DRESSER
ACTARIS GAS

Implementation of Modbus protocol in Dresser devices of the Corus Evo+ and Corus Evo Compact

810-2005-00
AA

EN **User manual**

Contents

Terms

char	8-bit integer value with sign
CRC	cyclic security code
dec	decimal representation
EEPROM	electrically deletable and programmable memory
float	32-bit value in float index line
hex	hexadecimal representation
int	16-bit integer value with sign
Modbus	communication protocol used in the area of controlling and automating technique supported by many manufacturers
MSB	the most significant byte
LSB	the least significant byte
RS 485	standard for data transmission using symmetric transmission line allowing realization of network with many devices
RTU	one of the regimes for data transmission using the communication protocol Modbus (Remote Terminal Unit)
uchar	8-bit integer value without sign
uint	16-bit integer value without sign
device	Corus Evo+ Corus Evo Compact
slave unit (slave)	slave device from the point of communication on bus bar using Modbus protocol, it is always the device in this manual
master unit (master)	controlling unit from the point of communication on bus bar using Modbus protocol, usually controlling computer or superior system
Modbus map	map of internal registers, in this implementation depends on device parameters

1. Introduction

In this manual may be found description of implementation of Modbus-RTU protocol for Corus Evo+, Corus Evo Compact.

All general information about protocols may be found in original literature [1].

In the text is used Czech terminology. English equivalent is mentioned in brackets during first presence of term. Digital values in the text are mentioned in decimal representation as far as not sad otherwise. Whether the values are mentioned in hexadecimal representation then are marked with suffix „h" or in the header of table „hex".

In the first part of text are mentioned some general characteristics given by definition of implemented protocol with attention to particularities, see the chapter 2 Survey of Modbus protocol.

Firmware of the device predominates with huge variability of configuration. Basic element of Modbus protocol is map of internal registers (furthermore Modbus map) which display quantities of the device. Due to the great variability of the device, the possibility to modify the Modbus map is implemented here. This is done through pre-prepared templates. For more details, see 3 Templates

2. Survey of Modbus protocol

2.1. Physical layer

Communication is running on bus bar RS-232 or RS-485, asynchronous serial communication, communication speed adjustable in the range 9600Bd – 38400Bd, 1 start bit, 8 data bit, 1 stop bit, without parity.

2.2. Transmission in Modbus network

Modbus protocol is using technique master-slave for sharing of access to the bus bar. On bus bar may be only one master unit and maximally 247 slave units. Only controlling unit may begin transmission of messages. There are two types of messages. In case of using transmission of type ´query / response´ the controlling unit is sending query for only one addressed slave unit which will send response after processing of the query. In case of using transmission of type ´broadcast / no response´ the broadcast is determined for all slave units which will process the query but will not send the response.

Query, response and broadcast are transferred on bus bar as frames with fixed defined structure. In case of transmission of RTU regime, each frame begins with idle interval on bus bar with the length equivalent to the time of transmission of at least 3,5 characters after which will start with transmission of scope. Address field is transmitted as first followed by function field, data field and with block with check sum (CRC field). After last transmitted character follows idle interval on bus bar of the length of 3,5 characters which indicates the end of the frame. After expiration of this interval may be transmitted next frame.

Complete frames may be transmitted as continuous stream without space. Whether idle period will appear inside of the frame between the characters longer than time corresponding to transmission of 1,5 character then receiving unit will ignore received part of the frame and following character will be considered as first character of new frame.

Format of frame

Start idle length 3,5 characters	Address block 1 character	Functional block 1 character	Data block n characters	CRC block 2 characters	End idle length 3,5 characters

Address block

By query, the address block contains address of slave unit for which the query is determined (slave address). By response, the address block contains address of slave unit which sent the response.

Valid addresses for slave units are in the range 1–247. Address 0 is determined for a broadcast. Any frame with address 0 is considered as a broadcast where all slave units have to evaluate but none of them will answer to this broadcast.

Functional block

By query or broadcast, the functional block contains code of function (function code) which shall be performed. By response, the functional block contains copy of function code which slave unit received in query. Whether the most important bit of function code is set in the response, it means, that during performing of command with slave unit appeared error and is sent exception response. In such case data block contains code which closely specifies type of error (exception code).

Data block

Data block contains different information in dependence on required or performed function.

