

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*..... 3
 - 03 (0x03): Reading of 16-bit registers (Read Holding Registers)..... 3
 - 04 (0x04): Reading 16-bit input gates (Read Input Registers)..... 4
 - 16 (0x10): Setting of several 16-bit registers (Write Multiple Registers)..... 4
 - 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 barometric pressure measurement ONLY*. 14
 - Reading of temperature OR barometric pressure 14
 - Commands for combined devices* 14
 - Reading of the temperature* 14
 - Reading of humidity* 14
 - Reading of computed value* 14
 - Reading of barometric pressure* 15
 - Examples of communication*..... 15
 - Format of data*..... 16
 - Error states..... 16
- ARION COMMUNICATION PROTOCOL - AMIT COMPANY 17
 - Supported frames*..... 17
 - Supported function description* 18

Description of communication protocols

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

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 bits¹, 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)*²

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)*
- 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
- 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.“

¹ Device sends two stop bits, for receive one stop bit is enough.

² See detailed description of this function.

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:

FUNCTION	Function code	0x10
----------	---------------	------

DATA	Initial address Hi	0x??
	Initial address Lo	0x??

	Number of registers Hi	0x??
	Number of registers Lo	0x??

	Number of Bytes (of sent data)	0x??
--	-----------------------------------	------

Response:

FUNCTION	Function code	0x10
----------	---------------	------

DATA	Initial address Hi	0x??
	Initial address Lo	0x??

	Number of registers Hi	0x??
	Number of registers Lo	0x??

Exceptional response:

FUNCTION	Function code	0x90
----------	---------------	------

DATA	Exception code	0x??
------	----------------	------

For detailed description see next chapter.

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

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

And next only for combined devices with relative humidity measurement

Variable	Unit	Address[hex] ^X	Address[dec] ^X	Format	Size	Status
Measured relative humidity	[%]	0x0032	50	Int*10	BIN16	R
Computed value *	[°C]	0x0033	51	Int*10	BIN16	R

And next only for combined devices with barometric pressure measurement

Variable	Unit*	Address[hex] ^X	Address[dec] ^X	Format	Size	Status
Barometric pressure	hPa	0x0034	52	Int*10	BIN16	R
	PSI			Int*1000		
	inHg			Int*100		
	mBar			Int*10		
	oz/in ²			Int*10		
	mmHg			Int*10		
	inH ₂ O			Int*10		
	kPa			Int*100		

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

Jumper is located next to connection terminals. 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*).

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 barometric pressure measurement ONLY

2Ch... combined devices of temperature and/or humidity and/or barometric pressure

CC ... code of communication speed

Code	Speed [Bd]
03	1200
04	2400
05	4800
06	9600
07	19200
08	38400
09	57600
0A	115200

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: **!AATCCFF 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 barometric pressure measurement ONLY

Reading of temperature OR barometric 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 barometric 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 barometric pressure*

Syntax command: **#AA3 cr**

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

Attention!

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

Unit**	Format
hPa	>+1013.1
PSI	>+14.123
inHg	>+028.12
mBar	>+1013.1
oz/in ²	>+0225.1
mmHg	>+0728.1
inH ₂ O	>+0380.1
kPa	>+101.12

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

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

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. Barometric pressure display depends on selected pressure unit, see command description for ”

Reading of barometric 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 barometric pressure.

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

0x20	ACK	Positive response
0x21	NAK	Negative response
0x22	STRQ (Resp.req. must be 1)	Status request
0x23	STT	Slave status, mode, guard time
0x24	MODE	Set Mode
0x26	VRRQ (Resp.req. must be 1)	Request for firmware and protocol version
0x27	VER	Response of firmware and protocol version
0x28	IDRQ (Resp.req. must be 1)	Module identification request
0x29	MID	Module identification
0x2A	FTRQ (Resp.req. must be 1)	Request for list of supported frames
0x2B	FTL	List of supported frames
0x2C	CLRQ (Resp.req. must be 1)	Request for class of module
0x2D	CLS	Class of module
0x30	DTRQ (Resp.req. must be 1)	Data request
0x56	data	24 analog values by 14 bit
0x58	data	8 analog values by 21 bit
0x80+Address+Response Request		Start of frame
0x80	Stop	End of frame

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 value³, the rest values are always zero. Range of all values is restricted from -819.2 to +819.1. For barometric 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 follow: 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 barometric pressure). Limit values are reserved for over range or error states indication. For barometric pressure there is the +999.9 right value (no error) and corresponds to barometric 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

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

Notes:
