Description of communication protocols of Txxxx series
DESCRIPTION OF COMMUNICATION PROTOCOLS ........................................3

MODBUS RTU .............................................................................................................3
Supported functions .................................................................................................3
Jumper and button .....................................................................................................3
Description of supported function .........................................................................4
  03 (0x03): Reading of 16-bit registers (Read Holding Registers) .........................4
  04 (0x04): Reading 16-bit input gates (Read Input Registers) .............................5
  16 (0x10): Setting of several 16-bit registers (Write Multiple Registers) .............5
Exception Responses ...............................................................................................7
Exception codes .......................................................................................................8
Modbus CRC ...........................................................................................................8
  Procedure during calculation of Modbus CRC ....................................................8
Modbus registers of the device ..............................................................................9
Example of communication ....................................................................................10
  Reading of temperature, address 0x0031 ............................................................10
  Reading of relative humidity, address 0x0032 ....................................................10
  Reading of computed value, address 0x0033 .....................................................10
  Reading of all values at once, address block 0x0031 to 0x0033 .........................11

PROTOCOL COMPATIBLE WITH ADVANTECH-ADAM STANDARD ........................12
Jumper .....................................................................................................................12
General syntax of commands .................................................................................12
Description of supported functions .....................................................................13
  Configuration of device .......................................................................................13
  Response of the device .......................................................................................13
  Check sum (CRC) ...............................................................................................14
Commands common for all device types ...............................................................14
  Query to adjusted configuration .........................................................................14
  Reading of device name .....................................................................................14
  Reading of firmware version .............................................................................14
Command for devices with temperature OR atmospheric pressure measurement ONLY 14
  Reading of temperature OR atmospheric pressure .............................................14
Commands for combined devices .........................................................................14
  Reading of the temperature* .............................................................................14
  Reading of humidity* .........................................................................................14
  Reading of computed value* .............................................................................14
  Reading of atmospheric pressure* ...................................................................15
How to read device’s unit setting from master device .........................................15
Examples of communication ...............................................................................15
Format of data .......................................................................................................16
  Error states ........................................................................................................17

ARION COMMUNICATION PROTOCOL - AMiT COMPANY ..........................17
Supported frames ....................................................................................................17
Jumper and button ..................................................................................................17
Supported function description ............................................................................18

COMMUNICATION WITH HWG POSEIDON UNITS ........................................19
Jumper .....................................................................................................................19
Description of supported commands ..................................................................20
  Read temperature ...............................................................................................20
  Read humidity ....................................................................................................20
  Read computed value .........................................................................................20
  Read pressure ....................................................................................................20
  Change of device address ..................................................................................20
  Device identification ..........................................................................................21

SETTING OF DEVICE WITH HYPERTERMINAL .........................................21
Description of communication protocols

Devices of Txxxx series with RS485 and RS232 serial interface support to select one of the communication protocols described lower.

Devices are preset from the manufacturer to Modbus RTU communication protocol. If you would like to use different communication protocol then Modbus RTU, it is necessary preset them through user’s software. Use user’s software Tsensor for setting of all device parameters (recommended). It is free to download at www.cometsystem.cz. It supports make the adjustment of the device too. This procedure is described at file „Calibration manual.pdf“ which is installed commonly with the software. Change of some parameters is possible to do without user’s software with Windows Hyperterminal (change of communication protocol, its parameters, LCD display setting). It is described at the end of this document.

Modbus RTU

Control units communicate on master-slave principle in half-duplex operation. Only master can send request and only addressed device responds. During sending of request no other slave station should respond. During communication, data transfer proceeds in binary format. Each Byte is sent as eight bit data word in format: 1 start bit, data word 8 bit (LSB first), 2 stop bits, without parity. Device supports communication speed from 110Bd to 115200Bd. Available address range is from 0 to 255, address 0 is reserved for broadcast and device doesn’t send reply for it. For more detailed communication protocol description see www.modbus.org.

Supported functions

03 (0x03): Reading of 16-bit registers (Read Holding Registers)
04 (0x04): Reading of 16-bit input gates (Read Input Registers)
16 (0x10): Setting of more 16-bit registers (Write Multiple Registers)*

Jumper and button

Jumper and button are located next to connection terminals. If communication protocol Modbus is selected the function of jumper and button is as follows:

- Jumper opened – device memory is protected from writing, from device side it is only enabled to read measured value, writing to memory is disabled (no change of device address, communication speed and LCD setting is enabled)
- Jumper closed – writing to device memory is enabled by means of User’s software or function 16 (0x10): Setting of several 16-bit registers (Write Multiple Registers)
- Up to device firmware version 02.30: (devices support temperature measurement in °C only, don’t support °F)
  - Jumper closed and button pressed for longer than three seconds – causes restoring of manufacturer setting of communication protocol, i.e. sets device address to 01h and communication speed to 9600Bd

---

1 Device sends two stop bits, for receive one stop bit is enough.
2 See detailed description of this function.
• **Since** firmware version **02.31** – included
  (devices support temperature measurement in °C or °F and all with pressure measurement):
  
  o Jumper closed and button pressed for longer than **six** seconds – causes restoring of manufacturer setting of communication protocol, i.e. sets Modbus RTU communication protocol, device address sets to 01h and communication speed to 9600Bd (after button press there is “dEF” message blinking at LCD display. Six seconds later message “dEF” stays shown, it means manufacturer setting of communication protocol is done).
  
  • Jumper opened and button shortly pressed – device goes to Info mode.