CRC block

CRC block contains 16-bit cyclic security code (CRC) which is used for integrity test of received frame.

2.3. Implemented functions

Code of function	Title	Standard	Note
1	Read coil status	Yes	
2	Read input status		
3	Read holding registers	Yes	
4	Force single coil	Yes	
10h	Preset multiple registers	Yes	
53h	Tunnel of Dresser protocol	No	Used by Wincor Evo software during communication with Modbus protocol

2.3.1. Tunnel of Dresser protocol

Function enables transmission of data parts of Dresser protocol using link layer of Modbus.
In the example is shown reading of real time of the device.

Query:

Byte No.	Meaning	Note	Example (hex)
1	Address		F8
2	Code of function		53
3	Length of data (LSB)	length of data including Type, Group and Port dest	03
4	Length of data (MSB)		00
5	Type	from head of Dresser	84
6	Group	from head of Dresser	6C
7	Port dest	from head of Dresser	00
8	Data	data layer Dresser	N=0
8+n			
9+n	CRC (LSB)		47
10+n	CRC (MSB)		85

Response:

Byte No.	Meaning	Note	Example (hex)
1	Address		F8
2	Code of function		53
3	Length of data (LSB)	length of data including Type, Group and Port dest	0B
4	Length of data (MSB)		00
5	Type	from head of Dresser	86
6	Group	from head of Dresser	6C
7	Port dest	from head of Dresser	00
8	Data	data layer Dresser	35
8+n			17
			10
			09
			11
			05
			0C
			3A
9+n	CRC (LSB)		6F
10+n	CRC (MSB)		7B

2.4. Map of internal registers

All quantities of the device (input, output, internal) and some parameters may be seen by help of parametrization in the frame of Modbus protocol in the space of internal register map (Modbus map). Quantities or parameters may be read using function 03h, some of them may be also written using function 10h. It is necessary to respect system of access rights for setting individual quantity or parameter during writing (password protection or switch protection), see 2.6.

Map of internal registers is defined by help of parametrization in the device, for definition is determined template. See chapter 3. Concrete map of Modbus is not mentioned in this manual because may be read from device using the application Wincor Evo.

Addresses of Modbus map are mentioned in the shape 4xxxx (4x references), with possible first address 40001.

Example:
For reading of register from address 40012 is necessary to use function 03h, as register address in communication frame to insert 11 (from address of Modbus map subtract 40001).

2.5. Map of discrete outputs

This map includes the possibility of controlling the device (reserved memory area for addresses <= 100) and reading or controlling the binary quantities of the device. It is possible to read the states of binary inputs, internal binary, binary outputs, setpoints and bit states of the instantaneous and sum status of the device. It is possible to set the status of binary outputs and internal binary.

Discrete outputs (coils) may be set using the function Setting of discrete output (Force single coil) with code 05h, reading the states of discrete outputs (coils) is possible with the function Read status of discrete output (Read coil status). Discrete outputs mediating device control (address <= 100) are write-only.

The device is controlled (address <= 100) by setting the appropriate discrete output. This starts the required action – e.g. resetting the archive, resetting the device. The assignment of addresses to the outputs controlling the device is described in chapter 2.5.1.

The map of discrete outputs with states of binary quantities of the device (addresses> 100) is defined by parameterization in the device. The template is used for definition. For more details, see chapter 3 Templates. For this reason, a specific Modbus map is not given in this manual, it can be read by Wincor Evo Software directly from the device.

The addresses in the Modbus map are given in the form 0xxxx (0x references), with the possible first address 101.

Note: as part of the implementation of the Modbus protocol in the device, binary quantities can be mirrored in the address space of discrete outputs (0x references, Modbus functions 1 and 5) and in the address space of internal registers (4x references, Modbus functions 3 and 10 hex).

2.5.1. Survey of discrete outputs map to control the device

Number of discrete outputs are mentioned in the text with the shape 0xxxx which correspond to Modbus specification for addressing of data items. Address of discrete outputs transmitted in communication frame may be calculated by subtraction 1 from the number of outputs in mentioned text.

In the following table is mentioned device survey of discrete outputs. Meaning of individual outputs is explained in next paragraphs.

