 | Byte | Measured value status (see Value Status Table) |
| 4 | Byte | Reserved |
| 1 | Byte | Probe type (see Probe Type Table) |

**Structure for digital inputs**

| | | |
|---|---|---|
| 12 | String | Input description |
| 1 | Byte | Logical state (0 or 1) |
| 1 | Byte | Reserved |

**Important:**

Regardless of the configuration of the instrument, both the measured and calculated values are sent in metric units. Relative humidity is always sent in %RH, even if the instrument displays water activity. Converting back the values to the instrument unit system must be done by the PC.

**Data format for humidity, temperature and the calculated parameter:**

Four bytes are used to send the data for each of these parameters. The low byte is sent first. Putting the four bytes back together in the proper order gives a 32-bit variable of the Single type that is formatted according to IEEE standard 754 as a floating-point binary number (scientific binary).

Detailed information on the IEEE standard 754 can be found at the following web sites:

http://research.microsoft.com/~hollasch/cgindex/coding/ieeefloat.html

http://www.randelshofer.ch/fhw/gri/float.html

## *Command 4: write the settings for logging data*

This command writes to the logger the instructions necessary for recording data. This command uses a total of up to 640 bytes. In order not to exceed the 246 bytes limit for data packets sent to the instrument, command 4 may have to be sent up to three times (or in 3 parts).

**Command format:**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ESC | „L“ | Addr | 4 | 0x00 | 240 (2 bytes) | CRC8 | data - part 1 | CRC16 |
| ESC | „L“ | Addr | 4 | 0x01 | 240 (2 bytes) | CRC8 | data - part 2 | CRC16 |
| ESC | „L“ | Addr | 4 | 0x02 | 160 (2 bytes) | CRC8 | data - part 3 | CRC16 |

Note the use of a command parameter (byte 5) to differentiate the data parts

**Response format:**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ESC | „L“ | Addr | 4+128 | 0x00 | 0x00 | 0x00 | CRC8 |

- The instrument answers after receiving each command. When sending more than one data - part, wait for the instrument answer before sending the next data - part. While the next data - part (if any) should be sent as soon as possible after the instrument answers, timing is not critical.

- Command 4 has an effect only with inputs that are not currently in the process of recording data. If necessary, use Command 6 prior to sending Command 4.

- Each data - part sets the date and time of the instrument. HW4 automatically uses the current date and time of the PC, but here this has to be specified.

- Data - part 2 and data - part 3 are used only when probe inputs 3 to 5 and inputs 6 and 7 need to be configured.

- No data will be logged when the start time is equal to or exceeds the stop time

Please note that the numbering of probes is offset by 1 compared to the numbering of probe inputs. Probe 1 is the probe connected to probe input 0, etc.

**Structure of data - part 1:**

| | | |
|---|---|---|
| 80 | Byte | probe 1 (input 0 - all HygroLog NT models) |
| 80 | Byte | probe 2 (input 1 - HygroLog NT3 only) |
| 80 | Byte | probe 3 (input 2 - HygroLog NT3 only) |

**Structure of data - part 2:**

| | | |
|---|---|---|
| 80 | Byte | probe 4 (input 3 - docking station) |
| 80 | Byte | probe 5 (input 4 - docking station) |
| 80 | Byte | probe 6 (input 5 - docking station) |

**Structure of data - part 3:**

| | | |
|---|---|---|
| 80 | Byte | probe 7 (input 6 - docking station) |
| 80 | Byte | digital input (input 7 - docking station) |

**Structure for inputs 0 to 6 (80 bytes):**

| | | |
|---|---|---|
| 4 | Long | Start time (see Time format) |
| 4 | Long | Stop time (see Time format) |
| 2 | Word | Log interval (in steps of 5 sec) |
| 1 | Byte | Values to be recorded (bit 7 is the MSB, bit 0 is the LSB):<br>bit 0 = Humidity (set to one = record, set to zero = do not record)<br>bit 1 = Temperature (set to one = record, set to zero = do not record)<br>bit 2 = calculated parameter (set to one = record, set to zero = do not record)<br><br>Example: 00000111 (decimal 7) = record all 3 values |
| 1 | Byte | Create a new log file:<br>0 = each 200,000 log intervals<br>1 = each hour<br>2 = each day<br>3 = each week<br>4 = each month |
| 4 | Long | Create the first log file on (see Time format) |
| 4 | Byte | Reserved |
| 32 | String | Description (user text) |
| 15 | Byte | Reserved for HW4 Software |
| 9 | Byte | Reserved for HW4 Software |
| 4 | Long | Current date and time (see Time format) |