**Description of supported function**

**03 (0x03): Reading of 16-bit registers (Read Holding Registers)**

Function serves for reading of values from device. Addresses of available registers are listed in chapter „Modbus registers of the device on page 9.“

**Request:**

```
| FUNCTION | Function code   | 0x03 |
```

```
| DATA | Initial address Hi | 0x?? |
|      | Initial address Lo  | 0x?? |
|      | Number of registers Hi | 0x?? |
|      | Number of registers Lo | 0x?? |
```

**Response:**

```
| FUNCTION | Function code   | 0x03 |
```

```
| DATA | Number of Bytes | 0x?? |
|      | States of register Hi | 0x?? |
|      | States of register Lo  | 0x?? |
|      | ... |
|      | States of register Hi | 0x?? |
|      | States of register Lo  | 0x?? |
```

**Exceptional response:**

```
| FUNCTION | Function code   | 0x83 |
```

```
| DATA | Exception code  | 0x?? |
```

During sending of query to device initial register address and number of registers to read are sent. Register addresses are indexed from zero – **register 0x31 is physically sent as value 0x30, 0x32 as 0x31**… (zero based addressing)
04 (0x04): Reading 16-bit input gates (Read Input Registers)
This function is also possible to use for reading values from device, syntax is the same as with function 03 (0x03): Reading of 16-bit registers. Addresses of available registers are specified in chapter „Modbus registers of the device on page 9.“

16 (0x10): Setting of several 16-bit registers (Write Multiple Registers)
Setting of device address and communication speed is possible to perform by writing to device registers (rest of parameters is possible to set only with User’s software).
Attention! During writing to device registers it is not enabled to write any number of registers. Always below procedure should be strictly followed. If procedure is not followed undoable loss of important settings stored in device can occur! It is strongly recommended to use User’s software to set all device parameters. It is free to download at www.cometsystem.cz.

Request:

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>Function code</th>
<th>0x10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>Initial address Hi</td>
<td>0x??</td>
</tr>
<tr>
<td></td>
<td>Initial address Lo</td>
<td>0x??</td>
</tr>
<tr>
<td></td>
<td>Number of registers Hi</td>
<td>0x??</td>
</tr>
<tr>
<td></td>
<td>Number of registers Lo</td>
<td>0x??</td>
</tr>
<tr>
<td></td>
<td>Number of Bytes (of sent data)</td>
<td>0x??</td>
</tr>
</tbody>
</table>

Response:

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>Function code</th>
<th>0x10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>Initial address Hi</td>
<td>0x??</td>
</tr>
<tr>
<td></td>
<td>Initial address Lo</td>
<td>0x??</td>
</tr>
<tr>
<td></td>
<td>Number of registers Hi</td>
<td>0x??</td>
</tr>
<tr>
<td></td>
<td>Number of registers Lo</td>
<td>0x??</td>
</tr>
</tbody>
</table>

Exceptional response:

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>Function code</th>
<th>0x90</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>Exception code</td>
<td>0x??</td>
</tr>
</tbody>
</table>

For detailed description see next chapter.
Procedure during writing to device registers:

- device address is stored at Modbus address 0x2001 as binary number
- code of communication speed is stored at Modbus address 0x2002

<table>
<thead>
<tr>
<th>Communication speed [Bd]</th>
<th>Code of communication speed [hex]</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>94F2</td>
</tr>
<tr>
<td>300</td>
<td>369D</td>
</tr>
<tr>
<td>600</td>
<td>1B4F</td>
</tr>
<tr>
<td>1200</td>
<td>0DA7</td>
</tr>
<tr>
<td>2400</td>
<td>06D4</td>
</tr>
<tr>
<td>4800</td>
<td>036A</td>
</tr>
<tr>
<td>9600</td>
<td>01B5</td>
</tr>
<tr>
<td>14400</td>
<td>0123</td>
</tr>
<tr>
<td>19200</td>
<td>00DA</td>
</tr>
<tr>
<td>38400</td>
<td>006D</td>
</tr>
<tr>
<td>56000</td>
<td>004B</td>
</tr>
<tr>
<td>57600</td>
<td>0049</td>
</tr>
<tr>
<td>115200</td>
<td>0024</td>
</tr>
</tbody>
</table>

1. Close jumper located next to connection terminals of the device.
2. Read entire area 0x2001 to 0x2040 to master device. At address 0x2040 check sum of entire area is stored. It is calculated as sum of 16bit values from addresses 0x2001 to 0x2039. Stored are lowest 16 bits of this sum - enables to check correct reading of the area.
3. Modify content corresponding to addresses 0x2001 and 0x2002 in master device as required. Setting of the other values should not be changed!
4. Calculate new check sum of entire area, i.e. sum of 16bit values corresponding to values at addresses 0x2001 to 0x2039 and store lowest 16 bits to position corresponding to address 0x2040.
5. Write such modified area together from master device to addresses 0x2001 to 0x2040.
6. Open jumper.

Example: Device with address 01h, communication speed 9600Bd, to change to address 9Fh and 115200Bd

For data area reading the following is sent via link: \textbf{01 03 20 00 00 40 4F FA}

- 01: device address
- 03: command for reading of 16-bit registers
- 20 00: initial address (sent address is indexed from zero, during Modbus request for reading of address 0x2001 physically is sent 0x2000 via link)
- 00 40: read 64 registers (0x40), i.e. 128 Byte
- 4F FA: CRC of Modbus

\textbf{response of the device is:}

\textbf{01 03 80 00 01 01 B5 00 00 30 30 3B 4B 77 D3 BD 35 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 84 44 AA 80 85 07 A8 D0 57 7E 5F 94 F3 DC 00 12 2E DD 78 0C 40 AA 77 D3 F2 C4 00 12 17 78 77 F5 F3 EC 00 12 ED BF 77 D5 4F 10 77 D8 FF FF FF FF FF FF FF FF 40 DE 77 D3 2E F7 78 0C 06 5C 00 01 00 00 00 00 F3 DC 00 12 42 9F 53 2D 2C 8C
device address
03 code of command (reading 16-bit registers)
80 number of Bytes of response (0x80, i.e. 128 Byte)
00 01 address content 0x2001, i.e. original address of device 01h
01 B5 address content 0x2002, i.e. code of corresponding communication speed 9600Bd
...
53 2D check sum of entire area 0x2001 to 0x2039
(0001+01B5+0000+3030+...+0012+429F = 0x532D)
2C 8C CRC of Modbus
Modification of data from read area:
00 9F 00 24 00 00 30 30 3B 4B 77 D3 BD 35 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 84 70 00 00 86 2A 00 00 84 44
AA 80 85 07 A8 D0 57 7E 5F 94 F3 DC 00 12 2E DD 78 0C 40 AA 77 D3 F2 C4 00 12 17 78 77
F5 F3 EC 00 12 ED BF 77 D5 4F 10 77 D8 FF FF FF 40 DE 77 D3 2E F7 78 0C 06 5C 00 01 00 00
00 00 00 00 F3 DC 00 12 42 9F 52 3A
00 9F new address of the device (original address was 00 01)
00 24 new code of corresponding communication speed 115200Bd (original was 01B5)
...
52 3A new check sum of entire area (009F+0024+0000+...+0012+429F = 0x523A)
Write such modified area together from master device to the device, area 0x2001 to 0x2040
01 10 20 00 00 40 80 00 9F 00 24 00 00 30 30 3B 4B 77 D3 BD 35 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 84 70 00 00 86 2A 00 00 84 44
AA 80 85 07 A8 D0 57 7E 5F 94 F3 DC 00 12 2E DD 78 0C 40 AA 77 D3 F2 C4 00 12 17 78 77
F5 F3 EC 00 12 ED BF 77 D5 4F 10 77 D8 FF FF FF 40 DE 77 D3 2E F7 78 0C 06 5C 00 01 00 00
00 00 00 00 F3 DC 00 12 42 9F 52 3A 61 22
01 original device address is still valid, i.e. 01h
10 code of command, setting more 16-bit registers
20 00 initial address
00 40 number of write registers
80 number of Bytes of sent data
61 22 CRC of Modbus
After successful writing to the device, device responds: 01 10 20 00 00 40 CA 39 (still with old address at original communication speed) and after response it sets to new values. In case of different number of data or incorrect check sum of the entire area writing to device is not performed.

Exception Responses
After sending query to the device, master device waits for normal response. After master device query one of the following events can occur:
1. If device receives a query without communication error and query is possible to process, master device receives response.
2. If device does not receive all queries due to communication error, no response is sent. Main program is able to process condition of exceeding of time for query.

3. If device receives a query, but detects communication error (CRC), no response is sent. Main program is able to process condition of exceeding of time for query.

4. If device receives a query without communication error, but cannot process it, master device receives exception response, which informs master device on error nature.

**Exception Response**
- has two fields to distinguish it from normal response:
  1. Function code field
  2. Data field.

**ad1 Function code field**
In normal response of slave device function code of original query corresponds to function code of response. All function codes have most significant bit (MSB) equal to 0. In exception response slave device sets most significant bit of function code to 1. Main station recognizes exception response by means of this bit and can check data field for exception code.

**ad2 Data field**
Device returns exception code in exception response in data field. Event causing exception is determined this way.

**Exception codes**

- **0x01** Invalid function. Function code in query is not allowed action for device.
- **0x02** Invalid data address. Data address received in query is not allowed address for device.

**Modbus CRC**
Check sums of entire Modbus messages are mostly automatically inserted to the end of request by communication programs themselves. In case there is a need to insert to generate Modbus CRC to program itself, the way of calculation is as follows:

**Procedure during calculation of Modbus CRC**

1. To fill 16-bit register with value 0xFFFF (all bits set to 1). Let us call this register „CRC register“.
2. Perform logic function Exclusive OR with first eight bit message Byte with lower eight bits of CRC register. Store result to CRC register.
3. Shift content of CRC register of one bit to the right (towards to LSB), enter 0 as upper bit of CRC. Memorize values of lowest shifted bit (LSB).
4. If LSB was 0, then repeat step 3 (other shift). If LSB was 1, then perform Exclusive OR CRC register with value 0xA001.
5. Repeat steps 3 and 4 as long as eight shifts proceed. After eight shifts eight bit Byte is processed.
6. Repeat steps 2 to 5 to the next of eight bit Byte of message as long as all Bytes are proceeded.
7. In the end after processing of all message Bytes check sum value is stored in CRC register.
8. During connection of check sum to the message lower Byte of CRC register is sent as first, then upper Byte of CRC register.
## Modbus registers of the device