Output No.	Output / command	Note
0001	Reset	Do not generate response
0017	Unlocking of user section	Necessary for writing of internal registers or discrete outputs which are protected by user password
0037	Deleting of data archive	
0038	Deleting of daily archive	
0039	Deleting of monthly archive	
0040	Deleting of extreme archive	
0041	Deleting of binary archive	
0042	Initialization of summary status	Will set bit of summary status according to actual state of bit
0043	Start-up of test device	

2.6. Access rights

To modify some quantities or parameters and set some discrete outputs (trigger actions), an access right to write in user level is required. It is necessary to know the password for full access and to have the user switch switched on. Parameterization can affect the behavior of the system controlling the right of access – only one of the elements can be used – a password or switch or the system can be completely disabled to allow modification to anyone. The password for full access for use in Modbus access must consist of a maximum of 6 digits (not letters).

Whether the password is activated it is necessary to write the password on the address defined in the Modbus map, further is necessary to connect discrete output 0017. The device will generate an exception response in case of parameter modification without access rights 02 – error address.

If user group passwords are enabled, you must use a password from the admin, maintenance, or user1 user groups, with the following ULONG encoding:

ID1 ID2 P1 P2 P3 P4 P5 P6

where ID1 and ID2 is the user ID as a two-digit number, i.e. user ID minus 801

P1..P6 is the password. If the user's password has less than 6 digits, it must be preceded by a zero number instead of missing characters when converting to LONG format.

Example: a user with ID 805 and password "1234" enters a password in the format ULONG = 04001234 dec.

The user password for use in Modbus must consist only of numbers, there must not be zeros at the beginning!!!

Only the user switch is respected, changes with the metrological switch are not allowed.

2.7. Error responses

Controlling unit expects the response for each query sent to the device. There are four cases during communication of type query / response:

- The device will send normal responses in case that will receive queries without communication error, and such queries may be processed.
- The device will not send any response in case that will not receive queries because of communication errors. Master unit may repeat query after set time.

- The device will not send any response in case that will receive query, but error of check sum is detected (CRC). Master unit may repeat query after set time.
- The device will send exception responses which informs about errors that appeared in case that the device will receive command without error which cannot be processed. (e.g. is required unknown function).

Error response contains two blocks which are different from normal response:

Functional block

In normal response, the device sends copy of function code received in query. All function codes have got the most important bit as 0 (value of the code is lower than 80h). In error response, the device sets the most important bit of function code as 1. It means that value of the code is about 80h higher than value in the query.

Whether the most important bit of function code is set to 1 then master unit will recognize error response and will decode the data block where is mentioned code of error.

Data block

Data blocks contain data generated by concrete functions in normal response. Data block contains code of error which appeared in error response.

2.7.1. Error codes generated by device

01 Unknown function (Illegal function)
Function with required code is not implemented.

02 Error address (Illegal data address)
Passed address of register or discreet output in query is not correct.

03 Error data (Illegal data value)
Value of data passed in query is not correct. Indicates incorrect length of data.

04 Critical error of slave unit (Slave device failure)
The device cannot generate answers because of critical error. Error is generated in case that command was correctly evaluated but could not be performed. For example: attempt to write to the register protected by password but without unlocking, attempt to write value out of range, attempt to unlock register using wrong password, not existing map of Modbus in the device or invalid parameters, attempt of parted reading of Mult register variable, attempt of reading / writing register without reading / writing rights for given register, attempt of writing to the group of registers from which at least one should be written individually…

06 Slave unit is busy (Slave device busy)
The device performs demanding operation and cannot process required function. Master unit must repeat query once the slave unit is ready. The device will generate error responses in case that reading or writing to EEPROM memory or to the periphery has started at the time when master unit was performing another operation.

2.8. Particularities of implementation

2.8.1. Service address

Device addresses may be set using parametrization. In some cases, may be found that device address is not known. From this reason was created address 248 on which the device always answers.

The device will always process and send responses in case that will receive query with address 248. Real set address of the device is sent in addressing block of response. Only one device may be connected to the bus bar using the service address.

2.8.2. Time of response

The device will send response after receiving last character of query to the time 2 to 2000 msec. Exact time of response depends on length of query, response and on performed function. Typical response time for reading values or parameters is 2 to 50 msec.