**Structure for input 7 - digital signals (80 bytes):**

| | | |
|---|---|---|
| 4 | Long | Start time (see Time format) |
| 4 | Long | Stop time (see Time format) |
| 2 | Word | Log interval (in steps of 5 sec) |
| 1 | Byte | Values to be recorded (bit 7 is the MSB, bit 0 is the LSB):<br>bit 0 = Digital input 1 (set to one = record, set to zero = do not record)<br>bit 1 = Digital input 2 (set to one = record, set to zero = do not record)<br>bit 2 = not used (set bit to zero) |
| 1 | Byte | Create a new log file:<br>0 = each 200,000 log intervals<br>1 = each hour<br>2 = each day<br>3 = each week<br>4 = each month |
| 4 | Long | Create the first log file on (see Time format) |
| 4 | Byte | Reserved |
| 32 | String | Description (user text) |
| 15 | Byte | Reserved for HW4 Software |
| 9 | Byte | Reserved for HW4 Software |
| 4 | Long | Current date and time (see Time format) |

Notes:

- This structure should be used for each input, even for those inputs where no probe is connected.
- To prevent data recording for any specific input, use a Start time that is in the past relative to the current date and time.

### *Command 6: stop logging data*

This command is used to force an end to the recording of data by any single input.

**Command format:**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ESC | „L“ | Addr | 6 | input number | 0x00 | 0x00 | CRC8 |

**Response format:**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ESC | „L“ | Addr | 6+128 | 0x00 | 0x00 | 0x00 | CRC |

Please note that the numbering of probes is offset by 1 compared to the numbering of probe inputs. Probe 1 is the probe connected to probe input 0, etc.

### *Command 15: set instrument date and time*

This command is used to set the date and time of the instrument.

**Command format:**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ESC | „L“ | Addr | 15 | 0x00 | 28 (2 bytes) | CRC8 | Data (28Bytes) | CRC16 |

**Response format:**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ESC | „L“ | Addr | 15+128 | 0x00 | 0x00 | 0x00 | CRC8 |

**Data structure:**

| | | |
|---|---|---|
| 15 | Byte | Reserved for HW4 Software |
| 9 | Byte | Reserved for HW4 Software |
| 4 | Long | Date and Time (see Time format) |

### *Command 30: delete a log file*

This command is used to delete a log file from the instrument flash memory card.

**Command format:**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ESC | „L“ | Addr | 30 | 0x00 | 11 (2 bytes) | CRC8 | file name | CRC16 |

File name: 11 characters (bytes): 8 for the file name, 3 for the extension without leading dot

**Response format:**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ESC | „L“ | Addr | 30+128 | 0x00 | 0x00 | 0x00 | CRC8 |

Note: deleting a large file may take a few seconds. The response is sent after the file deletion is completed.

## *Command 22: read the root directory of the flash memory card*

This command is used to read the root directory of the flash memory card present in the instrument.

**Command format:**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ESC | „L“ | Addr | 22 | 0x00 | 0x00 | 0x00 | CRC8 |

**Response format:**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ESC | „L“ | Addr | 22+128 | 0x00 | 32 (2 bytes) | CRC8 | Directory entry | CRC16 |

The instrument sends a separate response for each individual entry in the directory.

**Structure of directory entry** (see also FAT specification from Microsoft):

| | | |
|---|---|---|
| 11 | String | File name and extension: 11 ASCII characters (w/o separating dot) |
| 1 | Byte | Attribute:<br>0x01 = Read Only<br>0x02 = Hidden<br>0x04 = System<br>0x08 = Volume_ID<br>0x10 = Directory<br>0x20 = Archive<br>(0x0F = Long  Name) |
| 1 | Byte | Reserved |
| 1 | Byte | Creation time in hundredth of a second (not used) |
| 2 | Word | Creation time |
| 2 | Word | Creation date |
| 2 | Word | Date last accessed |
| 2 | Word | High Word Cluster Number (FAT32 only) |
| 2 | Word | Time of last modification |
| 2 | Word | Date of last modification |
| 2 | Word | Cluster Number (Position of the data in the flash memory card) |
| 4 | Long | File size |

**Date and time format** (from the FAT specification):

<u>Date Format:</u>

A FAT directory entry date stamp is a 16-bit field that is basically a date relative to the MS-DOS epoch of 01/01/1980. Here is the format (bit 0 is the LSB of the 16-bit word, bit 15 is the MSB of the 16-bit word):

Bits 0–4: day of month, valid value range 1-31 inclusive.
Bits 5–8: month of year, 1 = January, valid value range 1–12 inclusive.
Bits 9–15: count of years from 1980, valid value range 0–127 inclusive (1980–2107).

Time Format:

A FAT directory entry time stamp is a 16-bit field that has a granularity of 2 seconds.

Here is the format (bit 0 is the LSB of the 16-bit word, bit 15 is the MSB of the 16-bit word).
Bits 0–4: 2-second count, valid value range 0–29 inclusive (0 – 58 seconds):

Bits 5–10: Minutes, valid value range 0–59 inclusive.
Bits 11–15: Hours, valid value range 0–23 inclusive.

The valid time range is from Midnight 00:00:00 to 23:59:58.

## *Command 20: download a log file*

This command is used to download a log file from the instrument flash memory card to the PC.

**Command format:**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ESC | „L“ | Addr | 20 | 0x00 | 15 (2 bytes) | CRC8 | Parameter | CRC16 |

**Structure of Parameter:**

| | | |
|---|---|---|
| 11 | String | File name and extension: 11 ASCII characters (w/o separating dot) |
| 2 | Long | Offset |
| 2 | Long | Number of sectors (max. 100) |

See example below regarding the values to use for Offset and Number of sectors

**Response format:**

The requested data are transmitted in one block as they were recorded on the flash memory card. There is no header or CRC. The instrument does not send a checksum after sending the data. Log files recorded in text format are therefore not protected against transmission errors. Log files recorded in the proprietary HW4 binary format are protected against transmission errors because they include a checksum within the data itself. Note that HW4 is required to read this file format on the PC.

Important:

A limit of 100 sectors can be downloaded in one operation (1 sector = 512 bytes). Attempting to read a larger number of sectors will cause the instrument Watchdog to interrupt the transmission. In the case of a log file with more than 100 sectors, command 20 should be repeated as many times as necessary, downloading a maximum of 100 sectors each time.

<u>Example:</u> log file size 125,398 bytes

125398 / 512 = 244.91, meaning 244 sectors (0 to 243) are full and one sector is used partially
125398 mod 512 = 470, meaning the last sector (244) has 470 bytes of data

Step 1: read sectors 0 to 99

| | | |
|---|---|---|
| 11 | String | File name and extension: 11 ASCII characters (w/o separating dot) |
| 2 | Long | 0 |
| 2 | Long | 100 |

Step 2: read sectors 100 to 199

| | | |
|---|---|---|
| 11 | String | File name and extension: 11 ASCII characters (w/o separating dot) |
| 2 | Long | 100 |
| 2 | Long | 100 |

Step 3: read sectors 200 to 244

| | | |
|---|---|---|
| 11 | String | File name and extension: 11 ASCII characters (w/o separating dot) |
| 2 | Long | 200 |
| 2 | Long | 45 |

Note: in this example, the last sector (244) has only 470 bytes of data.

# Tables

## *Calculated Parameter*

| | |
|---|---|
| 0 | No calculation |
| 1 | Dew point (Dp) |
| 2 | Frost point (Fp) |
| 3 | Wet bulb temperature (Tw) |
| 4 | Enthalpy (H) |
| 5 | Vapor concentration (Dv) |
| 6 | Specific humidity (Q) |
| 7 | Mixing ratio by weight (R) |
| 8 | Vapor concentration at saturation (Dvs) |
| 9 | Vapor partial pressure (E) |
| 10 | Vapor saturation pressure (Ew) |