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>Address[hex]</th>
<th>Address[dec]</th>
<th>Format</th>
<th>Size</th>
<th>Status</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Measured temperature</td>
<td>[°C] [°F]*</td>
<td>0x0031</td>
<td>49</td>
<td>Int*10</td>
<td>BIN16</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>Address of device</td>
<td>[-]</td>
<td>0x2001</td>
<td>8193</td>
<td>Int</td>
<td>BIN16</td>
<td>R/W*</td>
<td></td>
</tr>
<tr>
<td>Code of communication speed</td>
<td>[-]</td>
<td>0x2002</td>
<td>8194</td>
<td>Int</td>
<td>BIN16</td>
<td>R/W*</td>
<td></td>
</tr>
<tr>
<td>Serial number of device Hi</td>
<td>[-]</td>
<td>0x1035</td>
<td>4150</td>
<td>BCD</td>
<td>BIN16</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>Serial number of device Lo</td>
<td>[-]</td>
<td>0x1036</td>
<td>4151</td>
<td>BCD</td>
<td>BIN16</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>Version of Firmware Hi</td>
<td>[-]</td>
<td>0x3001</td>
<td>12289</td>
<td>BCD</td>
<td>BIN16</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>Version of Firmware Lo</td>
<td>[-]</td>
<td>0x3002</td>
<td>12290</td>
<td>BCD</td>
<td>BIN16</td>
<td>R</td>
<td></td>
</tr>
</tbody>
</table>

And next only for combined devices with relative humidity measurement

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>Address[hex]</th>
<th>Address[dec]</th>
<th>Format</th>
<th>Size</th>
<th>Status</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Measured relative humidity</td>
<td>[%]</td>
<td>0x0032</td>
<td>50</td>
<td>Int*10</td>
<td>BIN16</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>Computed value</td>
<td>[°C]</td>
<td>0x0033</td>
<td>51</td>
<td>Int*10</td>
<td>BIN16</td>
<td>R</td>
<td></td>
</tr>
</tbody>
</table>

And next only for combined devices with atmospheric pressure measurement

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>Address[hex]</th>
<th>Address[dec]</th>
<th>Format</th>
<th>Size</th>
<th>Status</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Atmospheric pressure</td>
<td></td>
<td>0x0034</td>
<td>52</td>
<td>Int*10</td>
<td>BIN16</td>
<td>R</td>
<td></td>
</tr>
</tbody>
</table>

**Explanation:**
- * depends on device type and setting (User’s software)
- Int*10 register is in format integer*10
- R register is designed only for reading
- W* register is designed for writing, for details see description function **16 (0x10): Setting of several 16-bit registers (Write Multiple Registers)**
- X register addresses are indexed from zero – register 0x31 is physically sent as value 0x30, 0x32 as 0x31 (zero based addressing).

**Note:** In case there is a need for reading of measured values from device with higher resolution than one decimal, measured values in device are stored also in „Float“ format, which is not directly compatible with IEEE754.
Example of communication

In all examples communication with device at address 01h is supposed

**Reading of temperature, address 0x0031**

- Modbus command:
  - device address: 01
  - reading 16-bit registers 03
  - initial address Hi 00
  - initial address Lo 31
  - number read registers Hi 00
  - number read registers Lo 01
- Via link is physically sent: 01 03 00 30 00 01 84 05
- Received response from device: 01 03 02 00 F4 B9 C3
  - device address: 01
  - reading 16-bit registers 03
  - Number Byte 02
  - State of register Hi 00
  - State of register Lo F4 (0x00F4 i.e. 244 i.e. 24.4 °C)
  - Modbus CRC Lo B9
  - Modbus CRC Hi C3

**Reading of relative humidity, address 0x0032**

- Modbus command:
  - device address: 01
  - reading 16-bit registers 03
  - initial address Hi 00
  - initial address Lo 32
  - number read registers Hi 00
  - number read registers Lo 01
- Via link is physically sent: 01 03 00 31 00 01 D5 C5
- Received response from device: 01 03 02 01 6C B9 F9
  - device address: 01
  - reading 16-bit registers 03
  - Number Byte 02
  - State of register Hi 01
  - State of register Lo 6C (0x016C i.e. 364 i.e. 36.4 %RH)
  - Modbus CRC Lo B9
  - Modbus CRC Hi F9

**Reading of computed value, address 0x0033**

- Modbus command:
  - device address: 01
  - reading 16-bit registers 03
  - initial address Hi 00
  - initial address Lo 33
number read registers Hi 00
number read registers Lo 01

Via link is physically sent: 01 03 00 32 00 01 25 C5

Received response from device: 01 03 02 FF 3E 78 64
device address: 01
reading 16-bit registers 03
Number Byte 02
State of register Hi FF
State of register Lo 3E (0xFF3E i.e. -194 i.e. -19.4)
Modbus CRC Lo 78
Modbus CRC Hi 64

**Reading of all values at once, address block 0x0031 to 0x0033**

Modbus command:

device address: 01
reading 16-bit registers 03
initial address Hi 00
initial address Lo 31
number read registers Hi 00
number read registers Lo 03

Via link is physically sent: 01 03 00 30 00 03 05 C4

Received response from device: 01 03 06 FF C4 01 14 FF 38 C5 71
device address: 01
reading 16-bit registers 03
Number Byte 06
State of register Hi FF
State of register Lo C4 (0xFFC4 i.e. -60 i.e. -6.0 °C)
State of register Hi 01
State of register Lo 14 (0x0114 i.e. 276 i.e. 27.6 %RH)
State of register Hi FF
State of register Lo 38 (0xFF38 i.e. -200 i.e. -20.0 °C)*
Modbus CRC Lo C5
Modbus CRC Hi 71

* Computed value is preset by factory to Dew Point Temperature, to choice use User’s software
Protocol compatible with Advantech-ADAM standard

Control units communicate on master-slave principle in half-duplex operation. Only master can send requests and only addressed device responds. During sending request any of slave devices should respond. During communication data is transferred in ASCII format (in characters). Each Byte is sent as two ASCII characters (value 0x2F is sent as pair of characters 0x32, 0x46, i.e. characters „2“ and „F“). All commands and values MUST be entered in CAPITAL LETTERS! Device supports communication speed from 1200Bd to 115200Bd, parameters of communication link are 1 start bit + eight bit data word (LSB first) + 1 stop bit, without parity. Commands for reading of measured values are different for devices with temperature measurement only and for combined devices with temperature and relative humidity measurement – see later.