2.8.3. Calling the master system – sending a "CryOut message"

If the device is equipped with a modem (GSM or GPRS) and sending a message to the superior system is allowed, it will send the data described below to the preconfigured recipient in the event of a monitored event. The data is packed in the IFLAG protocol tunnel (see 2.3.1). After receiving a response from the master system, the device waits for queries from the master system. In case of idle on the communication channel, it hangs up the connection after the timeout expires and eventually switches off the modem.

Structure of sent data "Call from device, (service 87 hex)"

Sent data (sent by the device, forced by an event in the device)

Type	Group	Length of dec.	Telegram data
84 h	87 h	10 + 1 + 2 + 6	c + archive number + number of records + last date

length of the whole data structure (ushort)
structure version (now = 2)
GUID of this structure (16 bytes) = A4 80 6F 6A D5 64 41 19 B7 1B CE 96
  D 94 A4 40
  serves for correct identification (0. byte = 40 hex, last byte = A4 hex)
station name (17 char)
SIM card ID (10 char)
Modem ID (8 char)
protocol (char), 0 = IFLAG
address1 (ushort)
address2 (uchar)
GPRS signal strength (uchar)
diagnostics:
  number of GPRS connections (ulong)
  time of last GPRS connection (ulong)
  number of GPRS errors (ulong)
  time of last GPRS error (ulong)
  number of resets (ulong)
  last reset time (ulong)
  number of TCP data (ulong)
  number of all data (ulong)

In addition, the version of structure 2 contains the following items:
  serial number of the device (ulong)
  local IP address (ulong)
  time of last modem error (ulong)
  last modem error (uchar)
  modem battery capacity (ushort)
  modem battery voltage (ushort)
  FW modem version (33 bytes)

Response from the head-end system

Type	Group	Length of dec.	Telegram data
86 h	87 h	0a	none

2.9. GasNatural variant

The peculiarities of this variant concern the reading of archives. The requirement was to read the hourly records interspersed with the daily record, i.e. 24-hour records followed by one daily record. The read data of the daily and hourly archive must have an identical structure and must be defined in the template at identical addresses.

2.10. Modbus ENRON

The device is switched to Modbus ENRON mode after uploading a template with the selected "Enron" variant. Documentation for Modbus ENRON is not available, the main features implemented in the device are described below. In the following description, the convention of register addresses used so far is maintained – the address from the "holding registers" area 40001 means 0 on the bus. This convention is used for the clarity of the document even when creating templates, Modbus ENRON works with "addresses on the bus" by default.

2.10.1. 32-bit register area

From address 45001 (address 5000 on the bus) there are 32-bit registers. Thus, an increment of address by 1 means a 32-bit offset, e.g. a float variable is stored at address 47001 and another float variable follows at address 47002.

The division into the "ulong register" area (from address 45001) and the "float register" area (from address 47001), as described in the Modbus ENRON specification, can be achieved by placing the registers in a template, it has no significance for the device function.

2.10.2. Reading archives

Reading archives according to the "Modbus ENRON" specification is performed by reading from fixed addresses. The data (hourly) archive is read from address 40702 (701 on the bus), the daily archive is read from address 40703 (702). A modified FC:3 command is used to read the archives see. below. The registers are indexed 1..n. The length of the response depends on the specific parameters of the device.

The number of record registers can be configured in the template, for address 40702 the registers located in the template in the "Data archive" tab are sent, for address 40703 the registers located in the template in the "Daily archive" tab are sent. One archive record can be read with one query.

Query:

Byte number	Meaning	Remark	Example (hex)
1	Address		01
2	Function code		03
3	Address (MSB)	701 .. data archive 702 .. daily archive	02
4	Address (LSB)		BD
5	Index (MSB)	Record index (not number of registers !!!)	00
6	Index (LSB)		0A
7	CRC (LSB)		54
8	CRC (MSB)		51

Reply:

Byte number	Meaning	Remark	Example (hex)
1	Address		01
2	Function code		03
3	Address (MSB)	701 .. data archive 702 .. daily archive	
4	Address (LSB)		
5	Index (MSB)	Record index (not number of registers !!!)	
6	Index (LSB)		
7	CRC (LSB)		
8	CRC (MSB)		

When trying to read a record that does not exist, the device returns the response "Invalid response: Illegal data value".

3. Templates

There is option to modify Modbus map (map of internal registers) because of huge variability of measured or calculated quantities according to configuration of the device and customer´s needs. Template may be created for these purposes where is defined Modbus map according to requirements. Modbus map concerning concrete parameters is created during configuration of the device by help of template.

