

**MODBUS parameter specification  
for  
ebm-papst series 84 / 112 / 150 / 200 Lite**

**V5.01 Lite**

Document: ext001157859.doc.doc

**Excerpt**

based on MODBUS parameter specification  
for ebm-papst series 84 / 112 / 150 / 200  
V5.00

**Copyright © ebm - papst Mulfingen, 2012 - 2016  
All rights reserved.**

**11th Mai 2016**

## Table of contents

1	Frame protocol .....	6
1.1	Structure of a byte .....	6
1.2	Communication process .....	6
1.2.1	Command from master .....	7
1.2.2	Response from fan .....	8
1.3	Commands .....	9
1.3.1	Read holding register .....	9
1.3.2	Read input register .....	10
1.3.3	Write single register .....	11
1.3.4	Diagnostics .....	12
1.3.5	Write multiple register .....	13
1.3.6	Commands with addressing via serial number .....	14
1.3.6.1	Read Holding Register Addressed By Serial No. ....	16
1.3.6.2	Read Input Register Addressed By Serial No. ....	17
1.3.6.3	Write Single Register Addressed By Serial No. ....	18
1.3.6.4	Write Multiple Register Addressed By Serial No. ....	19
1.3.6.5	Applications .....	20
1.3.7	Other commands .....	23
2	Holding Register .....	24
2.1	Overview .....	24
2.2	Reset .....	28
2.3	Default set value .....	29
2.4	Password .....	30
2.5	Control default setting .....	30
2.6	Control customer setting .....	31
2.7	Operating hours counter .....	31
2.8	Operating minutes counter .....	32
2.9	Fan address .....	32
2.10	Source set value .....	33
2.11	Preferred running direction .....	33
2.12	Save set value .....	34
2.13	Internal parameter set .....	34
2.14	Control mode .....	35
2.15	Control parameters .....	35
2.16	Maximum modulation level .....	37
2.17	Minimum modulation level .....	37
2.18	Enable motor stop .....	38
2.19	Set value (EEPROM) .....	38
2.20	Starting modulation level .....	39
2.21	Maximum permissible modulation level .....	39
2.22	Minimum permissible modulation level .....	40
2.23	Maximum speed .....	40
2.24	Maximum permissible speed .....	41
2.25	Ramp-up curve / ramp-down curve .....	41
2.26	Limit speed .....	42
2.27	Potentiometer characteristic .....	42
2.28	Control limitation .....	44
2.29	Maximum power .....	44
2.29.1	Maximum power .....	44
2.29.2	Maximum permitted power .....	45
2.29.3	Power derating .....	45

2.30	Maximum coil current .....	47
2.31	Limiting speed for running monitor .....	47
2.32	Interface settings .....	48
2.32.1	Transfer rate .....	48
2.32.2	Parity configuration .....	48
2.32.3	Procedure for changing the interface settings .....	49
2.32.4	Interface emergency function: .....	50
2.33	Shedding function .....	51
2.34	Relay dropout delay .....	52
2.35	Fail safe function on/off .....	52
2.36	Fail safe function set value .....	53
2.37	Fail safe function time lag .....	54
2.38	Fail safe function running direction .....	54
2.39	Potentiometer characteristic limiting value for cable break .....	55
2.40	Sensor .....	56
2.41	Customer data .....	57
2.42	Error history .....	57
2.43	Reference value of DC-link voltage .....	59
2.44	Reference value of DC-link current .....	59
2.45	Production data .....	60
2.45.1	Fan serial number and production date .....	60
2.45.2	Fan type .....	61
3	Input Register .....	62
3.1	Overview .....	62
3.2	Identification .....	63
3.3	Maximum number of bytes .....	63
3.4	Software name of bus controller .....	64
3.5	Software version of bus controller .....	64
3.6	Software name of commutation controller .....	64
3.7	Software version of commutation controller .....	64
3.8	Actual speed .....	65
3.9	Motor status .....	65
3.10	Warning .....	66
3.11	DC-link voltage .....	66
3.12	DC-link current .....	67
3.13	Module temperature .....	67
3.14	Motor temperature .....	67
3.15	Interior electronics temperature .....	67
3.16	Current direction of rotation .....	68
3.17	Current modulation level .....	68
3.18	Current set value .....	68
3.19	Sensor actual values .....	69
3.20	Current parameter set .....	69
3.21	Current power .....	70

## Preface

This document details the specifications for the MODBUS parameters of the "series 84 / 112 / 150 / 200 Lite" unit from ebm-papst.

*This excerpt differs from the complete document in that the parameters for internal use by ebm-papst are not described here. These parameters are marked by the word "reserved".*

Knowledge of general MODBUS specifications is required:

- MODBUS over Serial Line Specification & Implementation guide V1.0
- MODBUS Application Protocol Specification V1.1

These documents are available on the Internet at MODBUS.org.

The general MODBUS specifications form the basis for this document and are valid in full with the exception of the restrictions described in this document.

## Change history

Version	Changes
5.00	Document annex
5.01	<p>Added: the following holding registers:</p> <p>D160 - D161 Minimum sensor value                      D162 - D163 Maximum sensor value                      D164 - D169 Sensor unit</p> <p>Added: the following input registers</p> <p>D01B Sensor actual value                      D023 Sensor actual value 1</p> <p>Change: Write authorisation level for holding registers:</p> <p>D14D Motor temperature at start of derating (customer)                      D14E Motor temperature at end of derating (customer)                      D150 Shedding function (customer)                      D151 Max. starting modulation level (ebm-papst)                      D152 Number of start attempts (customer)                      D15B Emergency operation running direction (customer)</p>

# 1 Frame protocol

Data are transferred using the MODBUS protocol defined in these specifications exclusively in an environment defined as a master/slave system. The orderly progression of data is defined by the master. A slave is required to respond to its command prompt. For this reason, it is important to ensure that no slave address is assigned more than once when constructing a system.

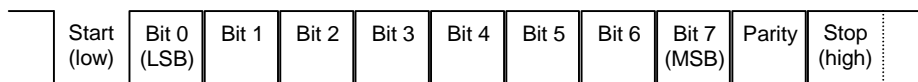
A twisted pair wire with RS485 standard should preferably be used.

Only RTU transmission mode is supported (see MODBUS over Serial Line Specification & Implementation guide V1.0, chapter 2.5.1)

ASCII transmission mode is not supported!

## 1.1 Structure of a byte

According to the MODBUS over Serial Line Specification & Implementation guide V1.0, a byte has the following structure:

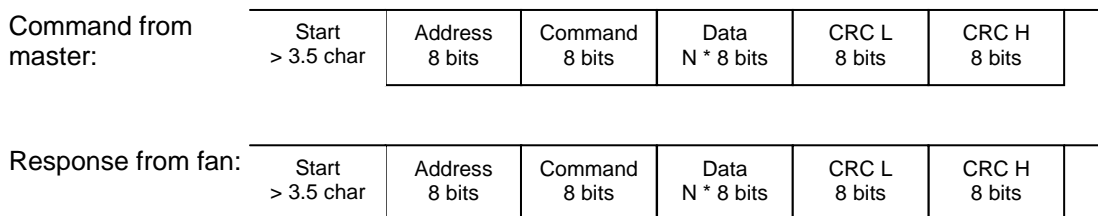


The definition of the parity bit ("Even", "Odd", "None") can be set with the parameter "Parity configuration" (see 2.32.2).

The transmission rate is variable and can be set with the "transmission rate" parameter (see 2.32.1).

## 1.2 Communication process

The MODBUS over Serial Line Specification & Implementation guide V1.0 defines the following framework for the transmission protocol:



In contrast to the general specifications, the maximum telegram length is 23 bytes!

### 1.2.1 Command from master

A master device is, for example, a PC or a control device.

Initial synchronisation:

A transmission pause of at least 3.5 bytes is used for initial synchronisation. The following byte is then interpreted as the first byte of a frame (i.e. address). The pause between the individual bytes of a frame may be a maximum of 1.5 bytes.

Address:

The address field has a size of 8 bits. Address values from 1 to 247 are permitted. The address 0 is reserved for broadcast commands (i.e. commands to all fans in the network).

Command:

The following commands from the "MODBUS Application Protocol Specification V1.1" are supported:

Code	Command
0x03	Read holding register
0x04	Read input register
0x06	Write single register
0x08	Diagnostics
0x10	Write multiple register

Other commands are not supported.

The following additional commands are defined by ebm-papst:

Code	Command
0x43	Read Holding Register Addressed By Serial No.
0x44	Read Input Register Addressed By Serial No.
0x46	Write Single Register Addressed By Serial No.
0x50	Write Multiple Register Addressed By Serial No.

Data:

Depending on the command concerned, the number of data bytes and their meaning may differ. Please refer to 1.3. Commands

### CRC L / CRC H

A CRC checksum is generated for the complete telegram.  
The polynomial for defining the checksum is  $1 + x^2 + x^{15} + x^{16}$  (i.e. XOR operation on 0xA001).  
The initial value is 0xFFFF.

The low byte of the checksum is transmitted first, then the high byte.

More detailed information about calculating the checksum can be found in the "MODBUS over Serial Line Specification & Implementation guide V1.0".

### **1.2.2 Response from fan**

A fan will only respond if

- it receives a message sent to its own address;  
for the broadcast address, no response is sent.
- the telegram length is at most 23 bytes.
- the correct number of data bytes was sent so that the telegram can be interpreted.
- the checksum was correctly recognised.

#### Initial synchronisation:

After the command from the master has been completed, the fan will wait for *at least* one transmission pause of 3.5 bytes. Depending on the command and on the processing time, the pause may be much longer (until the fan has received all the data it has requested)

#### Address:

The address is repeated by the command from the master (i.e. its own fan address)

#### Command:

If the command can be processed, the command code will be repeated.

If the command cannot be processed, the fan will respond with an exception.

Here, the MSB is set to command.

The command byte is then, for example, 0x83 for the command "Read holding register (0x03)".

#### Data:

Depending on the command concerned, the number of data bytes and their meaning may differ.  
Please refer to 1.3 Commands.

### CRC L / CRC H

A CRC checksum is generated for the complete telegram.  
The way the checksum is generated is no different from the procedure described above for the command from the master.



## 1.3 Commands

### 1.3.1 Read holding register

Command code: 0x03

This command is used to read out the content of a number of holding registers. Holding registers are parameters for which both read and write access are possible.

#### Command from master:

4 data bytes are transmitted:

- 1st MSB holding register address
- 1st LSB holding register address
- Number of MSB addresses to be read
- Number of LSB addresses to be read

The holding registers are described below.

#### Response from fan:

The following data bytes are transmitted:

- Byte count (number of addresses to be read \* 2)
- Data in 1st MSB holding register
- Data in 1st LSB holding register

Optional:

- Data from the following holding registers (0 to n)

#### Exception codes:

In case of error, only one data byte (the exception code) will be transmitted

Exception codes:

- 0x02: Permissible range of the holding registers 0xD000 to 0xD3FF exceeded  
0x03: Maximum telegram length for response (23 bytes) exceeded,  
i.e. either more than 9 holding registers or 0 holding registers were requested.

### 1.3.2 Read input register

Command code: 0x04

This command is used to read out the content from multiple input registers. Input registers are parameters for which only read access is possible.

#### Command from master:

4 data bytes are transmitted:

- 1st MSB input register address
- 1st LSB input register address
- Number of MSB addresses to be read
- Number of LSB addresses to be read

The input registers are described below.

#### Response from fan:

The following data bytes are transmitted:

- Byte count (number of addresses to be read \* 2)
- Data in 1st MSB holding register
- Data in 1st LSB holding register

Optional:

- Data from the following input registers (0 to n)

#### Exception codes:

In case of error, only one data byte (the exception code) will be transmitted

Exception codes:

0x02: Permissible range of the input registers 0xD000 to 0xD0FF exceeded

0x03: Maximum telegram length for response (23 bytes) exceeded,  
i.e. either more than 9 input registers or 0 input registers were requested.

### 1.3.3 Write single register

Command code: 0x06

This command is used to write the content of *one* holding register.

#### Command from master:

4 data bytes are transmitted:

- MSB holding register address
- LSB holding register address
- MSB data to be written
- LSB data to be written

The holding registers are described below.

#### Response from fan:

4 data bytes are transmitted:

- MSB holding register address
- LSB holding register address
- MSB data to be written
- LSB data to be written

#### Exception codes:

In case of error, only one data byte (the exception code) will be transmitted

Exception codes:

0x02: Permissible range of the holding registers 0xD000 to 0xD3FF exceeded

0x04: There is no write access in this authorisation level (password).

### 1.3.4 Diagnostics

Command code: 0x08

This command is used to check the MODBUS function

#### Command from master:

The following data bytes are transmitted:

- MSB subfunction code
- LSB subfunction code
- 1 - 17 data bytes

Only subfunction code 0000 is supported!

#### Response from fan:

The following data bytes are transmitted:

- MSB subfunction code
- LSB subfunction code
- 1 - 17 data bytes

#### Exception codes:

In case of error, only one data byte (the exception code) will be transmitted

Exception codes:

0x01: Subfunction code not supported ( $\neq$  0000)

### 1.3.5 Write multiple register

Command code: 0x10

This command is used to write the content of *several* holding registers.

#### Command from master:

The following data bytes are transmitted:

- MSB holding register address
- LSB holding register address
- Number of MSB addresses to be written
- Number of LSB addresses to be written
- Byte count (number of addresses to be written \* 2)
- Data to be written in 1st MSB holding register
- Data to be written in 1st LSB holding register

Optional:

- Data to be written to the following holding registers (0 to n)

The holding registers are described below.

#### Response from fan:

4 data bytes are transmitted:

- MSB holding register address
- LSB holding register address
- Number of MSB addresses to be written
- Number of LSB addresses to be written

#### Exception codes:

In case of error, only one data byte (the exception code) will be transmitted

Exception codes:

- 0x02: Permissible range of the holding registers 0xD000 to 0xD3FF exceeded
- 0x03: - The number of maximum possible registers has been exceeded,  
i.e. either more than 123 holding register data or 0 holding register data were defined.  
- Byte count  $\neq 2 * \text{number of registers}$   
- Number of data bytes  $\neq \text{byte count}$
- 0x04: There is no write access in this authorisation level (password).

### 1.3.6 Commands with addressing via serial number

These commands also use the fan serial numbers for addressing:  
 An identifier for the fan serial number (6 bytes) is transmitted in the data section of the commands.  
 Only the fan that has the MODBUS address and serial number given in the command will react and respond.

The command with addressing using the serial number is made up of the standard command, to which a serial number identifier is added after the command byte.

*Serial number format:*

ebm-papst assigns an individual serial number to each fan. The part that our customers can extract/read consists of the production date and the consecutive number. Addressing can be easily done using this data. In the unlikely event that you encounter any problem doing so, please let us know.

Format: YYWW00XXXX

- YY : Production year
- WW : Calendar week of production
- 00 : Fixed value 00
- XXXX : Sequential number

The first 4 digits contain the production date (year/calendar week).  
 With the beginning of each production week, the number XXXX starts counting up from zero and is increased by 1 for each fan. Each character can represent values from 0-9 and from A-Z. The maximum number of characters that can be encoded for each digit is thus 36, i.e.  $36^4 = 1,679,616$  devices/week.

Example:  
 ebm-papst serial number: 09230012GY  
 Year of production: 09 (2009)  
 Calendar week: 23 (23)  
 Sequential no.: 12GY (49525)

Year of production and calendar week are each encoded as hex values in the command.  
 For the series number, each character is encoded as an ASCII value.

In the commands with addressing via the serial number, the serial number must be given in the following manner:

Byte no.	Meaning	Example	Corresponds to holding register
Byte 0:	[Address]		
Byte 1:	[Command code]		
Byte 2:	Production year	0x09 (-> 2009)	D1A4 MSB
Byte 3:	Calendar week	0x17 (-> 23)	D1A4 LSB
Byte 4:	Sequential number 1st character	0x31 (-> 1)	D1A3 MSB
Byte 5:	Sequential number 2nd character	0x32 (-> 2)	D1A3 LSB
Byte 6:	Sequential number 3rd character	0x47 (-> G)	D1A2 MSB
Byte 7:	Sequential number 4th character	0x59 (-> Y)	D1A2 LSB

The order depends on the sequence in which the serial number is stored in the holding registers D1A2 - D1A4 (see 2.45.1)!

*Broadcast addressing*

If the serial number identifier value 0x00 is transmitted in one or more bytes, the corresponding part of the serial number will not be checked by the fan. All fans with the remaining part of the serial number identifier will then respond.

If all bytes are transmitted with the value 0x00, all fans will respond.

In the response, the fans' own addresses and serial numbers will be given in place of the broadcast address / serial number 0x00.

*Options for identifying the serial number:*

- Printed on the fan type plate
- Read the serial number from the holding register D1A2 - D1A4 (see 2.45.1)
- Broadcast addressing: The connected fan responds with its serial number

### 1.3.6.1 Read Holding Register Addressed By Serial No.

Command code: 0x43

This command is used to read out the content of a number of holding registers. Holding registers are parameters for which both read and write access are possible.

In contrast to "MODBUS over Serial Line Specification & Implementation Guide V1.0", the fan also responds to a broadcast command (MODBUS address = 0).

#### Command from master:

10 data bytes are transmitted:

- Serial No. byte 1
- Serial No. byte 2
- Serial No. byte 3
- Serial No. byte 4
- Serial No. byte 5
- Serial No. byte 6
- 1st MSB holding register address
- 1st LSB holding register address
- Number of MSB addresses to be read
- Number of LSB addresses to be read

The holding registers are described below.

#### Response from fan:

The following data bytes are transmitted:

- Serial No. byte 1
- Serial No. byte 2
- Serial No. byte 3
- Serial No. byte 4
- Serial No. byte 5
- Serial No. byte 6
- Byte count (number of addresses to be read \* 2)
- Data in 1st MSB holding register
- Data in 1st LSB holding register

Optional:

- Data from the following holding registers (0 to n)

#### Exception codes:

In case of error, only one data byte (the exception code) will be transmitted

Exception codes:

- 0x02: Permissible range of the holding registers 0xD000 to 0xD3FF exceeded  
0x03: Maximum telegram length for response (23 bytes) exceeded,  
i.e. either more than 6 holding registers or 0 holding registers were requested.



### 1.3.6.2 Read Input Register Addressed By Serial No.

Command code: 0x44

This command is used to read out the content from multiple input registers. Input registers are parameters for which only read access is possible.

In contrast to "MODBUS over Serial Line Specification & Implementation Guide V1.0", the fan also responds to a broadcast command (MODBUS address = 0).

#### Command from master:

10 data bytes are transmitted:

- Serial No. byte 1
- Serial No. byte 2
- Serial No. byte 3
- Serial No. byte 4
- Serial No. byte 5
- Serial No. byte 6
- 1st MSB input register address
- 1st LSB input register address
- Number of MSB addresses to be read
- Number of LSB addresses to be read

The input registers are described below.

#### Response from fan:

The following data bytes are transmitted:

- Serial No. byte 1
- Serial No. byte 2
- Serial No. byte 3
- Serial No. byte 4
- Serial No. byte 5
- Serial No. byte 6
- Byte count (number of addresses to be read \* 2)
- Data in 1st MSB holding register
- Data in 1st LSB holding register

Optional:

- Data from the following input registers (0 to n)

#### Exception codes:

In case of error, only one data byte (the exception code) will be transmitted

Exception codes:

0x02: Permissible range of the input registers 0xD000 to 0xD0FF exceeded.

0x03: Maximum telegram length for response (23 bytes) exceeded,  
i.e. either more than 6 input registers or 0 input registers were requested.

### 1.3.6.3 Write Single Register Addressed By Serial No.

Command code: 0x46

This command is used to write the content of *one* holding register.

In contrast to "MODBUS over Serial Line Specification & Implementation Guide V1.0", the fan also responds to a broadcast command (MODBUS address = 0), provided that no broadcast identifier (0) is used in the serial number. Because no two different fans can have the same serial number, it is not possible for more than one fan to respond.

#### Command from master:

10 data bytes are transmitted:

- Serial No. byte 1
- Serial No. byte 2
- Serial No. byte 3
- Serial No. byte 4
- Serial No. byte 5
- Serial No. byte 6
- MSB holding register address
- LSB holding register address
- MSB data to be written
- LSB data to be written

The holding registers are described below.

#### Response from fan:

10 data bytes are transmitted:

- Serial No. byte 1
- Serial No. byte 2
- Serial No. byte 3
- Serial No. byte 4
- Serial No. byte 5
- Serial No. byte 6
- MSB holding register address
- LSB holding register address
- MSB data to be written
- LSB data to be written

The command from the master is repeated in the response.

#### Exception codes:

In case of error, only one data byte (the exception code) will be transmitted

Exception codes:

0x02: Permissible range of the holding registers 0xD000 to 0xD3FF exceeded.

0x04: There is no write access in this authorisation level (password).

### 1.3.6.4 Write Multiple Register Addressed By Serial No.

Command code: 0x50

This command is used to write the content of *several* holding registers.

#### Command from master:

The following data bytes are transmitted:

- Serial No. byte 1
- Serial No. byte 2
- Serial No. byte 3
- Serial No. byte 4
- Serial No. byte 5
- Serial No. byte 6
- MSB holding register address
- LSB holding register address
- Number of MSB addresses to be written
- Number of LSB addresses to be written
- Byte count (number of addresses to be written \* 2)
- Data to be written in 1st MSB holding register
- Data to be written in 1st LSB holding register

Optional:

- Data to be written to the following holding registers (0 to n)

The holding registers are described below.

#### Response from fan:

10 data bytes are transmitted:

- Serial No. byte 1
- Serial No. byte 2
- Serial No. byte 3
- Serial No. byte 4
- Serial No. byte 5
- Serial No. byte 6
- MSB holding register address
- LSB holding register address
- Number of MSB addresses to be written
- Number of LSB addresses to be written

#### Exception codes:

In case of error, only one data byte (the exception code) will be transmitted

Exception codes:

- 0x02: Permissible range of the holding registers 0xD000 to 0xD3FF exceeded.
- 0x03: - The number of maximum possible registers has been exceeded,  
 i.e. either more than 123 holding register data or 0 holding register data were defined.  
 - Byte count  $\neq 2 * \text{number of registers}$   
 - Number of data bytes  $\neq \text{byte count}$
- 0x04: There is no write access in this authorisation level (password).

**1.3.6.5 Applications**

**Initialising a system**

All fans have the same address 1 when they leave the factory. So that each fan can be addressed individually, this address must be modified in such a way that each fan is assigned its own separate address when a system comprising several fans is being configured.

Conventional method (command 0x06 Write Single Register):

1. Switch on the first fan (all others remain switched off)
2. Change the address via command 0x06 Write Single Register
3. Switch on the next fan
4. Repeat steps 2 and 3 until all fans have an individual address.

Procedure for known serial numbers (e.g. printed on type plate)

1. Switch on all fans
2. Change the addresses using command 0x46 Write Single Register Addressed By Serial No.

Procedure with unknown serial numbers:

1. Switch on all fans
2. Identification of the serial numbers using command 0x43 Read Holding Register Addressed By Serial No. or 0x44 Read Input Register Addressed By Serial No. in broadcast address with collision detection
3. When a serial number has been identified:  
Change the MODBUS address via command 0x46 Write Single Register Addressed By Serial No.

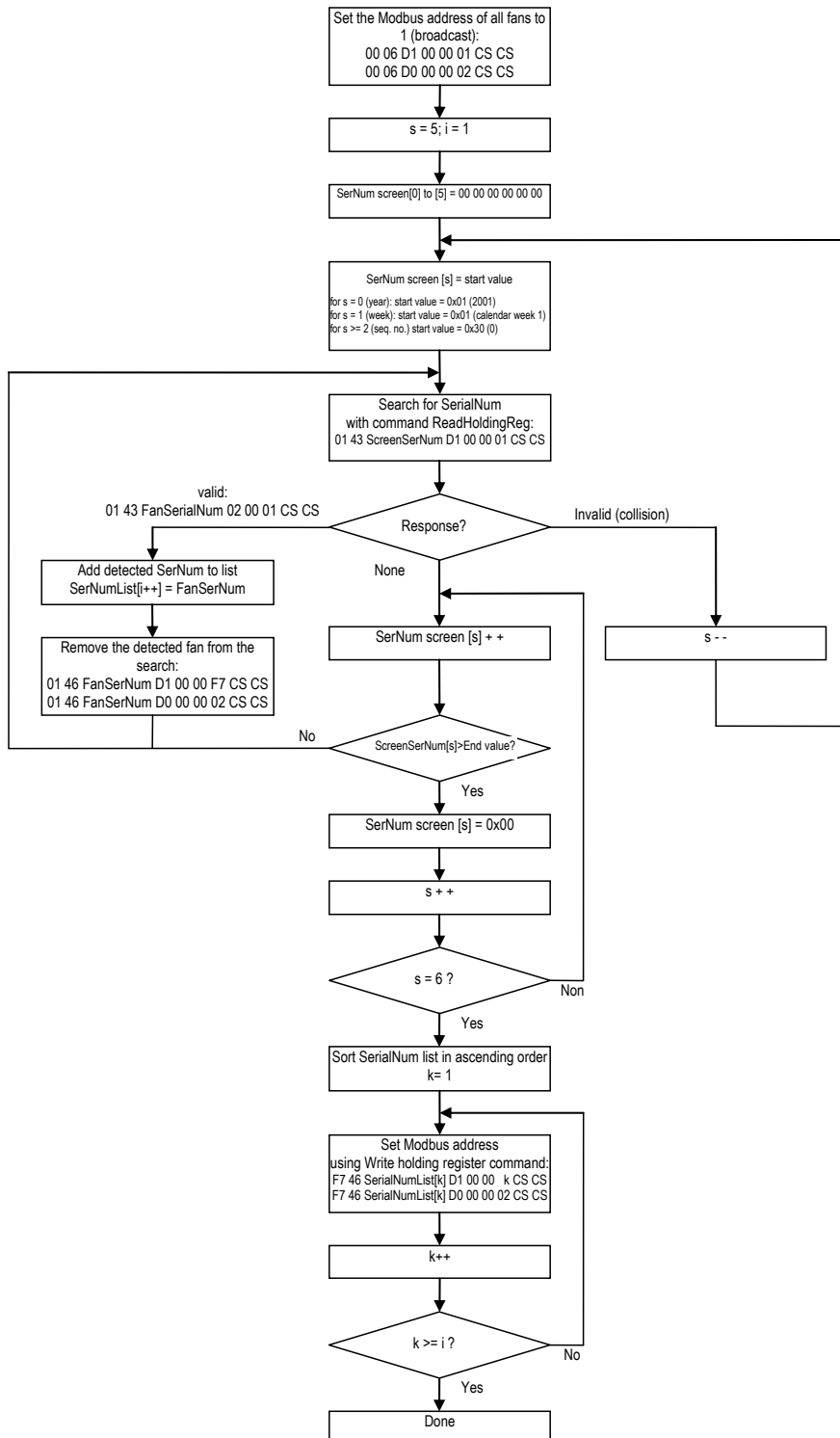
The advantage over the conventional procedure is that all the fans can be switched on at once during the initialisation of a system. Despite the same address 1, each fan can be addressed individually. Each fan can be assigned an individual address by the master device (e.g. PC) without the need to perform switching actions on the system.

*Implementation in the master device (e.g. PC):*

For known serial numbers, we recommend the following procedure for implementation in a master device:

1. The master device is furnished with all serial numbers via manual entry
2. Each fan is assigned a MODBUS address via the serial number.  
For this purpose, 2 broadcast commands are required per fan:
  - a. Set the holding register fan address (D100):  
00 46 SNr SNr SNr SNr SNr SNr SNr D1 00 00 Adr CS CS  
(CS = CRC checksum)
  - b. Set the holding register reset (D000) to 0x02 Adopt parameters:  
00 46 SNr SNr SNr SNr SNr SNr SNr D0 00 00 02 CS CS  
SNr = determined serial number of the fan; Adr = assigned MODBUS address

For unknown serial numbers, we recommend the following procedure for implementation in a master device:



## Function:

*When the system is being set up, the fans have to be arranged in ascending order by serial number. This is the prerequisite for ensuring that the automatically assigned MODBUS addresses can easily be correlated with the fans in the system.*

At the beginning, all fans are set to MODBUS address 0x01.

For this purpose, 2 broadcast commands are required:

- Set the holding register fan address (D100) to 0x01: 00 06 D1 00 00 01 CS CS  
(CS = CRC checksum)
- Set the holding register reset (D000) to 0x02 Adopt parameters: 00 06 D0 00 00 02 CS CS

All of the serial numbers in the system are determined in a loop:

The screen for serial number addressing is set initially to broadcast addressing (00 00 00 00 00). The last byte is set to the start value of 0x30 (0).

The SerNr screen is thus 00 00 00 00 00 30.

This screen searches for fans that have the value 0x30 (0) in the last position of the serial number.

Any command for reading a holding register or input register can be used, e.g. Read holding register fan address: 01 43 00 00 00 00 00 30 D1 00 00 01 CS CS

There are several possibilities for the response:

- There is a unique response.  
The connected fan responds with its serial number in the address field (SerNrVent).  
This serial number is stored in a list.  
Then, the fan is blocked from further queries by setting the MODBUS address to 0xF7 (247).  
The same serial number then has to be queried again, as the possibility cannot be eliminated that another fan was also addressed, but did not respond because it already detected the beginning of the other fan's response due to run time differences.
- There is an invalid response due to overlapping responses from multiple fans.  
Then the serial number screen needs to be limited further by also setting the next-to-last byte to the start value of 0x30 (0). The next query now gets a response only from fans that have the value 0x30 (0) in the last two digits of the serial number.
- There is no response.  
In this case, all serial numbers that have the value 0x30 in the last position can be eliminated. The last byte of the screen is increased to the value 0x31 (1). On the next query, all fans that have the value 0x31 (1) in the last position of their serial numbers will respond.

The loop is now continued until all serial numbers have been queried.

- If a valid response is given, the serial number of the fan is stored in the list.
- In case of an invalid response, the serial number range is restricted further by masking an additional byte, beginning with the start value. For the last 4 bytes, the start value is 0x30 (0). The first two bytes have the start value 0x01, as the year (2001) and calendar week (week 1) are encoded here.
- If there is no response, the respective position is increased by 1 until the end value is reached.  
For the last 4 bytes, the end value is 0x5A (Z). For the first byte (year), the end value is 0x63 (2099); for the second byte (calendar week), the end value is 0x35 (week 53).  
When the end value is reached, the respective position is again addressed via broadcast (0x00) and the next position is increased by 1. When addressing for the last byte reaches the end value 0x5A (Z), all serial numbers have been checked and serial number querying can be concluded.

Afterwards, all serial numbers found in the master are sorted in ascending order.

Each fan is assigned a MODBUS address via the serial number.

For this purpose, 2 commands are required per fan:

- Set the holding register fan address (D100): F7 46 SNr SNr SNr SNr SNr SNr D1 00 00 Adr CS CS
- Set the holding register reset (D000) to 0x02 Adopt parameters:  
F7 46 SNr SNr SNr SNr SNr SNr D0 00 00 02 CS CS  
SNr = determined serial number of the fan; Adr = assigned MODBUS address

### ***Extension of the address space***

Up to 247 fans can be addressed with standard MODBUS commands.

Commands with addressing via serial number offer the possibility of addressing an unlimited number of ebmpapst fans using the MODBUS interface. In this case, the address of the fan is not just made up on one byte, but rather of 7 bytes (MODBUS address + 6 byte serial number). This address is defined in the factory and cannot be changed.

For the purpose of addressing, the serial number must be known in the master device (e.g. PC) as a scan of all serial numbers is not possible due to the large numbers involved. To identify the serial number, see above (Initialising a system).

Only commands with addressing via the serial number can be used with an extended address space. When standard commands are used, conflicts are inevitable as several fans will have the same MODBUS address.

### ***Identifying an unknown MODBUS address***

- Only one fan may be connected to the bus
- Command 0x43 Read Holding Register Addressed By Serial No. or 0x44 Read Input Register Addressed By Serial No. in broadcast addressing:  
MODBUS address and serial number identification are specified using 0x00
- The connected fan responds with its MODBUS address and serial number

### **1.3.7 Other commands**

All other commands are not supported.

A command is always responded to with exception code 0x01.

## 2 Holding Register

### 2.1 Overview

The holding registers are stored in the RAM and in the EEPROM of the fan.

The following ranges are defined:

Address	Memory
D000 to D0FF	RAM
D100 to D3FF	EEPROM

The following list gives an overview of all parameters.

Apart from the MODBUS address and the designation, it shows which authorisation level is required to write a parameter, and the address of the memory space for default setting and customer setting (if applicable).

The function of the parameters is described in the following chapters

MODBUS address	Designation	Write ebm-papst	Write customer	Write end customer	Default address	Cust. setting address
D000	Reset	X	X	X	-	-
D001	Default set value	X	X	X	-	-
D002 D003 D004	Password	X	X	X	-	-
D005	Control default setting	X	X <sup>*)</sup>	-	-	-
D006	Control customer setting	X	X	X <sup>*)</sup>	-	-
D007	No function	X	-	-	-	-
D008	Reserved	X	-	-	-	-
D009	Operating hours counter	X	-	-	-	-
D00A	Operating minutes counter	X	-	-	-	-
D00B	Reserved	-	-	-	-	-
D00C - D0FF	Vacant	-	-	-	-	-
D100	Fan address	X	X	X	D280	D200
D101	Source set value	X	X	X	D281	D201
D102	Preferred running direction	X	X	X	D282	D202
D103	Save set value	X	X	X	D283	D203
D104	No function	X	X	X	D284	D204
D105	Internal parameter set	X	X	X	D285	D205
D106	Control mode (parameter set 1)	X	X	X	D286	D206
D107	Control mode (parameter set 2)	X	X	X	D287	D207
D108	No function	X	X	X	D288	D208
D109	No function	X	X	X	D289	D209
D10A	P-factor (parameter set 1)	X	X	X	D28A	D20A
D10B	P-factor (parameter set 2)	X	X	X	D28B	D20B
D10C	I-factor (parameter set 1)	X	X	X	D28C	D20C
D10D	I-factor (parameter set 2)	X	X	X	D28D	D20D
D10E	Maximum modulation level (parameter set 1)	X	X	-	D28E	D20E



MODBUS address	Designation	Write ebm-papst	Write customer	Write end customer	Default address	Cust. setting address
D10F	Maximum modulation level (parameter set 2)	X	X	-	D28F	D20F
D110	Minimum modulation level (parameter set 1)	X	X	X	D290	D210
D111	Minimum modulation level (parameter set 2)	X	X	X	D291	D211
D112	Motor stop enable (parameter set 1)	X	X	X	D292	D212
D113	Motor stop enable (parameter set 2)	X	X	X	D293	D213
D114	Set value (parameter set 1)	X	X	X	D294	D214
D115	Set value (parameter set 2)	X	X	X	D295	D215
D116	Starting modulation level	X	-	-	D296	-
D117	Max. permissible modulation level	X	-	-	D297	-
D118	Min. permissible modulation level	X	-	-	D298	-
D119	Maximum speed	X	X	-	D299	D219
D11A	Max. permissible speed	X	-	-	D29A	-
D11B	Reserved	X	-	-	D29B	-
D11C	Reserved	X	-	-	D29C	-
D11D	Vacant	X	X	X	D29D	D21D
D11E	No function	X	-	-	D29E	-
D11F	Ramp-up time	X	X	X	D29F	D21F
D120	Ramp-down time	X	X	X	D2A0	D220
D121	Reserved	X	-	-	D2A1	-
D122	Reserved	X	-	-	D2A2	-
D123	Reserved	X	-	-	D2A3	-
D124	Reserved	X	-	-	D2A4	-
D125	Reserved	X	-	-	D2A5	-
D126	Reserved	X	-	-	D2A6	-
D127	Reserved	X	-	-	D2A7	-
D128	Limit speed	X	-	-	D2A8	-
D129	Reserved	X	-	-	D2A9	-
D12A	Potentiometer characteristic point 1 X-coordinate (par. 1)	X	X	X	D2AA	D22A
D12B	Potentiometer characteristic point 1 Y-coordinate (par. 1)	X	X	X	D2AB	D22B
D12C	Potentiometer characteristic point 2 X-coordinate (par. 1)	X	X	X	D2AC	D22C
D12D	Potentiometer characteristic point 2 Y-coordinate (par. 1)	X	X	X	D2AD	D22D
D12E	No function	X	X	X	D2AE	D22E
D12F	Control limitation	X	-	-	D2AF	-
D130	No function	X	X	-	D2B0	D230
D131	Reserved	X	-	-	D2B1	-
D132	No function	X	-	-	D2B2	-
D133	Reserved	X	-	-	D2B3	-
D134	Reserved	X	-	-	D2B4	-
D135	Max. permitted power	X	-	-	D2B5	-
D136	Maximum power at derating end	X	X	-	D2B6	D236
D137	Module temperature at start of derating	X	X	-	D2B7	D237
D138	Module temperature at end of derating	X	X	-	D2B8	D238
D139	Reserved	X	-	-	D2B9	-
D13A	Reserved	X	-	-	D2BA	-
D13B	Maximum coil current	X	-	-	D2BB	-
D13C	Potentiometer characteristic point 1 X-coordinate (par. 2)	X	X	X	D2BC	D23C
D13D	Potentiometer characteristic point 1 Y-coordinate	X	X	X	D2BD	D23D

MODBUS address	Designation	Write ebm-papst	Write customer	Write end customer	Default address	Cust. setting address
	(par. 2)					
D13E	Potentiometer characteristic point 2 X-coordinate (par. 2)	X	X	X	D2BE	D23E
D13F	Potentiometer characteristic point 2 Y-coordinate (par. 2)	X	X	X	D2BF	D23F
D140 - D143	No function	X	X	X	D2C0 - D2C3	D240 - D243
D144	No function	X	X	-	D2C4	D244
D145	Limiting speed for running monitor	X	X	-	D2C5	D245
D146	Reserved	X	-	-	D2C6	-
D147	No function	X	X	X	D2C7	D247
D148	No function	X	X	X	D2C8	D248
D149	Transfer rate	X	X	-	D2C9	D249
D14A	Parity configuration	X	X	-	D2CA	D24A
D14B	No function	X	-	-	D2CB	-
D14C	No function	X	-	-	D2CC	-
D14D	Motor temperature at start of derating	X	X	-	D2CD	D2CD
D14E	Motor temperature at end of derating	X	X	-	D2CE	D2CE
D14F	No function	X	-	-	D2CF	-
D150	Shedding function	X	X	-	D2D0	D250
D151	Max. starting modulation level	X	-	-	D2D1	-
D152	Number of start attempts	X	X	-	D2D2	D252
D153	Relay dropout delay	X	X	X	D2D3	D253
D154	No function	X	-	-	D2D4	-
D155	Maximum power	X	X	-	D2D5	D255
D156 - D15A	Vacant	X	X	X	D2D6 - D2DA	D256 - D25A
D15B	Fail safe function running direction	X	X	-	D2DB	D25B
D15C	Fail safe function on/off	X	X	-	D2DC	D25C
D15D	Set value for fail safe function	X	X	-	D2DD	D25D
D15E	Fail-safe dropout delay	X	X	-	D2DE	D25E
D15F	Potentiometer characteristic limiting value for cable break	X	X	-	D2DF	D25F
D160 D161	Minimum sensor value	X	X	X	D2E0 D2E1	D260 D261
D162 D163	Maximum sensor value	X	X	X	D2E2 D2E3	D262 D263
D164 - D169	Sensor unit	X	X	X	D2E4 - D2E9	D264 - D269
D16A - D16F	Vacant	X	X	X	D2EA - D2EF	D26A - D26F
D170 - D17F	Customer data	X	X	-	D2F0 - D2FF	D270 - D27F
D180	Operating hours counter	X	-	-	-	-
D181	Operating minutes counter	X	-	-	-	-
D182	Error indicator	X	-	-	-	-
D183	Vacant	X	-	-	-	-
D184	1st error	X	-	-	-	-
D185	1st error timing	X	-	-	-	-
D186 -	Error history	X	-	-	-	-

MODBUS address	Designation	Write ebm-papst	Write customer	Write end customer	Default address	Cust. setting address
D19F	Error history timing					
D1A0	Reference value of DC-link voltage	X	-	-	-	-
D1A1	Reference value of DC-link current	X	-	-	-	-
D1A2 D1A3	Fan serial number	X	-	-	-	-
D1A4	Fan production date	X	-	-	-	-
D1A5 - D1AA	Fan type	X	-	-	-	-
D1AB - D1AC	Reserved	X	-	-	-	-
D1AD - D1AE	Reserved	X	-	-	-	-
D1AF	Reserved	X	-	-	-	-
D1B0	Reserved	X	-	-	-	-
D1B1	Reserved	X	-	-	-	-
D1B2	Reserved	X	-	-	-	-
D1B3	Reserved	X	-	-	-	-
D1B4 - D1B5	Reserved	X	-	-	-	-
D1B6	Reserved	X	-	-	-	-
D1B7	Reserved	X	-	-	-	-
D1B8 - D1BF	Reserved	X	-	-	-	-
D1C0 - D1C1	Reserved	X	-	-	-	-
D1C2 - D1E6	No function	X	-	-	-	-
D1E7	Reserved	X	-	-	-	-
D1E8	Reserved	X	-	-	-	-
D1E9	Reserved	X	-	-	-	-
D1EA - D1F8	Vacant	X	-	-	-	-
D1F9 - D1FF	No function	X	-	-	-	-
D300 - D340	No function	X	-	-	-	-
D341 - D35F	Vacant	X	-	-	-	-
D360 - D37F	Reserved	X	-	-	-	-
D380 - D3FF	Vacant	X	-	-	-	-

<sup>1)</sup> only in part

"No function" parameters have no influence on functionality in the Lite version. These parameters are only defined for the non-Lite version.