Jumper

Its location is described into User’s manual. If communication protocol compatible with standard Advantech-ADAM is selected, its function is the following:

- If jumper during switching ON the power is closed, device always communicates with following parameters regardless stored setting in the device: communication speed 9600 Bd, without check sum, device address 00h
- If jumper during switching ON the power is not closed, device communicates in accordance with stored setting.
- If jumper is closed during device operation, device temporarily changes its address to 00h, it will communicate in the same communication speed as before closing jumper and will communicate without check sum. After jumper is opened setting of address and check sum is reset in accordance with values stored in device.
- Communication speed and check sum are possible to change only if jumper is closed (see chapter Configuration of device).
- Since firmware version 02.31 – included (devices support temperature measurement in °C or °F and all with pressure measurement):
  Jumper closed and button pressed for longer than six seconds – causes restoring of manufacturer setting of communication protocol, i.e. sets Modbus RTU communication protocol, device address sets to 01h and communication speed to 9600Bd (after button press there is “dEF” message blinking at LCD display. Six seconds later message “dEF” stays shown, it means manufacturer setting of communication protocol is done). Older FW versions DON’T support this feature.

General syntax of commands

[ distinguishing character ][ device address ][ command ][ data ][ check sum ][ CR ]

Valid distinguishing characters towards to device are: $, #, %

Device address contents 2 ASCII bytes in hexadecimal code (upper case letters) representing one byte of binary address (e.g. „3“ „F“ corresponds to address 3Fh, i.e. 63, is sent as 0x33, 0x46)

Check sum: enabled to switch ON/switch OFF

CR ... 1 byte (0Dh)
Description of supported functions

Configuration of device

Syntax of command: \%AANNTTCCFF cr

Meaning of symbols:

AA … current address of device 00…FF (hexadecimal)
NN … new address of device 00…FF (hexadecimal)
TT … code of device: 2Bh… devices with temperature OR atmospheric pressure measurement ONLY
      2Ch… combined devices of temperature and/or humidity and/or atmospheric pressure

CC … code of communication speed

<table>
<thead>
<tr>
<th>Code</th>
<th>Speed [Bd]</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td>1200</td>
</tr>
<tr>
<td>04</td>
<td>2400</td>
</tr>
<tr>
<td>05</td>
<td>4800</td>
</tr>
<tr>
<td>06</td>
<td>9600</td>
</tr>
<tr>
<td>07</td>
<td>19200</td>
</tr>
<tr>
<td>08</td>
<td>38400</td>
</tr>
<tr>
<td>09</td>
<td>57600</td>
</tr>
<tr>
<td>0A</td>
<td>115200</td>
</tr>
</tbody>
</table>

FF … data format and check sum:

D7 D6 D5 D4 D3 D2 D1 D0

xxxx xx00 format "Engineering units"

x0xx xx00 check sum switched OFF

x1xx xx00 check sum switched ON

- Communication speed and check sum are possible to change only if jumper is closed
  - Change in communication speed activates only after device power is switched OFF and switched ON again.
  - Change in setting of check sum activates immediately after jumper is opened

- If address is changing
  - and jumper is closed, device responds with address 00h again, and newly set address will be activated after jumper is opened.
  - and jumper is not closed, change is activated immediately.

- If attempt occurs to write incorrect data to the device (and syntax is correct), device responds with error message.

Response of the device

1. If syntax of command is not correct, device does not respond at all (e.g. no check sum is received though it is switched ON, check sum is not correct, string is not complete or contents invalid character).

2. If syntax is correct, but required operation is not correct, device returns error message in format
   ? AA cr
this state appears if we try to change communication speed and check sum and jumper is not closed.

3. If command is executed, device responds:
   ! AA cr

**Check sum (CRC):** it is the sum of all characters before it, its lowest byte is applied.

**Commands common for all device types**

**Query to adjusted configuration**
- Syntax command: $AA2 cr
- Response: !AATTCCFF cr 
  symbols correspond with paragraph "Configuration of device"

**Reading of device name**
- Syntax command: $AAM cr
- Response: !AATxxxx cr (AAT3411 cr accordingly with device model)

**Reading of firmware version**
- Syntax command: $AAF cr
- Response: !AA(version) cr reads version number of device firmware

Command for devices with temperature OR atmospheric pressure measurement ONLY

**Reading of temperature OR atmospheric pressure**
- Syntax command: #AA cr
- Response: > (temperature/pressure) cr (e.g. >-012.30 cr)

**Commands for combined devices**
(with temperature and/or relative humidity and/or atmospheric pressure measurement)

**Reading of the temperature***
- Syntax command: #AA0 cr
- Response: > (temperature) cr (e.g. >-012.30 cr)

**Reading of humidity***
- Syntax command: #AA1 cr
- Response: > (humidity) cr (e.g. >+044.30 cr)

**Reading of computed value***
- Syntax command: #AA2 cr
- Response: > (computed value) cr (e.g. >+004.30 cr)
Reading of atmospheric pressure*

Syntax command:  \#AA3 cr

Response:  > (atmospheric pressure) cr  (e.g. >+1013.2 cr)

Attention!

Atmospheric pressure is stored in next format (depends on selected unit):

<table>
<thead>
<tr>
<th>Unit**</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>hPa</td>
<td>&gt;+1013.1</td>
</tr>
<tr>
<td>PSI</td>
<td>&gt;+14.123</td>
</tr>
<tr>
<td>inHg</td>
<td>&gt;+028.12</td>
</tr>
<tr>
<td>mBar</td>
<td>&gt;+1013.1</td>
</tr>
<tr>
<td>oz/in²</td>
<td>&gt;+0225.1</td>
</tr>
<tr>
<td>mmHg</td>
<td>&gt;+0728.1</td>
</tr>
<tr>
<td>inH₂O</td>
<td>&gt;+0380.1</td>
</tr>
<tr>
<td>kPa</td>
<td>&gt;+101.12</td>
</tr>
</tbody>
</table>

* Depends on the device type and setting. If the value is not supported, returns still ? AA cr

** Depends on device setting (User’s software)

How to read device’s unit setting from master device

If you have installed devices into your system and you need to read device’s unit setting for temperature (°C or °F) or for pressure (hPa, PSI, inHg, mBar, oz/in², mmHg, inH₂O or kPa) it is possible (with Modbus protocol only). It is necessary to read from Modbus address 0x203F (8255 dec). Received value can discover unit setting by next rules:

- Combination of the two lowest bits Bit0 and Bit1 describes temperature setting: 0...°C, 1...°F, 2 and 3 not used yet.
- Combination of Bits 2, 3 and 4 describes pressure unit: 0 = hPa, 1 = PSI, 2 = inHg, 3 = mBar, 4 = oz/in², 5 = mmHg, 6 = inH₂O, 7 = kPa.

For example received value is:

- 0x0000, it means hPa and °C
- 0x0015, it means mmHg and °F
- 0x001C, it means kPa and °C

Examples of communication

Example 1: Change of device address during operation (without closed jumper, CRC switched OFF)

Device, which had address 23h is configured to address 24h, speed 9600 Bd, without CRC, setting of communication speed and CRC must not change (setting of communication speed and CRC is not possible to change without closed jumper).

Command:  %23242C0600 cr
Response:  !24 cr

Example 2: Reading of temperature from device with temperature measurement only, without closed jumper, device address 01h, without check sum.
- without check sum
Command:  #01 cr
Response:  > +020.50 cr

- with check sum:
Command:  #01 84 cr
where it is sent:  23 30 31 38 34 0D
calculation of CRC: 23h+30h+31h = 84h, then CRC = 84h, it is sent as 0x38, 0x34
Response:  >+020.50 8E cr
where it is sent:  3E 2B 30 32 30 2E 35 30 38 45 0D
calculation of CRC: 3Eh+2Bh+30h+32h+30h+2Eh+35h+30h=18Eh then CRC = 8Eh, then it is sent as 0x38, 0x45

Example 3:  Reading of temperature from combined device (temperature and relative humidity measurement) without closed jumper, device address 01h.

- without check sum
Command:  #010 cr
Response:  > +020.50 cr

- with check sum:
Command:  #010 B4 cr
where it is sent:  23 30 31 30 42 34 0D
calculation of CRC: 23h+30h+31h+30h = B4h, then CRC = B4h, it is sent as 0x42, 0x34
Response:  >+020.50 8E cr
where it is sent:  3E 2B 30 32 30 2E 35 30 38 45 0D
calculation of CRC: 3Eh+2Bh+30h+32h+30h+2Eh+35h+30h=18Eh then CRC = 8Eh, then it is sent as 0x38, 0x45

Example 4:  Setting of device to address 9F, communication speed remains 9600 Bd, check sums are switched ON (jumper should be closed because setting of CRC will be changed):
- with jumper, device always reports from address „00“ without CRC
Command:  %009F2C0640 cr
Response:  !00
After jumper is opened device address changes to 9Fh

Format of data

Device uses data format „Engineering units“, i.e. fixed decimal point. Temperature, humidity and computed value are displayed with 2 digits behind decimal point, second digit behind decimal point is always zero. Atmospheric pressure display depends on selected pressure unit, see command description for "Reading of atmospheric pressure*" on page 15. General writing: „>±xxx.x0 cr“ (except of pressure)
Examples: 

>-050.20 cr    >+000.00 cr    >+025.80 cr

**Error states**

>-0000 cr  lower limit of temperature, error in measurement of humidity, computed value or atmospheric pressure.

>+9999 cr  upper limit of temperature, error in measurement of humidity and computed value. Is NOT used for atmospheric pressure error alarm.

---

**ARION communication protocol - AMiT company**

The device supports communication protocol ARION version 1.00, simplex and half duplex modes and communication speeds 9600, 19200, 38400 and 57600 Baud.