3.1. Creation of template

Template may be created using the application Wincor Evo via command „Setting / Modbus template ". Depending on the type of the entered quantity and according to the Modbus function (indicated in the header of the individual tabs), which is to be read, it is necessary to select the correct tab for the location of the quantity. Entering the command opens a window for selecting an existing template. If a new template name is entered, an empty template is created, into which individual elements can either be inserted with the "+" button on the table control bar or (if the cursor is at the end of the table) elements can be added by pressing the cursor key down.

Existing groups of Modbus map element types:

- actual values of quantities (e.g. pressure, temperature, volume, ...)
- parameters of quantities (e.g. conversion constant of counter/gas meter, ...)
- system quantities (e.g. time of device)
- system parameters (e.g. composition of gas)
- archive values of quantities (e.g. pressure, temperature, volume, statistical values, …) stored in data, binary, daily or monthly archive located in separate tabs of the template
- archive values stored in the status archive located in a separate tab of the template
- binary inputs, binary outputs, internal binary, setpoints, individual bits of the sum or instantaneous status of the device located in a separate tab of the template

First four types of elements are concentrated in the window of Modbus template under one bookmark, for archive quantities of individual archives are available separate windows marked with bookmarks with name of the archive.

In the windows of archive map is necessary (except for individual quantities and their addresses) to specify order of values and number of archive records. Order of values may be chosen „After structure „or „After quantity". First option (After structure) means that values stored at one time including date and time will be stored alongside each other. Second option (After quantity) means saving value of one quantity in archival time moments directly under each other, to the block separately from other quantities. Number of archive records means what archive depth will be available for reading, is not related to the possible number of records stored in the device. According to defined number of records will be created number of value groups with structure will be created according to template definition. Mentioned addresses in template are addresses of first record (youngest record, for versions up to 2008 the oldest record of the defined archive depth), according to chosen way of order are incremented for other records. Not existing records are transmitted as neutral values.

During creation of algorithm for reading of archive values is necessary to treat option of new archive record creation during reading process (at time between individual packets, nay during one packet) and thereby movement of history about one element in Modbus map. Elements with the youngest historic date (for versions up to 2008 with the oldest date) are located on the lowest address in Modbus map.

For first two types of elements and archive values where is necessary to determine concrete quantity may be identified quantity according to its address in visualization or according to type (analog, binary, counter, ...) and its consecutive number.

Whether elements of Modbus map are connected to quantities through the visualization address then change of this address in device parameters is blocked so it is not possible to modify the address. It will also prevent errors on Modbus mapping. By this way of identification, the quantities may be added, removed, and changed their consecutive number without impact to Modbus map and its template.

For the second way where the elements of Modbus mapping are connected to quantities through their type and consecutive number, Modbus map will be automatically updated by application Wincor Evo and the user will be notified that Modbus map was adjusted and that the map no longer matches the template from which it was

created. In this case visualization address may be used for different purposes.

It is necessary to fill up following columns in the table of elements:
- RW -„R“ means only reading of the value,„W“ only writing of the value,„RW“ reading and writing of the value
- Address – location of element in addressing register of Modbus (from 40001 and higher or 101 higher)
- Bit number – for the Modbus 3 function means the bit number of the register where the binary value is to be stored (i.e. if the binary value is "1" and bit number 3 is selected, the value of the output register is 0x0008) - for Modbus function 1 means the bit number of the device status word from which the value is taken
- Type – required data type – chapter 3.4 Data types used in the device
- Digit – multiplicative conversion constant
- Offset – additive conversion constant

Final value of Modbus will be as follows:
Digit * original value of the device + Offset.

3.2. Modbus map variant

By selecting "Variants" in the upper left corner of the template window, the extension can be activated according to the selected variant. In the current firmware version, in addition to the standard variants (marked "default"), it is possible to select the Modbus ENRON (marked "Enron") and GasNatural (marked "GasNatural") variants.

3.3. Saving of Modbus map to the device

Parameters of the device may be read using the application Wincor Evo into window of parameters. In the window of parameters may be added Modbus map from the menu „Add object of parametrization / Address of Modbus map“. This menu will be displayed after pressing the right button on the mouse and „Calculated measurands „must be selected. Application also requires selecting template according to which will be mapped elements of Modbus into address field. It is necessary to be aware that any changes cannot be made into already existing Modbus map, means to add, delete or change some elements. Change of map may be performed by removing and creating new mapping. It may be done manually or by restoring the revised template with relevant button.