**Encoding of parameters:**

**Unless specified otherwise, parameters are encoded in "big endian" format, i.e. the byte with the highest-value bits comes first. This is particularly true for parameters that include multiple holding registers.**

## 2.2 Reset

Address : D000  
 Write authorisation : ebm-papst, customer, end customer

Encoding:

MSB	0	0	0	0	0	0	0	0
LSB	0	0	0	0	Reset	Error	Parameter	AWS Rst

A bit will trigger the following action in the fan when it is set:

- Reset : Software reset (includes "Reset error" and "Adopt parameters")  
 The software is launched at the start of booting
- Error : Errors are reset
- Parameter : All parameters are copied from the EEPROM into the RAM  
 This bit must be set to make revised parameters valid
- Reset AWS : User software reset  
 (includes "Reset error" and "Adopt parameters")  
 The software is reinitialised

After the action has been executed, the bit will automatically be reset by the fan.


When reset (bit 3), the program will be relaunched at the start of booting. Commands from the master will then be neither responded to nor executed for a few seconds.  
 Reset (bit 3) is only recommended if new user software is to be loaded using a bootloader (because communication is interrupted).  
 If only the software is to be reinitialised, user software reset (bit 0) is recommended. Then communication is not interrupted.

### 2.3 Default set value

Address : D001  
 Write authorisation : ebm-papst, customer, end customer

The "Default set value" parameter is used in MODBUS to specify a set value for each control mode. The condition for this is that the "Source set value" RS485 (1) is specified (see 2.10 Source set value). Otherwise, the parameter will have no function.

If the "Store" function is activated in the "Store set value" parameter (see 2.12 Save set value),, the value in the set value parameter (EEPROM) will be stored every time the default set value is write-accessed (see 2.19 Set value (EEPROM)).



The "Store set value" function must not be activated if the default set value is repeatedly write-accessed at frequent intervals!  
 If this instruction is not observed, there is a risk that the memory will be permanently destroyed.

If the function is activated, the maximum number of times the default set value can be write-accessed over the entire service life of the fan is 100,000.  
 For more information, see the fan data sheet.

The "Internal parameter set" parameter is used to select whether the value is stored in "Set value (parameter set 1)" or in "Set value (parameter set 2)" (see 2.13 Internal parameter set).

Following a reset, the motor will again run with this value, provided the parameter set selection has not been changed. (see 2.12 Save set value)

**Important!**  
 If the "Store set value" function is activated and the parameter set is changed (from 1 to 2 or vice versa), the default set value will automatically be changed to the corresponding parameter "Set value (parameter set 1)" or "Set value (parameter set 2)"!

Encoding:

Note: The 4 LSBits are of no relevance for the set value and will always be assumed to be 0.

a) in closed-loop speed control

The default set value denotes a speed:

$$Default\ set\ value\ [rpm] = \frac{Data\ bytes}{64000} \cdot nMax\ [rpm]$$

nMax [rpm] to maximum speed in revolutions per minute (see 2.23 Maximum speed)

The value zero means motor standstill

b) in open-loop PWM control

The default set value denotes a modulation level:

$$Default\ set\ value\ [\%] = \frac{Data\ bytes}{65536} \cdot 100\%$$

The value zero means motor standstill

**2.4 Password**

Address : D002 - D004  
 Write authorisation : ebm-papst, customer, end customer

Encoding:

$$Password = Data\ bytes$$

In order to prevent unauthorised writing of certain parameters, these are only written if the correct password for the necessary authorisation is entered at this point.  
 If a user forgets to reset the password, it will automatically be reset to 0x000000000000 after 4 minutes of inactivity.

When the "Password" parameter is read, the value 0 is always output to prevent a user with a lower authorisation level gaining access to a password to a higher level.

**2.5 Control default setting**

Address : D005  
 Write authorisation : ebm-papst, customer (in part)

Encoding:

MSB	0	0	0	0	0	0	0	0
LSB	0	0	0	0	0	Error	D -> W	W -> D

Setting the bit D -> W causes all parameters in the data range (D100 to D17F) to be copied to the default setting (range D280 to D2FF).  
 The authorisation level "ebm-papst" is needed to set this bit!

Setting the bit W -> D causes all parameters in the default setting (D280 to D2FF) to be copied to the data range (range D100 to D17F).  
 The authorisation level "customer" is sufficient to set this bit.

Once the copying process is complete, the bit will automatically be reset by the fan.

**Important!**  
 A copy procedure can only be carried out after the fan has stopped.  
 Otherwise the "Error" bit will be set and the copy procedure aborted.

## 2.6 Control customer setting

Address : D006  
 Write authorisation : ebm-papst, customer, end customer (in part)

Encoding:

MSB	0	0	0	0	0	0	0	0
-----	---	---	---	---	---	---	---	---

LSB	0	0	0	0	0	Error	D -> K	K -> D
-----	---	---	---	---	---	-------	--------	--------

Setting the bit D -> K causes all parameters in the data range (D100 to D17F) for which the customer has write authorisation to be copied to the customer setting (range D200 to D27F). The minimum authorisation level needed to set this bit is "customer"!

Setting the bit K -> D causes all parameters in the customer setting (range D200 to D27F) for which the customer has write authorisation to be copied to the data range (D100 to D17F). The authorisation level "end customer" is sufficient to set this bit.

Once the copying process is complete, the bit will automatically be reset by the fan.

**Important!**  
 A copy procedure can only be carried out after the fan has stopped.  
 Otherwise the "Error" bit will be set and the copy procedure aborted.

## 2.7 Operating hours counter

Address : D009  
           D180 (alternative)  
 Write authorisation : ebm-papst

Encoding:

$$\text{Operating time [h]} = \text{Data bytes}$$

After every hour the fan runs, the operating hours counter increases by 1.

With 16 bits, a maximum of 65,535 hours (approx. 7.5 years) can be counted.  
 If the operating hours counter overshoots this figure, it will no longer be written to, i.e. it will stay at 65,535.

To maintain compatibility with MODBUS 5.0 (non-Lite version), the operating hours counter can be addressed via the address D009 as well as via D180.  
 We recommend D009.

## 2.8 Operating minutes counter

Address : D00A  
          : D181 (alternative)  
Write authorisation : ebm-papst

Encoding:

Operating minutes [min] = Data bytes

After every full minute the fan runs, the operating minutes counter increases by 1.

To maintain compatibility with MODBUS 5.0 (non-Lite version), the operating minutes counter can be addressed via the address D009 as well as via D181.  
We recommend D00A.

## 2.9 Fan address

Address : D100  
Write authorisation : ebm-papst, customer, end customer

Encoding:

*Fan address* = *Data bytes (LSB)*

The MSB is of no relevance!

Permitted value range: 1 to 247



### 2.10 Source set value

Address : D101  
 Write authorisation : ebm-papst, customer, end customer

Encoding:

This parameter specifies the source from which the set value is taken:

Value	Source set value
0	Analogue input 0 to 10 V
1	RS485 (default set value parameter D001)

The MSB is of no relevance!

Permitted value range: 0 to 1

### 2.11 Preferred running direction

Address : D102  
 Write authorisation : ebm-papst, customer, end customer

Encoding:

This parameter specifies the preferred running direction.

Value	Preferred running direction
0	anti-clockwise
1	clockwise

The MSB is of no relevance!

Permitted value range: 0 to 1

After this parameter is written, the new value becomes applicable immediately (no reset necessary).

If the fail safe function is activated and a break in the MODBUS connection is detected, the direction of rotation is set to the value of "Fail safe function running direction" (see 2.38). The parameter Preferred running direction is not relevant then.

### 2.12 Save set value

Address : D103  
 Write authorisation : ebm-papst, customer, end customer


Encoding:

This parameter specifies whether or not an incoming default set value (D001) is also stored in the EEPROM under set value (EEPROM) (D114 / D115).

Value	Function
0	Set value is not stored The fan is stationary after a reset
1	Set value is stored in EEPROM After reset, the fan will run with the stored set value

The MSB is of no relevance!

Permitted value range: 0 to 1



The function must not be activated if the default set value is repeatedly write-accessed at frequent intervals (D001, see 2.3 Default set value)!  
 If this instruction is not observed, there is a risk that the memory will be permanently destroyed.

If the function is activated, the maximum number of times the default set value can be write accessed over the entire service life of the fan is 100,000.  
 For more information, see the fan data sheet.

### 2.13 Internal parameter set

Address : D105  
 Write authorisation : ebm-papst, customer, end customer

Encoding:

This parameter specifies whether parameter set 1 or parameter set 2 is used.

Value	Parameter set
0	Parameter set 1
1	Parameter set 2

The MSB is of no relevance!

Permitted value range: 0 to 1

After this parameter is written, the new value becomes applicable immediately (no reset necessary).

### 2.14 Control mode

Address of parameter set 1 : D106  
 Address of parameter set 2 : D107  
 Write authorisation : ebm-papst, customer, end customer

The "Internal parameter set" parameter is used to select whether the value in "Control mode (parameter set 1)" or the value in "Control mode (parameter set 2)" is applicable (see 2.13 Internal parameter set).

Encoding:

Value	Control mode
0	Closed-loop speed control
2	Open loop PWM control

The MSB is of no relevance!

Permitted value range: [0; 2]

A sensor can only be connected to measure and display the sensor actual value. Closed-loop control based on the sensor signal is not possible.. Control mode „Closed-loop sensor control“ (1) is not supported.

### 2.15 Control parameters

For closed-loop speed control, 2 parameters are provided:

- P factor kp
- I factor ki

Address of P factor (parameter set 1) : D10A  
 Address of P factor (parameter set 2) : D10B  
 Address of I factor (parameter set 1) : D10C  
 Address of I factor (parameter set 2) : D10D  
 Write authorisation : ebm-papst, customer, end customer

The "Internal parameter set" parameter is used to select whether the values in "P factor / I factor (parameter set 1)" or the values in "P factor / I factor (parameter set 2)" are applicable (see 2.13 Internal parameter set).

Encoding:

Each control parameter consists of 2 bytes.

a) P factor

$$P \text{ factor} = \frac{\text{Data bytes}}{256} \cdot 100\%$$

This means that values can be set for the P factor between 0 and 25,600%, in steps of 0.39%

b) I factor

$$I \text{ factor} = \frac{\text{Data bytes}}{65536} \cdot 100\%$$

This means that values can be set for the I factor between 0 and 100%, in steps of 0.00153%

Remarks concerning control behaviour:

After each scanning cycle time  $T_a = 50 \text{ ms}$ , the fan's controller computes a new modulation level according to the equation below:

$$a(n) = \frac{kp}{100\%} \cdot xd(n) + \sum_{i=0}^n \frac{ki}{100\%} \cdot xd(i)$$

$a(n)$  ..... Modulation level at time  $n$ , normalised to range 0 to 65536; i.e. 65536 - > 100%

$xd(n)$  ..... Normalised control deviation at time  $n$  (set value - actual value)

$kp$  ..... P factor in %

$ki$  ..... I factor in %

The normalised control variable is calculated from the set value and the actual value. Set value and actual value are also normalised to the maximum speed.

The normalised control equation stated above and the generally applicable control equation

$$A(n) = kr \cdot \left( Xd(n) + \frac{Ta}{Tn} \sum_{i=0}^n Xd(i) \right)$$

$A(n)$  ..... Modulation level at time  $n$  (0 to 100%)

$Xd(n)$  ..... Absolute control deviation at time  $n$  (set value - actual value)

$kr$  ..... Proportional gain

$Tn$  ..... Integral action time

$Ta$  ..... Sampling time

... can be converted to each other with

$$kr = \frac{kp}{\text{Max reference value} - \text{Min reference value}} \quad \text{or} \quad kp = kr \cdot (\text{Max reference value} - \text{Min reference value})$$

$$Tn = \frac{kp}{ki} \cdot Ta \quad \text{or} \quad ki = \frac{Ta}{Tn} \cdot kr \cdot (\text{Max reference value} - \text{Min reference value})$$

with

$$Ta = 50\text{ms}$$

$$\text{Max reference value} = 1,024 \cdot n\text{Max}$$

$$\text{Min reference value} = 0$$

$$n\text{Max} \dots\dots\dots \text{Maximum speed [rpm]}$$

## 2.16 Maximum modulation level

Address of parameter set 1 : D10E  
 Address of parameter set 2 : D10F  
 Write authorisation : ebm-papst, customer

The "Internal parameter set" parameter is used to select whether the value in "Maximum modulation level (parameter set 1)" or the value in "Maximum modulation level (parameter set 2)" is applicable (see 2.13 Internal parameter set).

Encoding:

$$\text{Max. modulation level [\%]} = \frac{\text{Data bytes}}{256} \cdot 100\%$$

The MSB is of no relevance!

Limitation:

8% < Maximum modulation level < Maximum permissible modulation level  
 (see 2.21 Maximum permissible modulation level)

## 2.17 Minimum modulation level

Address of parameter set 1 : D110  
 Address of parameter set 2 : D111  
 Write authorisation : ebm-papst, customer, end customer

The "Internal parameter set" parameter is used to select whether the value in "Minimum modulation level (parameter set 1)" or the value in "Minimum modulation level (parameter set 2)" is applicable (see 2.13 Internal parameter set).

Encoding:

$$\text{Min. modulation level [\%]} = \frac{\text{Data bytes}}{256} \cdot 100\%$$

The MSB is of no relevance!

Limitation:

Minimum modulation level > Minimum permissible modulation level  
 (see 2.22 Minimum permissible modulation level)

## 2.18 Enable motor stop

Address of parameter set 1 : D112  
 Address of parameter set 2 : D113  
 Write authorisation : ebm-papst, customer, end customer

The "Internal parameter set" parameter is used to select whether the value in "Enable motor stop (parameter set 1)" or the value in "Enable motor stop (parameter set 2)" is applicable (see 2.13 Internal parameter set).

Encoding:

Value	Motor stop
0	Motor runs continuously (even if set value = 0)
1	Motor stops if set value = 0

The MSB is of no relevance!

## 2.19 Set value (EEPROM)

Address of parameter set 1 : D114  
 Address of parameter set 2 : D115  
 Write authorisation : ebm-papst, customer, end customer

The information in these parameters is only applicable if the "Source set value" parameter has the value "RS485" (1) (see 2.10 Source set value) and the "Store set value" function is activated (see 2.12 Save set value). Otherwise, both parameters will have no function.

If the parameter "Default set value" is changed, the corresponding parameter "Set value (parameter set 1)" or "Set value (parameter set 2)" is set to the same value provided the function "Store set value" is activated (see 2.3 Default set value, 2.12 Save set value).

The "Internal parameter set" parameter is used to select whether the value is stored in "Set value (parameter set 1)" or in "Set value (parameter set 2)" (see 2.13 Internal parameter set).

As the default set value is stored in the volatile memory, it is necessary to store this value in the non-volatile memory as well (when needed). The "Set value (EEPROM)" parameter is provided for this purpose.

Encoding:

Note: The 4 LSBits are of no relevance for the set value and will always be assumed to be 0.

a) in closed-loop speed control

The set value denotes a speed:

$$\text{Set value [rpm]} = \frac{\text{Data bytes}}{64000} \cdot n_{\text{Max}} [\text{rpm}]$$

$n_{\text{Max}}$  [rpm] to maximum speed in revolutions per minute (see 2.23 Maximum speed)

The value zero means motor standstill

b) in open-loop PWM control

The set value denotes a modulation level:

$$\text{Set value [\%]} = \frac{\text{Data bytes}}{65536} \cdot 100\%$$

The value zero means motor standstill

**2.20 Starting modulation level**

Address : D116  
Write authorisation : ebm-papst

The starting modulation level specifies the PWM with which the motor is started after a start command.

Encoding:

$$\text{Starting modulation level [\%]} = \frac{\text{Data bytes}}{256} \cdot 100\%$$

The MSB is of no relevance!

The motor always starts with the starting modulation level.

A high starting modulation level results in a high current when the motor starts!

**2.21 Maximum permissible modulation level**

Address : D117  
Write authorisation : ebm-papst

This parameter defines the upper limit for the maximum modulation level (see 2.16 Maximum modulation level).

Encoding:

$$\text{Max. perm. modulation level [\%]} = \frac{\text{Data bytes}}{256} \cdot 100\%$$

The MSB is of no relevance!

## 2.22 Minimum permissible modulation level

Address : D118  
Write authorisation : ebm-papst

This parameter defines the lower limit for the minimum modulation level (see 2.17 Minimum modulation level).

Encoding:

$$\text{Min. perm. modulation level [\%]} = \frac{\text{Data bytes}}{256} \cdot 100\%$$

The MSB is of no relevance!

If too small a value is selected for this parameter (especially values < 8%), the motor may shut down with the error "Locked motor"!

## 2.23 Maximum speed

Address : D119  
Write authorisation : ebm-papst, customer

This parameter has two functions:

- All parameters with speed information (set values, actual values) are based on this value. The value 64,000 in this speed information corresponds to the maximum speed value given here.
- In the "open-loop PWM control" mode, the speed is limited to the value given here. (In closed-loop speed control, the speed is controlled in any event.)

Encoding:

$$\text{Max. speed [rpm]} = \text{Data bytes}$$

The maximum speed is made up of 2 bytes.

Limitation:

Maximum speed < Maximum permissible speed (see 2.24 Maximum permissible speed)



## 2.24 Maximum permissible speed

Address : D11A  
Write authorisation : ebm-papst

This parameter defines the upper limit for the maximum speed (see 2.23 Maximum speed).

Encoding:

*Max. perm. speed [rpm] = Data bytes*

The maximum permissible speed is made up of 2 bytes.

## 2.25 Ramp-up curve / ramp-down curve

Address of ramp-up curve : D11F  
Address of ramp-down curve : D120  
Write authorisation : ebm-papst, customer, end customer

These parameters define the ramp time for a set value change of 256 steps (i.e. the time for a change of the MSB set value by one step).  
The parameter "Ramp-up curve" defines the time for a positive change of the set value.  
The parameter "Ramp-down curve" defines the time for a negative change of the set value.

With closed-loop speed control, 256 steps correspond to a change in speed of  
 $\Delta n \text{ [rpm]} = n_{\text{Max}} \text{ [rpm]} / 250$

With open-loop PWM control, 256 steps correspond to a PWM change of  
 $\Delta \text{PWM} \text{ [\%]} = 0.39\%$

Encoding:

*Time for set value change by 256 steps [ms] = Data bytes · 10ms*

The MSB is of no relevance!

The result of a change from the minimum value 0 to the maximum value 64,000 is a ramp-up time or ramp-down time of

*Ramp time [s] = 250 · Time for set value change by 256 steps [ms] = Data bytes · 2,5s*

If the value is 0, the new set value is immediately effective without any time lag.

## 2.26 Limit speed

Address : D128  
 Write authorisation : ebm-papst

This parameter defines the limit value for the safety function "limit speed". The closed-loop speed control in this safety function ensures that the limit speed for the impeller is not exceeded.

Encoding:

$$\text{Limit speed [rpm]} = \text{Data bytes}$$

Note:

The limit speed is not related to the maximum speed!