## *Units*

| | |
|---|---|
| 0 | No unit |
| 1 | Relative humidity % |
| 2 | Water activity Aw |
| 3 | Temperature °C |
| 4 | Temperature °F |
| 5 | Pressure hPa |
| 6 | Pressure In.Hg |
| 7 | Pressure PSI |
| 8 | Enthalpy kJ/kg |
| 9 | Enthalpy BTU/lb |
| 10 | Density per volume unit g/m3 |
| 11 | Density per volume unit gr/cu ft |
| 12 | Density per weight unit g/kg |
| 13 | Density per weight unit gr/lb |

## *Value Status*

| | |
|---|---|
| 0 | Value is valid |
| 1 | N/A (for calculated parameter) |
| 2 | Value not visible |
| 3 | Value is not valid (no probe) |
| Status Bits: | |
| 16 | Trend UP |
| 32 | Trend DOWN |
| 64 | Alarm |
| 128 | Logging in progress |

Note: 16+32 = Trend is stable

### *Probe Type*

| | |
|---|---|
| 0 | HygroClip |
| 1 | Reserved |
| 2 | Pressure |
| 3 | Third party probe |
| 4 | Reserved |
| 5 | Reserved |
| 6 | No probe is connected |
| 7 | Digital SHT probe |

### *Option Status*

| | |
|---|---|
| 0 | No docking station with inputs |
| 1 | Docking station RS232/ HygroClip |
| 2 | Docking station USB/ HygroClip |
| 3 | Docking station RS232/ PT100 |
| 4 | Docking station USB/ PT100 |

# Appendix

### *Structure of the log file names*

The HygroLog NT automatically generates log file names as explained below:

4 characters = last 4 digits of the instrument serial number
1 character = probe number (1 to 3 for probes directly attached to the logger, 4 to 7 for probes attached to the optional docking station and 8 for the docking station digital input)
3 characters = 3-digit running number 000…999.

Each time that the logger creates a new file for whatever reason, the running number is increased by 1. After reaching the value 999, the running number is reset to 000.

Note: The logger uses a separate running number for each input

### *Log file types*

Extension XLS

Files with extension XLS are text files that can also be opened with Microsoft Excel. These files are not in Excel format. If Excel is used to open and then save such a file, the file can no longer be opened either with a text editor (such as Notepad) or with the ROTRONIC HW4 software.

Extension LOG

Files with the LOG extension are created using the proprietary ROTRONIC HW4 binary format. This format uses a check sum to detect error for each data entry. Files with the extension LOG can only be read with the ROTRONIC HW4 software.

### *Visual Basic Code*

## Date and time format

Intervals of 5 seconds are used counting from 1.1.2000 00:00:00 (January 1, 2000 - 00:00:00).

Code for Visual Basic.NET:

```
Dim myDate as Date
Dim LongValue as Long
LongValue = DateDiff(DateInterval.Second, #1/1/2000#, myDate) \ 5
```