Setting is finished by writing parameters to the device.

3.4. Data types used in the device

Several different data types are used to represent values and parameters in the device's internal registers:

BIT	**1 bit (1 byte)**
BYTE	An 8-bit unsigned value in the range 0 to 255. The value is stored in the lower byte of the 16-bit register. The higher byte of the registry is ignored, but it is recommended to always write it as 00h.
WRD	16-bit unsigned value in the range 0 to 65535
WRDrev	16-bit unsigned value in the range 0 to 65535, reverse order (2 bytes)
INT	16-bit signed value in the range -32768 to +32767
LNG	A 32-bit unsigned value stored in two consecutive 16-bit registers. The higher bits are stored in the register at address n, the lower bits are stored in the register at address n + 1. Range 0 to 4294967295. example: the number 318 is stored as 0h, 13Eh
LNGrev	unsigned long (4 byte), reverse order
LNFL	format consisting of two values, unsigned long (4 byte) - whole part of number, float (4 byte) - decimal part
LNFLrev	format consisting of two values, unsigned long (4 byte) - whole part of number, float (4 byte) - decimal part, reverse order

LNGTOT	unsigned long (4 byte), for converting double to long format, with 9-digit trimming (decimal)
LNGLNG	2 consecutive LNG formats (8 bytes)
LNG64	Unsigned long64 (8 byte)
FLT	32-bit single-precision floating-point value with simple accuracy, stored in two consecutive 16-bit registers example: 0.01 is stored as 3C23h, D70Ah
FLTrev	float (4 byte), reverse order
FLTswp	float (4 bytes), rearranged order (1st with 2nd and 3rd with 4th byte)
FLTswr	float (4 bytes), reversed and rearranged order (1st with 2nd and 3rd with 4th byte)
RFLT	float (4 bytes), the decimal remainder of the value in float format
DBL	64-bit high-precision floating-point value stored in four consecutive 16-bit registers example: the number 0.01 is stored as 3F84h, 7AE1h, 47AEh, 147Bh
DT1	date and time stored in 6 bytes in BCD format in three consecutive 16-bit registers example: 15:28:54 09.11.2005 is saved as 5428h, 1509h, 1105h
DT2	date and time stored in 6 bytes in BCD format in three consecutive 16-bit registers, the order is reversed than in the case of DT1 format
DT3	time and date in unsigned long format (number of seconds since 00:00:00 1.1.1970)
DT4	time and date in BCD format (6 registers) example: 7/16/2013 2:06:41 PM saved as 000Dh 0007h 0010h 000Eh 0006h 0029h
DT5	date and time stored in 6 bytes in BCD format in three consecutive 16-bit registers identical to DT1, in addition it transmits information about DST (System Daylight Saving Time) example: 7/16/2013 3:26:22 PM LČ is saved as 2226h D516h 0713h
DT6	date and time stored in 7 consecutive 16-bit registers example: Tuesday 7/16/2013 2:17:17 PM is saved as 0011h 0011h 000Eh 0010h 0007h 07DDh 0002h (day of the week 0..6, Sunday = 0)
DT7	date in MMDDYY format saved in float format (2 registers)
DT8	time in HHMMSS format stored in float format (2 registers)
DT9	time in MMDDYY and HHMMSS format stored in 2 x float format (4 registers)
BCD4	ulong number converted to BCD number, stored in two 16-bit registers
String	a string stored in the form of ASCII characters in 16-bit registers, if the length of the string does not reach the length of the defined field, the string is terminated by the character 0h, the following characters are insignificant. The following lengths are used: S3, S6, S8, S9, S12, S16, S32, S48, S80
StringR	a string stored in ASCII characters in 16-bit right-aligned registers, the following lengths are used: S16
Event	Status (ulong GasNatural) and time in DT3 format - especially for use in the GasNatural template
Fix	3 bytes whole part + 1 byte decimal part with sign
FixLNG	6 flats whole part and 2 flats decimal part without sign

Order of bytes is according to Modbus specification „Big-endian", means that MSB is on lower address than LSB, unless "reverse order" is given for each type.