### Supported frames

<table>
<thead>
<tr>
<th>Frame Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x20</td>
<td>ACK  Positive response</td>
</tr>
<tr>
<td>0x21</td>
<td>NAK  Negative response</td>
</tr>
<tr>
<td>0x22</td>
<td>STRQ (Resp.req. must be 1)</td>
</tr>
<tr>
<td>0x23</td>
<td>STT  Slave status, mode, guard time</td>
</tr>
<tr>
<td>0x24</td>
<td>MODE</td>
</tr>
<tr>
<td>0x26</td>
<td>VRRQ (Resp.req. must be 1)</td>
</tr>
<tr>
<td>0x27</td>
<td>VER  Response of firmware and protocol version</td>
</tr>
<tr>
<td>0x28</td>
<td>IDRQ (Resp.req. must be 1)</td>
</tr>
<tr>
<td>0x29</td>
<td>MID  Module identification</td>
</tr>
<tr>
<td>0x2A</td>
<td>FTRQ (Resp.req. must be 1)</td>
</tr>
<tr>
<td>0x2B</td>
<td>FTL  List of supported frames</td>
</tr>
<tr>
<td>0x2C</td>
<td>CLRQ (Resp.req. must be 1)</td>
</tr>
<tr>
<td>0x2D</td>
<td>CLS  Class of module</td>
</tr>
<tr>
<td>0x30</td>
<td>DTRQ (Resp.req. must be 1)</td>
</tr>
<tr>
<td>0x56</td>
<td>data</td>
</tr>
<tr>
<td>0x58</td>
<td>data</td>
</tr>
<tr>
<td>0x80+Address+Response Request</td>
<td>Start of frame</td>
</tr>
<tr>
<td>0x80</td>
<td>Stop</td>
</tr>
</tbody>
</table>

---

**Jumper and button**

Since firmware version 02.31 – included

(devices support temperature measurement in °C or °F and all with pressure measurement):

> Jumper closed and button pressed for longer than six seconds – causes restoring of manufacturer setting of communication protocol, i.e. sets Modbus RTU communication protocol, device address sets to 01h and communication speed to 9600Bd (after button press there is “dEF” message blinking at LCD display. Six seconds later message “dEF” stays
shown, it means manufacturer setting of communication protocol is done). Older FW versions DON’T support this feature.

**Supported function description**

**0x24 MODE – Set mode**

Device supports next modes:
- 0x00, 0x01 Simplex
- 0x02, 0x03 Half Duplex over RS232, RS485 line

Lost of connection detection (Guard time) is always set to zero, it means lost of connection is not detected, guard time value sent by command is not important, it is not checked.

**0x56 – 24 analog values by 14 bit:**
The values are sent in data sequence: temperature, relative humidity, computed value, the rest values are always zero. Range of all values is restricted from -819.2 to +819.1. For atmospheric pressure value it is necessary to add 300hPa offset to read value. Right values are into range from -819.1 to +819.0, limit values are reserved for over range or error states indication, where -819.2 = Error2, 819.1 = Error1. Application setup follows:

- Bipolar converter: Yes
- Range  8192
- ElMin  -8192
- ElMax  8191
- PhysMin -819.2
- PhysMax  819.1

**0x58 – 8 analog values by 21 bit:**
Similarly as 0x56, range of values is from -999.9 to +999.9 (except of atmospheric pressure). Limit values are reserved for over range or error states indication. For atmospheric pressure there is the +999.9 right value (no error) and corresponds to atmospheric pressure. There is not necessary to add any offset to pressure value, maximal value is 1999.9)

For detailed protocol description see ARION protocol description at [www.amit.cz](http://www.amit.cz)

Actual adjusted address of device is available by press of button next to connection terminals (jumper should be opened). Then device displays its address on LCD display at decimal base. Next press of button exits this mode and actual measured values are displayed. While address shown, device don’t communicate.

---

3 Selection of computed value type is provided by User’s Software
Communication with HWg Poseidon units

This communication protocol is supported by Txxxx series since Firmware version 02.33 (included) and newer.

Protocol restrictions:
Measured temperature is transferred always at °C (doesn’t support °F), next conversion from °C to °F is possible inside of HWg Poseidon unit.
Computed value is selectable between dew point temperature and absolute humidity. Rest of computed values aren’t supported by this communication protocol. Selection is provided by user’s software Tsensor.
Pressure value is transferred at kPa with one decimal point resolution.

Possible device address range is from „A“ to „Z“ and from „a“ to „z“. There is possible to show actual device address on LCD display with using of “Info mode” (see User’s manual of device). Device address is shown as number corresponding with ASCII code of address. It is first address used by the device. Device occupies next number of addresses by device type (number of measured values). Addresses „T“ and „t“ are reserved and can’t be used as device address. These addresses are skipped!
Address change with using of command „T#<new address>“ is permitted for first ten seconds after power up only! Trying to change address latter brings write error message (protection of random address change during activity).

Next table shows four examples of address assignment to each measured values according to device type (number of measured values). First column shows example when address of device is “A”, second column address “R”, third address “h” and fourth address “k”.

Example of address assignment by device types to each measured value:

<table>
<thead>
<tr>
<th>Values measured by the device (device type)</th>
<th>Presets address (see „Info mode“)</th>
</tr>
</thead>
<tbody>
<tr>
<td>temperature</td>
<td>A      R      h      k</td>
</tr>
<tr>
<td>temperature</td>
<td>A      R      h      k</td>
</tr>
<tr>
<td>relative humidity</td>
<td>B      S      i      l</td>
</tr>
<tr>
<td>computed value</td>
<td>C      U      j      m</td>
</tr>
<tr>
<td>temperature</td>
<td>A      R      h      k</td>
</tr>
<tr>
<td>pressure</td>
<td>A      R      h      k</td>
</tr>
<tr>
<td>temperature</td>
<td>A      R      h      k</td>
</tr>
<tr>
<td>relative humidity</td>
<td>B      S      i      l</td>
</tr>
<tr>
<td>computed value</td>
<td>C      U      j      m</td>
</tr>
<tr>
<td>pressure</td>
<td>D      V      k      n</td>
</tr>
</tbody>
</table>

Jumper

- Its placement is described at user’s manual of the device.
If jumper is closed and the button is pressed for longer than six seconds then factory default of communication protocol is done – it is Modbus RTU protocol, device address 01h and communication speed 9600Bd („dEF“ text is blinking at LCD after button is pressed, six second latter it stays on, communication factory default is done).
Description of supported commands

Examples are for device with measuring of temperature, relative humidity, computed value and pressure with address set to „A“

Read temperature
Request for temperature:   T<address>I       example: TAI
Device response       *<address><temperature><CR>   example: *A+020.5C
Device response (error)   *<address><Err><CR>   example: *AErr
      <address> character from „A“ to „Z“ or „a“ to „z“ except of „T“ and „t“
      <CR> character 0xD or 13 dec – End Of Line
      <temperature> syntax *A+020.5C

Read humidity
Request for humidity:   T<address>I   example: TBI
Device response       *<address><humidity><CR>   example: *B062.1%
Device response (error)   *<address><Err><CR>   example: *BErr
      <address> character from „A“ to „Z“ or „a“ to „z“ except of „T“ and „t“
      <CR> character 0xD or 13 dec – End Of Line
      <humidity> syntax *B062.1%

Read computed value
Type of computed value is described by one character placed at the end of response - „d“ for dew point temperature and „h“ for absolute humidity.
Request for value: T<address>I       example: TCI
Device response       *<address><computed_value><CR>   example: *C+013.3d (for dew point temperature)
      example: *C+011.6h (for absolute humidity)
Device response (error) *<address><Err><CR>   example: *CErr
      <address> character from „A“ to „Z“ or „a“ to „z“ except of „T“ and „t“
      <CR> character 0xD or 13 dec – End Of Line
      <computed_value> syntax *C+013.3d
      *C+011.6h

Read pressure
Request for pressure: T<address>I       example: TDI
Device response       *<address><pressure><CR>   example: *D+101.3P
Device response (error)   *<address><Err><CR>   example: *DErr
      <address> character from „A“ to „Z“ or „a“ to „z“ except of „T“ and „t“
      <CR> character 0xD or 13 dec – End Of Line
      <pressure> syntax *D+101.3P i.e. 101.3kPa

Change of device address
Assign new address: T#< new address >   example: T#A
Device response       *< new address >OK<CR>   example: *AOK
Device response (error)   *< old address >Err<CR>   example: *BErr
      <address> character from „A“ to „Z“ or „a“ to „z“ except of „T“ and „t“
      <CR> character 0xD or 13 dec – End Of Line
Address change with using of command „T#< new address>“ is permitted for first ten seconds after power up only! Trying to change address latter brings write error
message (protection of random address change during activity). **This command can be used when the communication bus is connected with one device only (no other devices connected to bus)!**

**Device identification**

Request for device type: \(T\langle \text{address} \rangle?\)  

Device response: \(*\langle \text{address} \rangle \langle \text{Identification} \rangle \langle \text{CR} \rangle\)  

\(\langle \text{Identification} \rangle \) “Txxxx ... device type, 0233 ... FW version  

\(\langle \text{CR} \rangle\) character 0xD or 13 dec – End Of Line

**Setting of device with Hyperterminal**

There is possible to set some settings of device with digital communication interface RS232/RS485 with Hyperterminal (without user’s software). Each new setting of device parameters is stored in its internal memory\(^4\). If modification of device setting is needed (communication protocol, its parameters, setting of LCD display), it is recommended to use of **User’s software** available at [www.cometsystem.cz](http://www.cometsystem.cz) (providing all settings of the device parameters) or proceed in following way (support only basic setting):

- Connect device with PC, for devices with RS485 interface use converter RS485/RS232 (with automatic switching control).
- Run program „HyperTerminal“ on your computer – it is a part of Windows operating system (Start \(\rightarrow\) Program Files \(\rightarrow\) Accessory \(\rightarrow\) Communication \(\rightarrow\) HyperTerminal) and set transmission parameters. I.e. enter name of new connection (e.g. Transmitter), press button OK, window „Connect“ appears. In field „Connect by means of“ select COM port, which device is connected to (e.g. COM1) and press OK. In following window „COM1 – Features“ enter to field:

  - „Bits per second“ 9600
  - „Data bits“ 8
  - „Parity“ None
  - „Number of stop-bits“ 1
  - „Flow control“ None

Continue by pressing OK.

---

\(^4\) During writing to device memory no power failure should appear! Such failure can cause damage of other settings stored in the device!
Select item „File“, „Features“ in menu. Select bookmark „Setup“, in the window, click on „ASCII Setup...“ and modify all items in accordance with following figure and press OK.

For industrial device types unscrew 4 screws of device cover and remove it. For indoor device types remove the front part (see Instruction manual of the device). Then plug in the jumper.

Keep the button inside the device pressed, only now you can connect power voltage and then release the button. If device is powered directly from PC communication port (for interface RS232), it is necessary first to disconnect communication cable, then to keep the button pressed, to connect communication cable (and also power) and only now release the button.

LCD display of the device displays „PC“. 
• Running HyperTerminal displays main menu.

Select required items to set the device:

0 … Displays actual setting of the device
1 … Selection of communication protocol and setting of its parameters (device address, communication speed…). For entering numeric values use digits 0 to 9 and characters „A“ to „F“ (upper case letters), device address is entered always as two digits hexadecimal number (no decimal, e.g. address 11 i.e. Bh must be entered as 0B).
2 … Setting of LCD display of the device (display switched ON/switched OFF, selection of displayed values)
8 … Setting communication to default value from manufacturer (Modbus, address 01h, 9600 Bd)
9 … End of setting

All changes in setting are continuously stored automatically.

• If needed close device again with cover.