# Computing the CRC8 checksum (VBA code)

```
Private Sub CRC8(ByRef Data() As Variant)

    '--------------------------------------------------------------------------
    ' Function     :    (Private) CRC8
    ' Description  :    computes a 1 byte checksum using bytes 1 to 7 of the
    '                   Header 8-byte array (Protocol Header)
    '
    ' Parameter    :    Data ()As Byte  | each byte group is assumed to be High byte first, Low byte last
    ' Return Value :  CRC8 as Byte   | CRC value
    '--------------------------------------------------------------------------
    Dim CRC As Byte
    Dim Length As Byte
    Dim Index As Byte
    Dim CRCtable1() As Variant

    CRC = 0
    Length = 1


    CRCtable1 = Array( _
    &H0, &H7, &HE, &H9, &H1C, &H1B, &H12, &H15, &H38, &H3F, &H36, &H31, &H24, &H23, &H2A, &H2D, &H70,
&H77, &H7E, &H79, &H6C, &H6B, &H62, &H65, &H48, &H4F, &H46, &H41, &H54, &H53, &H5A, &H5D, _
    &HE0, &HE7, &HEE, &HE9, &HFC, &HFB, &HF2, &HF5, &HD8, &HDF, &HD6, &HD1, &HC4, &HC3, &HCA, &HCD,
&H90, &H97, &H9E, &H99, &H8C, &H8B, &H82, &H85, &HA8, &HAF, &HA6, &HA1, &HB4, &HB3, &HBA, &HBD, _
    &HC7, &HC0, &HC9, &HCE, &HDB, &HDC, &HD5, &HD2, &HFF, &HF8, &HF1, &HF6, &HE3, &HE4, &HED, &HEA,
&HB7, &HB0, &HB9, &HBE, &HAB, &HAC, &HA5, &HA2, &H8F, &H88, &H81, &H86, &H93, &H94, &H9D, &H9A, _
    &H27, &H20, &H29, &H2E, &H3B, &H3C, &H35, &H32, &H1F, &H18, &H11, &H16, &H3, &H4, &HD, &HA, &H57,
&H50, &H59, &H5E, &H4B, &H4C, &H45, &H42, &H6F, &H68, &H61, &H66, &H73, &H74, &H7D, &H7A, _
    &H89, &H8E, &H87, &H80, &H95, &H92, &H9B, &H9C, &HB1, &HB6, &HBF, &HB8, &HAD, &HAA, &HA3, &HA4,
&HF9, &HFE, &HF7, &HF0, &HE5, &HE2, &HEB, &HEC, &HC1, &HC6, &HCF, &HC8, &HDD, &HDA, &HD3, &HD4, _
    &H69, &H6E, &H67, &H60, &H75, &H72, &H7B, &H7C, &H51, &H56, &H5F, &H58, &H4D, &H4A, &H43, &H44, &H19,
&H1E, &H17, &H10, &H5, &H2, &HB, &HC, &H21, &H26, &H2F, &H28, &H3D, &H3A, &H33, &H34, _
    &H4E, &H49, &H40, &H47, &H52, &H55, &H5C, &H5B, &H76, &H71, &H78, &H7F, &H6A, &H6D, &H64, &H63, &H3E,
&H39, &H30, &H37, &H22, &H25, &H2C, &H2B, &H6, &H1, &H8, &HF, &H1A, &H1D, &H14, &H13, _
    &HAE, &HA9, &HA0, &HA7, &HB2, &HB5, &HBC, &HBB, &H96, &H91, &H98, &H9F, &H8A, &H8D, &H84, &H83,
&HDE, &HD9, &HD0, &HD7, &HC2, &HC5, &HCC, &HCB, &HE6, &HE1, &HE8, &HEF, &HFA, &HFD, &HF4, &HF3 _
    )

    Do While Length < 7
        Index = CRC Xor Data(Length)
        CRC = CRCtable1(Index)
        Length = Length + 1
    Loop

    CHK8 = CRC

End Sub
```

## Computing the CRC16 checksum (VBA code)

```
Private Sub CRC16(ByRef Data() As Variant, ByVal Length As Integer)

    '--------------------------------------------------------------------------
    ' Function      : (Private) CRC16
    ' Description  : Computes the 16-bit CRC of a byte array
    '
    ' Parameter    : Data As Byte()        | each byte group is assumed to be High byte first, Low byte last
    '                      Length As Integer    | number of bytes
    ' Return Value : CRC16 As Integer  | CRC value High byte first, Low byte last
    '--------------------------------------------------------------------------

    Dim CRC As Currency
    Dim C1, C2 As Long
    Dim Count, Index As Integer
    Dim CRCtable2() As Variant

    CRC = &HFFFF&
    Count = 0
    C1 = 4294967295#

    CRCtable2 = Array( _
    &H0&, &H1021&, &H2042&, &H3063&, &H4084&, &H50A5&, &H60C6&, &H70E7&, &H8108&, &H9129&, &HA14A&,
&HB16B&, &HC18C&, &HD1AD&, &HE1CE&, &HF1EF&, &H1231&, &H210&, &H3273&, &H2252&, &H52B5&,
&H4294&, &H72F7&, &H62D6&, _
    &H9339&, &H8318&, &HB37B&, &HA35A&, &HD3BD&, &HC39C&, &HF3FF&, &HE3DE&, &H2462&, &H3443&,
&H420&, &H1401&, &H64E6&, &H74C7&, &H44A4&, &H5485&, &HA56A&, &HB54B&, &H8528&, &H9509&, &HE5EE&,
&HF5CF&, &HC5AC&, &HD58D&, _
    &H3653&, &H2672&, &H1611&, &H630&, &H76D7&, &H66F6&, &H5695&, &H46B4&, &HB75B&, &HA77A&,
&H9719&, &H8738&, &HF7DF&, &HE7FE&, &HD79D&, &HC7BC&, &H48C4&, &H58E5&, &H6886&, &H78A7&,
&H840&, &H1861&, &H2802&, &H3823&, _
    &HC9CC&, &HD9ED&, &HE98E&, &HF9AF&, &H8948&, &H9969&, &HA90A&, &HB92B&, &H5AF5&, &H4AD4&,
&H7AB7&, &H6A96&, &H1A71&, &HA50&, &H3A33&, &H2A12&, &HDBFD&, &HCBDC&, &HFBBF&, &HEB9E&,
&H9B79&, &H8B58&, &HBB3B&, &HAB1A&, _
    &H6CA6&, &H7C87&, &H4CE4&, &H5CC5&, &H2C22&, &H3C03&, &HC60&, &H1C41&, &HEDAE&, &HFD8F&,
&HCDEC&, &HDDCD&, &HAD2A&, &HBD0B&, &H8D68&, &H9D49&, &H7E97&, &H6EB6&, &H5ED5&, &H4EF4&,
&H3E13&, &H2E32&, &H1E51&, &HE70&, _
    &HFF9F&, &HEFBE&, &HDFDD&, &HCFFC&, &HBF1B&, &HAF3A&, &H9F59&, &H8F78&, &H9188&, &H81A9&,
&HB1CA&, &HA1EB&, &HD10C&, &HC12D&, &HF14E&, &HE16F&, &H1080&, &HA1&, &H30C2&, &H20E3&, &H5004&,
&H4025&, &H7046&, &H6067&, _
    &H83B9&, &H9398&, &HA3FB&, &HB3DA&, &HC33D&, &HD31C&, &HE37F&, &HF35E&, &H2B1&, &H1290&,
&H22F3&, &H32D2&, &H4235&, &H5214&, &H6277&, &H7256&, &HB5EA&, &HA5CB&, &H95A8&, &H8589&,
&HF56E&, &HE54F&, &HD52C&, &HC50D&, _
    &H34E2&, &H24C3&, &H14A0&, &H481&, &H7466&, &H6447&, &H5424&, &H4405&, &HA7DB&, &HB7FA&,
&H8799&, &H97B8&, &HE75F&, &HF77E&, &HC71D&, &HD73C&, &H26D3&, &H36F2&, &H691&, &H16B0&, &H6657&,
&H7676&, &H4615&, &H5634&, _
    &HD94C&, &HC96D&, &HF90E&, &HE92F&, &H99C8&, &H89E9&, &HB98A&, &HA9AB&, &H5844&, &H4865&,
&H7806&, &H6827&, &H18C0&, &H8E1&, &H3882&, &H28A3&, &HCB7D&, &HDB5C&, &HEB3F&, &HFB1E&,
&H8BF9&, &H9BD8&, &HABBB&, &HBB9A&, _
    &H4A75&, &H5A54&, &H6A37&, &H7A16&, &HAF1&, &H1AD0&, &H2AB3&, &H3A92&, &HFD2E&, &HED0F&,
&HDD6C&, &HCD4D&, &HBDAA&, &HAD8B&, &H9DE8&, &H8DC9&, &H7C26&, &H6C07&, &H5C64&, &H4C45&,
&H3CA2&, &H2C83&, &H1CE0&, &HCC1&, _
    &HEF1F&, &HFF3E&, &HCF5D&, &HDF7C&, &HAF9B&, &HBFBA&, &H8FD9&, &H9FF8&, &H6E17&, &H7E36&,
&H4E55&, &H5E74&, &H2E93&, &H3EB2&, &HED1&, &H1EF0& _
    )

    Do While Count < Length
        Index = ((CRC \ 256) Xor Data(Count)) And &HFF
        CRC = CRCtable2(Index) Xor ((CRC And &HFFFF&) * 256)
        Count = Count + 1
    Loop

    CHK16 = CRC And &HFFFF&
```

```
'Convert CHK16 into 2 bytes

    TCRC16_B1 = "&H" + Mid(Hex(CHK16), 1, 2)
    TCRC16_B2 = "&H" + Mid(Hex(CHK16), 3, 2)
    CRC16_B1 = Val(TCRC16_B1)    'High byte
    CRC16_B2 = Val(TCRC16_B2)    'Low byte

 End Sub
```