3.5. Data type of individual quantities and parameters

In the following table are mentioned basic (native) data types of quantities and parameters. In these types of figure quantities or parameters of firmware in case that easy conversion is performed. Basic data type of quantity or parameter is chosen so it will comply to given use. It is possible to convert these basic types to others during definition of template, see 3.6 Options for data types conversion.

Type of element	Description	Data type
Quantities – actual values, archive values	Analog, Flow rate, Standard flow rate, Compressibility factor, Compressibility ratio, Compressibility, Combustion heat, Statistical values – minimum and maximum analogs	Float
	Actual tariff	Wrd
	Binary, Setpoint	Bit
	Counter, Error counter, Timer, Statistical values – maximum consumption of primary volume per hour or per day, Differential counter (base volume; hourly, daily, monthly), Status	Double
	Base volume counter, Error base volume counter, Corrected volume counter, Tariff (base volume) counter, Energy, Error energy, Statistical values – maximum consumption of base volume per hour or per day	Double
	Status	Ulong64
Parameter of quantities	Conversion constants – counter, base volume counter	Double
	Conversion constants – flow rate, Base flow rate	Float
	Serial number of the temperature, pressure sensor	Ulong
	Gas meter serial number	Ulong or String S16
	Setpoint limit, maximum gas meter flow, spare/ constant value of temperature, pressure and compressibility	Float
	Gas meter type, number of displayed decimal places of the operating counter	Byte
System quantities, archive values	System time, System daylight saving time (Summer time), Initial time. FTP, date and time of the archive record, date and time of reaching the extreme of the statistical quantity	DT1
	Actual, actual extended, summary, summary extended status, Password for full access	Ulong
	Instantaneous status, summary status	Ulong64

	Position of the last saved data archive record, position of the last saved daily archive record, number of saved data archive records, number of saved daily archive records. Position of the last saved monthly archive record, position of the last saved binary archive record, number of saved monthly archive records, number of saved binary archive records	Wrd
	Position of the last saved status archive record, number of saved status archive records	Wrd
System parameters	Identification of station	String S16
	SIM card PIN	String S9
	SIM card own number	String S16
	Telephone number for data calling or SMS	String S16 or S32
	Tariff archive period	String S3
	Modem initialization	String S32
	GPRS user name	String32 or String S48
	GPRS password	String S12 or S16 or S32
	Special modem initialization	String S32 or S80
	Type of device, FW version of device, CRC of parameters, Version of Modbus implementation, Device address, Period of data archive, Gas meter type, number of whole places primary (base) volume, IP port, http IP port, Modem type, Communication speed	Uint
	Serial numbers, IP address, IP address http	Ulong
	Gas day hour, Compressibility calculation method, Summer/Wintertime mode, offset from GMT, service switch meaning, Gas Day hour without Summertime, Send new Billing archive record, Device address2, Modem type, Communication speed, Communication protocol	Byte
	Base pressure, Base temperature, Relative density, Combustion heat, Composition of gas, Barometric pressure, Sea-level altitude	Float
	Billing archive switching time	DT1
Constant	Constant value	Ulong, Uint

Meaning of individual bit of actual and summary status:

Bit number	Description	Remark
0	E0 CRC program	
1	E1 CRC loader	
2	E2 CRC parameters	
3	E3 memory error	
4	E26 clock synchronization	
5	E5 setup archive is full	
6	E6 sensor change	
7	E7 sensor communication	
8	E8 sensor failure	
9	E9 low battery	
10	E10 compressibility table	
11	W0 sensor warning	
12	W30 bat m voltage	
13	W2 battery EEPROM	
14	W3 overcurrent	
15	E27 bat m kapac	
16	W5 external power failure	
17	E11 compressibility failure	
18	W29 batt m EEPROM	
19	E13 battery disconn	
20	E28 encoder fault	
21	W6 setup archive crowded	
22	E29 CRC metrolog	
23	E30 CRC table	
24	W7 tamper 1 fault	
25	W8 tamper 2 fault	
26	E31 FRAM error	
27	E32 FLASH error	
28	W31 comm failure	
29	W32 user cover	
30	M0 not configured	
31	M1 maintenance	
32	E14 P1 min limit	
33	E15 P1 max limit	
34	W9 P1 min threshold	