## 2.27 Potentiometer characteristic

Address of point 1 X-coordinate, parameter set 1 : D12A  
 Address of point 1 Y-coordinate, parameter set 1 : D12B  
 Address of point 2 X-coordinate, parameter set 1 : D12C  
 Address of point 2 Y-coordinate, parameter set 1 : D12D

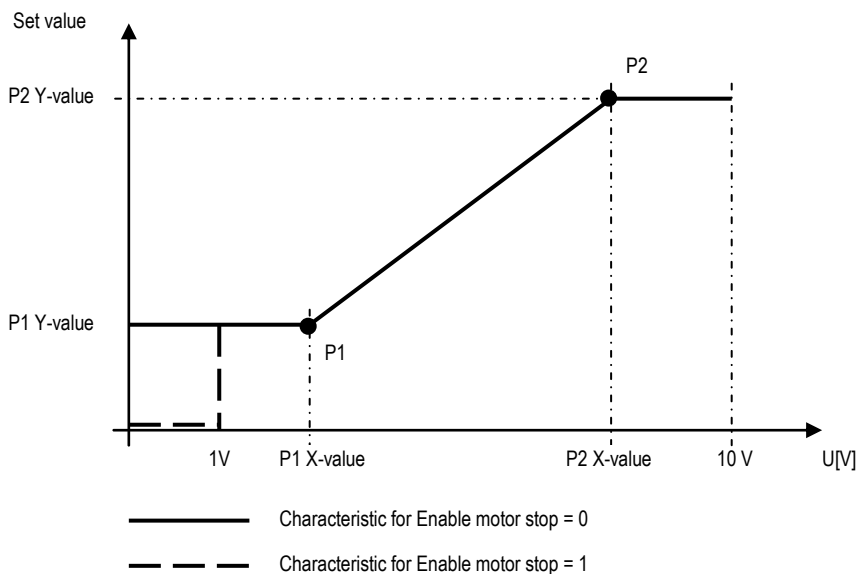
Address of point 1 X-coordinate, parameter set 2 : D13C  
 Address of point 1 Y-coordinate, parameter set 2 : D13D  
 Address of point 2 X-coordinate, parameter set 2 : D13E  
 Address of point 2 Y-coordinate, parameter set 2 : D13F

Write authorisation : ebm-papst, customer, end customer

2 different characteristics can be defined.

The "Internal parameter set" parameter is used to select whether the "Parameter set 1" values or the "Parameter set 2" values are applicable (see 2.13 Internal parameter set).

These parameters are used to assign a set value to the voltage at the control input.



Points P1 and P2 may be moved at random within the diagram.

Encoding:

The X-coordinate defines a voltage value between 0 and 10 V for the analogue input:

$$U[V](Px) = \frac{\text{Data bytes}}{65536} \cdot 10V$$

The Y-coordinate defines the associated set value for this point. Depending on the control mode, this may be a speed (closed-loop speed control), a sensor value (closed-loop sensor control) or a modulation level (open-loop PWM control).

a) closed-loop speed control:

$$\text{Set value [rpm]} = \frac{\text{Data bytes}}{64000} \cdot nMax \text{ [rpm]}$$

nMax [rpm] to maximum speed in revolutions per minute (see 2.23 Maximum speed)

b) in open-loop PWM control

$$\text{Set value [%]} = \frac{\text{Data bytes}}{65536} \cdot 100\%$$

For voltages at the analogue input that are less than the value defined by the point 1 X-coordinate, the set value is the value defined in the point 1 Y-coordinate.  
 For voltages at the analogue input that are greater than the value defined by the point 2 X-coordinate, the set value is the value defined in the point 2 Y-coordinate.  
 In between, the set value changes linearly between the two values specified in the Y-coordinates (see diagram above).

Limitation:

X-coordinate (point 1) ≤ X-coordinate (point 2)

Note:

If Y-coordinate (point 1) > Y-coordinate (point 2) is selected, the gradient of the characteristic will be negative. In such cases, the motor stop function will become effective at a voltage > 9 V at the analogue input.

### 2.28 Control limitation

Address : D12F  
 Write authorisation : ebm-papst

This parameter defines which limitation functions are activated.

Encoding:

Reserved	Reserved	0	0	Reserved	Reserved	I	P
----------	----------	---	---	----------	----------	---	---

The MSB is of no relevance!

I ..... Coil current limit  
 P ..... Output limit

### 2.29 Maximum power

#### 2.29.1 Maximum power

Address of maximum power : D155  
 Write authorisation : ebm-papst, customer

Encoding:

$$P_{max} [W] = \frac{\text{Data bytes}}{256} \cdot \text{Reference } U_z [V] \cdot \text{Reference } I_z [A]$$

P max ..... Maximum power  
 Ref. Uz ..... Reference variable DC-link voltage (see 2.43 Reference value of DC-link voltage).  
 Ref. Iz ..... DC-link current reference variable (see 2.44 Reference value of DC-link current)

The MSB is of no relevance!

The motor will limit the power to the value specified here if the function in the parameter "Control limitation" is activated (see 2.28 Control limitation).

Limitation:

Maximum power ≤ maximum permitted power (see 2.29.2 Maximum permitted power)

**2.29.2 Maximum permitted power**

Address for maximum permitted power : D135  
 Write authorisation : ebm-papst

Encoding:

$$P_{max, perm} [W] = \frac{Data\ bytes}{256} \cdot Reference\ Uz [V] \cdot Reference\ Iz [A]$$

P max, zul ..... Maximum permitted power  
 Ref. Uz ..... Reference variable DC-link voltage (see 2.43 Reference value of DC-link voltage).  
 Ref. Iz ..... DC-link current reference variable (see 2.44 Reference value of DC-link current)

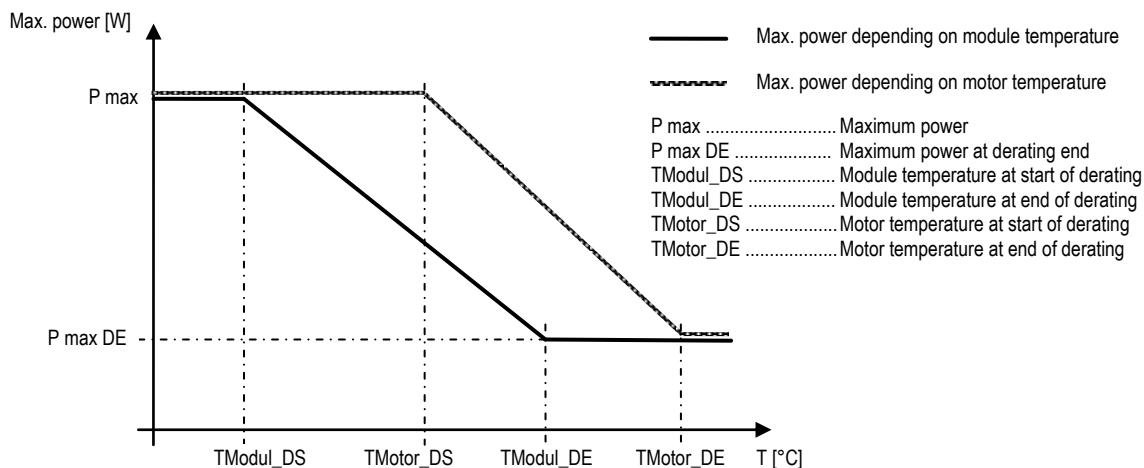
The MSB is of no relevance!

This parameter defines the upper limit for the maximum power (see 2.29.1) and the maximum power at derating end (see 2.29.3).

**2.29.3 Power derating**

Address for maximum power at derating end : D136  
 Address for module temperature power derating start : D137  
 Address for module temperature power derating end : D138  
 Address for motor temperature power derating start : D14D  
 Address for motor temperature power derating end : D14E  
 Write authorisation : ebm-papst, customer

Depending on the module temperature (see 3.13 Module temperature) and the motor temperature (see 3.14 Motor temperature), the maximum power decreases based on the following curve. Each temperature can have its own defined curve.



From this curve, the follow result

- A max. power value depending on the module temperature (Pmax\_TModul)
- A max. power value depending on the motor temperature (Pmax\_TMotor)

The power is limited to the smaller of the two Pmax\_TModul and Pmax\_TMotor values.

Encoding:

$$P_{max\_DE} [W] = \frac{\text{Data bytes}}{256} \cdot \text{Reference } U_z [V] \cdot \text{Reference } I_z [V]$$

$$T_{Modul\_DS} [^{\circ}C] = \text{Data bytes}$$

$$T_{Modul\_DE} [^{\circ}C] = \text{Data bytes}$$

$$T_{Motor\_DS} [^{\circ}C] = \text{Data bytes}$$

$$T_{Motor\_DE} [^{\circ}C] = \text{Data bytes}$$

P max DE..... Maximum power at derating end

Ref. Uz ..... Reference variable DC-link voltage (see 2.43 Reference value of DC-link voltage).

Ref. Iz ..... DC-link current reference variable (see 2.44 Reference value of DC-link current)

TModul DS ..... Module temperature at start of derating

TModul DE ..... Module temperature at end of derating

TMotor DS ..... Motor temperature at start of derating

TMotor DE ..... Motor temperature at end of derating

The MSB is of no relevance for any of these!

The values "Maximum power" and "Maximum power for temperature derating end" are the same for both curves.

For temperatures below the value defined in TModul DS or TMotor DS, maximum power corresponds to the value of P max.

For temperatures above the value defined in TModul DE or TMotor DE, maximum power corresponds to the value of P max DE.

In between, the maximum power changes linearly between the values of P max and P max\_DE. (see diagram above).

Limitation:

If T DS ≥ T DE, T DS is limited to T DE

If P max DE > P<sub>max</sub> or P<sub>max zul</sub>, P max DE is limited to P<sub>max</sub> or P<sub>max zul</sub>

### 2.30 Maximum coil current

Address : D13B  
 Write authorisation : ebm-papst

Encoding:

$$I_{max, eff} [A] = \frac{Data\ bytes}{170} \cdot Reference\ I_z [A]$$

Ref. Iz ..... DC-link current reference variable (see 2.44 Reference value of DC-link current)

The MSB is of no relevance!

If the function has been activated in the "Control limitation" parameter, the motor will limit the coil current (effective value) to the value specified here (see 2.28 Control limitation).  
 The coil current is made up of the DC-link current / modulation level

### 2.31 Limiting speed for running monitor

Address : D145  
 Write authorisation : ebm-papst, customer

Encoding:

The limit speed for the running monitor is made up of the parameter and the maximum speed:

$$Limit\ speed\ for\ running\ monitor [rpm] = \frac{Data\ bytes}{64000} \cdot nMax [rpm]$$

nMax [rpm] to maximum speed (in rpm) - see 2.23 Maximum speed

If the actual speed (see 3.8 Actual speed) is less than the limiting speed for the running monitor, the error relay will trip. The "n\_Low" flag is set in the "Warning" input register (see 3.10 Warning)  
 If "Limiting speed for running monitor" = 0, the entire function is deactivated.

## 2.32 Interface settings

### 2.32.1 Transfer rate

Address : D149  
 Write authorisation : ebm-papst, customer

Encoding:

Value	Transfer rate
0x01	2400 bit/sec
0x02	4800 bit/sec
0x03	9600 bit/sec
0x04	19,200 bit/sec
0x05	38,400 bit/sec
0x06	57,600 bit/sec
0x07	115,200 bit/sec

The MSB is of no relevance!