## Converting a floating-point binary number to a regular decimal number (VBA code)

```
Sub BFP_Format_to_Decimal()

'In the following examples, the bytes are assumed to be in the proper order (High byte first, Low byte last), not in the send / receive order

'Example 1: decimal -0.0123 (cannot be represented exactly in BFP format)

Byte_1 = "10111100"
Byte_2 = "01001001"
Byte_3 = "10000101"
Byte_4 = "11110000"  'this is the low byte

'Example 2: decimal 13.13 (cannot be represented exactly in BFP format)

'Byte_1 = "01000001"
'Byte_2 = "01010010"
'Byte_3 = "00010100"
'Byte_4 = "01111011"

'Example 3: NaN (not a number)

'Byte_1 = "01111111"
'Byte_2 = "11010010"
'Byte_3 = "00010100"
'Byte_4 = "01111011"

'Example 4: Zero

'Byte_1 = "00000000"
'Byte_2 = "00000000"
'Byte_3 = "00000000"
'Byte_4 = "00000000"

'Step1: extract the sign, exponent and significand from the 4 bytes

Sign_bit = Mid(Byte_1, 1, 1)                    'Sign bit
Expo = Mid(Byte_1, 2, 7) + Mid(Byte_2, 1, 1)     'Exponent
Sgnd = Mid(Byte_2, 2, 7) + Byte_3 + Byte_4       'Significand


'Step2: Handle exceptions

If Expo = "00000000" Then   'zero or denormalized (small) number
   If Val(Sgnd) = 0 Then
      Dec_Nbr = 0
      GoTo Terminate
   Else
      GoTo Terminate 'this case corresponds to a denormalized number
   End If
End If

If Expo = "11111111" And Val(Sgnd) = 0 Then
   GoTo Terminate 'This exception is reserved for infinities (+ or -)
End If

If Expo = "11111111" And Val(Sgnd) <> 0 Then
   GoTo Terminate 'This exception is reserved for NaNs (not a number)
End If
```

```
'Step 3: Remove bias from exponent

For I = 1 To 8
   Dec_expo = Dec_expo + Val(Mid(Expo, I, 1)) * 2 ^ (8 - I)    'Base 10 value of exponent
Next

Dec_expo = Dec_expo - 127 'Exponent unbiased, expressed in Base 10 (always a signed integer)
Sgn_expo = Sgn(Dec_expo)  'Sign of exponent

'Step 4a: compute Base 10 value of FPU - General numbers, integer and fraction

If Sgn_expo = 1 Then

   BinStr_Int = "1" + Mid(Sgnd, 1, Dec_expo) 'Integer portion of binary number

   For I = 1 To (Dec_expo + 1)
      Dec_Int = Dec_Int + Val(Mid(BinStr_Int, I, 1)) * 2 ^ (Dec_expo + 1 - I)  'Integer Decimal value
   Next

   BinStr_Fr = Mid(Sgnd, (Dec_expo + 1), (23 - Dec_expo))  'Fractional portion of binary number

   For I = 1 To (23 - Dec_expo)
      Dec_Fr = Dec_Fr + Val(Mid(BinStr_Fr, I, 1)) * 2 ^ (-1 * I) 'Fraction Decimal value
   Next

   Dec_Nbr = Dec_Int + Dec_Fr

End If

'Step 4b: compute Base 10 value of FPU - Fractional numbers only

If Sgn_expo = -1 Then

   BinStr_Fr = "1" + Sgnd
   Cnv_expo = Dec_expo

   For I = 1 To 24
      Dec_Fr = Dec_Fr + Val(Mid(BinStr_Fr, I, 1)) * (2 ^ Cnv_expo) 'Fraction Decimal value
   Cnv_expo = Cnv_expo - 1
   Next

   Dec_Nbr = Dec_Fr

End If

If Sign_bit = "1" Then Dec_Nbr = -1 * Dec_Nbr

Terminate:

‘Dec_NBr is the decimal number corresponding to the floating-point binary number

End Sub
```