35	W10 P1 max threshold	
36	E16 P1 failure	
37	E17 T1 min limit	
38	E18 T1 max limit	
39	W11 T1 min threshold	
40	W12 T1 max threshold	
41	E19 T1 failure	
42	W13 Q1 min threshold	
43	W14 Q1 mx threshold	
44	W15 Qb1 min threshold	
45	W16 Qb1 myx threshold	
46	E33 bar press fail	
47	W36 volume diff	
48	E20 P2 min limit	
49	E21 P2 max limit	
50	W19 P2 min. limit	
51	W20 P2 max. limit	
52	E22 P2 error	
53	E23 T2 min limit	
54	E24 T2 max limit	
55	W21 T2 min. limit	
56	W22 T2 max. limit	
57	E25 T2 error	
58	W38 RTC setting	
59	W39 Prim volume setting	
60	W37 metrolog cover	
61	W33 ext power modem	
62	W34 metrol switch	
63	W35 user switch	

Meaning of individual bits in the status (archives):

Bit number	Description	Remark
0	General error	
1	General warning	
2	External power error	
3	Tamper is active	
4	Conversion error channel 1	
5	Transducer error channel 1	
6	Value out of range	
7	Value out of limits	
8	t1 error	
9	p1 error	
10	Q1 error	
11	Battery voltage below limit	
12	Setting archive full – W6	
13	Setting archive full – E5	
14	Device time synchronization error	
15	Device in maintenance mode	
16	Conversion error channel 2	
17	t2 error	
18	p2 error	
19	Q2 error	

3.6. Options for data types conversion

There is an option implemented to convert basic types of quantities or parameters to different types. It is necessary to pay attention that during performing of conversion may be decreased resolution of value or even degraded total value. Used conversion is checked during creating of template.

Legend: yes – conversion may be performed, no – conversion may not be performer

	Data types of firm-ware.	Bit	Byte	Uint	Ulong	Float	Double	DT1	String
Data types of Modbus	BIT	Yes	Yes	Yes	Yes	Yes	Yes	No	No
	BYTE	Yes	Yes	Yes	Yes	Yes	Yes	No	No
	WRD	Yes	Yes	Yes	Yes	Yes	Yes	No	No
	WRDrev	Yes	Yes	Yes	Yes	Yes	Yes	No	No
	INT	Yes	Yes	Yes	Yes	Yes	Yes	No	No
	LNG	Yes	Yes	Yes	Yes	Yes	Yes	No	No
	LNGrev	Yes	Yes	Yes	Yes	Yes	Yes	No	No
	LNFL	Yes	Yes	Yes	Yes	Yes	Yes	No	No
	LNFLrev	Yes	Yes	Yes	Yes	Yes	Yes	No	No
	LNGTOT	Yes	Yes	Yes	Yes	Yes	Yes	No	No
	LNGLNG	Yes	Yes	Yes	Yes	Yes	Yes	No	No
	FLT	Yes	Yes	Yes	Yes	Yes	Yes	No	No
	FLTrev	Yes	Yes	Yes	Yes	Yes	Yes	No	No
	FLTswp	Yes	Yes	Yes	Yes	Yes	Yes	No	No
	FLTswr	Yes	Yes	Yes	Yes	Yes	Yes	No	No
	DBL	Yes	Yes	Yes	Yes	Yes	Yes	No	No
	DT1	No	No	No	No	No	No	Yes	Yes
	DT2	No	No	No	No	No	No	Yes	Yes
	DT3	No	No	No	No	No	No	Yes	Yes
	DT4	No	No	No	No	No	No	Yes	Yes
	DT5	No	No	No	No	No	No	Yes	Yes
	DT6	No	No	No	No	No	No	Yes	Yes
	S3 .. S80	No	No	No	ulong -> string	No	No	No	Yes
	RFLT	Yes						No	No

4. Bibliography

[1] Modicon Modbus Protocol Reference Guide, Modicon Inc., Industrial Automation Systems, 1996 - Modbus.org/docs/PI_MBUS_300.pdf
[2] Wincor Evo Software

Dresser Utility Solutions GmbH

Hardeckstr. 2
76185 Karlsruhe
T: +49 (0)721 / 5981 - 0
info.karlsruhe@dresserutility.com

DRESSER®
UTILITY SOLUTIONS
www.dresserutility.com

810-2005-00 Modbus User Guide DUS.AGM.EN.AA
06.26