The standard value 19,200 bit/sec (0x04) is recommended for the transmission rate.

Make sure that the same transmission rate is selected for the fan as for the master (e.g. EC Control). Otherwise, communication with the fan will not be possible!

If the value set by the customer (holding register D249) or the factory default setting (holding register D2C9) differ from the value selected here, ensure that the transmission rate is also changed after data transfer from the customer setting or factory default setting (see 2.5 and 2.6). No communication is then possible with the previous setting

### 2.32.2 Parity configuration

Address : D14A  
 Write authorisation : ebm-papst, customer

Encoding:

Value	Parity configuration	Number of data bits	Parity	Number of stop bits
0x00	8E1	8	Even	1
0x01	8O1	8	Odd	1
0x02	8N2	8	None	2
0x03	8N1	8	None	1

The MSB is of no relevance!

The standard value 8E1 (0x00) is recommended for the parity configuration.



*Note:*

*If parity configuration 8N1 is selected, the unit will no longer correspond to the "MODBUS over Serial Line Specification & Implementation Guide V1.0" as this specifies a mandatory frame size of 11 bits.*

Make sure that the same parity configuration is selected for the fan as it is for the master (e.g. EC Control). Otherwise, communication with the fan will not be possible!

If the value set by the customer (holding register D249) or the factory default setting (holding register D2C9) differ from the value selected here, ensure that the parity configuration is also changed after data transfer from the customer setting or factory default setting (see 2.5 and 2.6). No communication is then possible with the previous setting.

**2.32.3 Procedure for changing the interface settings**

If the parameters "Transmission rate" and "Parity configuration" do not correspond to the interface setting in the master (e.g. EC Control), no communication will be possible.

If the settings are different, the interface settings will not be checked by the fan, i.e. the interface settings of the fan **MUST** be known.

*To prevent complications caused by interfaces with different settings, it is advisable to leave the interface settings at the preset values:*

*Transmission rate: : 19,200 bit/sec (0x04)*  
*Parity configuration : 8E1 (0x00)*

If a change is necessary, take care to ensure that the master and fan have the same settings at all times to guarantee error-free communications.

The new interface settings for the fan will only be adopted after an appropriate command on the "Reset" parameter (Adopt parameter or Reset) - see 2.2 - or when the unit is switched off ("Power off"). Until then, the fan will continue working with the previous settings.

Recommended procedure for changing the interface settings, for example when changing from 19,200 bit/sec, 8E1 to 38,400 bit/sec, 8N2:

- Interface settings at the master must initially be 19,200 bit/sec, 8E1.
- Write password: XX 10 D0 02 00 03 06 PP PP PP PP PP CS CS  
(changing the interface settings requires at least "Customer" authorisation)
- Write the transmission speed with the value 38,400 bit/sec: XX 06 D1 49 00 05 CS CS
- Write the parity configuration with the value 8N2: XX 06 D1 4A 00 02 CS CS
- Trigger Adopt parameters: XX 06 D0 00 00 02 CS CS
- After the fan has confirmed the command, it will work with the new setting.
- Change the interface setting in the master: 38,400 bit/sec, 8N2

XX ..... Fan address  
 PP ..... Password  
 CS ..... CRC checksum

#### 2.32.4 Interface emergency function:

Two default settings for the interface are stored in the fan.

The fan switches automatically to these default settings for short periods if faulty telegrams are received (for example, as a result of different interface settings on the master and slave):

- After 10 failed attempts, the fan will automatically communicate using the setting 19,200 bit/sec, 8E1.
- After another 10 failed attempts, the fan will automatically communicate using the setting of 2,400 bit/sec, 8E1.  
Exception: For configuration of 115,200 bit/sec, this step is omitted.
- After yet another 10 failed attempts, the fan switches back to the settings for "Transmission speed" and "Parity configuration" stored in the holding registers to prevent unwanted continuous activations of the interface emergency function.
- For additional failed attempts, the interface is no longer switched over automatically, i.e. the interface emergency function can be activated only by switching the device off and on again or carrying out a software reset.

The interface emergency function can only be activated when no valid telegram has been detected from the fan. To make sure that the interface emergency function is enabled, the operating voltage of the unit should first be switched off and back on again.

#### *Application:*

If the interface settings of the fan are unknown, they can be identified with the interface emergency function.

Recommended procedure:

- Switch fan on
- Set interface on master to standard values 19,200 bit/sec, 8E1
- Read interface setting: XX 03 D1 49 00 02 CS CS
- After 10 failed attempts, the fan will automatically communicate using the standard setting of 19,200 bit/sec, 8E1. In other words, the command given above has to be transmitted 11 times until the fan transmits a response.
- The fan's interface settings can be determined from the response it transmits.
- The interface settings of the master and slave can now be synchronised.

#### *Note:*

The fan should not be run continuously in the interface emergency function as the values defined in the parameters "Transmission rate" and "Parity configuration" are activated again after every "Reset" or "Adopt parameters" command (see 2.2).

The interface settings of the master and fan must be synchronised!

### 2.33 Shedding function

Address of shedding function : D150  
 Write authorisation : ebm-papst, customer

Address for maximum starting modulation level : D151  
 Write authorisation : ebm-papst

Address for number of startup attempts : D152  
 Write authorisation : ebm-papst, customer

Encoding:

a) Shedding function

Value	Shedding function
0	Inactive
1	Active

The MSB is of no relevance!  
 Permitted value range: 0 to 1

b) Maximum starting modulation level

$$\text{Max. starting modulation level [\%]} = \frac{\text{Data bytes}}{256} \cdot 100\%$$

The MSB is of no relevance!

c) Number of startup attempts

$$\text{Number of startup attempts} = \text{Data bytes}$$

The MSB is of no relevance!

Function:

The shedding function is intended to enable a fan with fan blades that have frozen into place to shed the motor free by attempting to start the fan in both directions in alternation; during this process, the modulation level is increased during each attempt.

The first attempt begins with the value specified under starting modulation level (see 2.20 Starting modulation level) and the desired direction of rotation. If this does not succeed in getting the fan rotating, during each additional attempt, the direction of rotation is reversed and the starting modulation level is increased by 5%, to a value no higher than that specified in "Maximum starting modulation level". At the same time, a shedding function warning is generated (see 3.10 Warning).

This process is continued until the number of attempts specified in the "Number of startup attempts" is reached. If this also does not succeed in getting the fan rotating, the "Locked motor" error message (see 3.9 Motor status) is generated. During additional startup attempts, the modulation level is returned to the value defined in the parameter "Starting modulation level" and not increased any more.

**2.34 Relay dropout delay**

Address : D153  
 Write authorisation : ebm-papst, customer, end customer

In the event of a fault, the relay drops out after a delay equal to the time specified here.

Encoding:

$$Relay\ dropout\ delay\ [s] = Data\ bytes$$

The MSB is of no relevance!

**2.35 Fail safe function on/off**

Address : D15C  
 Write authorisation : ebm-papst, customer

Encoding:

This parameter specifies whether the fail safe function is active or inactive.

Value	Fail safe function
0	Inactive
1	Active

The MSB is of no relevance!  
 Permitted value range: 0 to 1

**Function:**

Fail safe function active (1)

The fail safe function monitors the signal wire of the set value and reacts to a failure of the signal. The function depends on the "Source set value" parameter (see 2.10):

*Source set value = 0 (analogue input): Monitoring of the input for set value (potentiometer):*

If the voltage at the set value input drops below the value in the "Potentiometer characteristic – limit for cable break" parameter (see 2.39), the fan will automatically switch to the set value for fail safe function (see 2.36).

*Source set value = 1 (RS485): Monitoring the RS485 bus interface:*

If no command is sent to the fan for the time specified in the "fail safe function time lag" parameter (see 2.37) the fan will automatically switch to the fail safe function set value (see 2.36).

The direction of rotation is simultaneously set to the value of "fail safe function running direction" (see 2.38).

Fail safe function inactive (0)

If the fail safe function is inactive, the signal wire for the set value is not monitored and is not switched to the set value for fail safe function.

**2.36 Fail safe function set value**

Address : D15D  
Write authorisation : ebm-papst, customer

If a failure of the set value signal is detected, the fan will automatically switch to the set value for fail safe function given here. For more information about the function, see 2.35.

Encoding:

Remarks: The encoding corresponds to the encoding of the default set value (D001)  
The 4 LSBits are of no relevance for the set value and will always be assumed to be 0.

a) in closed-loop speed control

The fail safe function set value denotes a speed:

$$\text{Fail safe function set value [rpm]} = \frac{\text{Data bytes}}{64000} \cdot nMax \text{ [rpm]}$$

nMax [rpm] to maximum speed in revolutions per minute (see 2.23 Maximum speed)

The value zero means motor standstill

b) in open-loop PWM control

The fail safe function set value denotes a modulation level:

$$\text{Fail safe function set value [\%]} = \frac{\text{Data bytes}}{65536} \cdot 100\%$$

The value zero means motor standstill

**2.37 Fail safe function time lag**

Address : D15E  
Write authorisation : ebm-papst, customer

*Only for source set value = 0x01 (RS485):*

If no command is sent to the fan for the time specified here, the fan will automatically switch to the fail safe function set value (see 2.36 Fail safe function set value), provided that the fail safe function is activated. For more information about the function, see 2.35.

Encoding:

$$\text{Fail safe function time lag [ms]} = \text{Data bytes} \cdot 100\text{ms}$$

The MSB is of no relevance!

**2.38 Fail safe function running direction**

Address : D15B  
Write authorisation : ebm-papst, customer

If the fail safe function is activated and a break in the MODBUS connection is detected, the direction of rotation is set to the value defined here. The setting for the Preferred running direction parameter (see 2.11) is then irrelevant.

Encoding:

Value	Fail safe function running direction
0	anti-clockwise
1	Clockwise
2	none (configured direction of rotation remains unchanged)

The MSB is of no relevance!

Permitted value range: 0 to 2

**2.39 Potentiometer characteristic limiting value for cable break**

Address : D15F  
 Write authorisation : ebm-papst, customer

*Only for source set value = 0x00 (analogue input):*

If the voltage at the set value input drops below the value defined here, the fan will automatically switch to the fail safe function set value (see 2.36), provided the fail safe function is activated. For more information about the function, see 2.35.

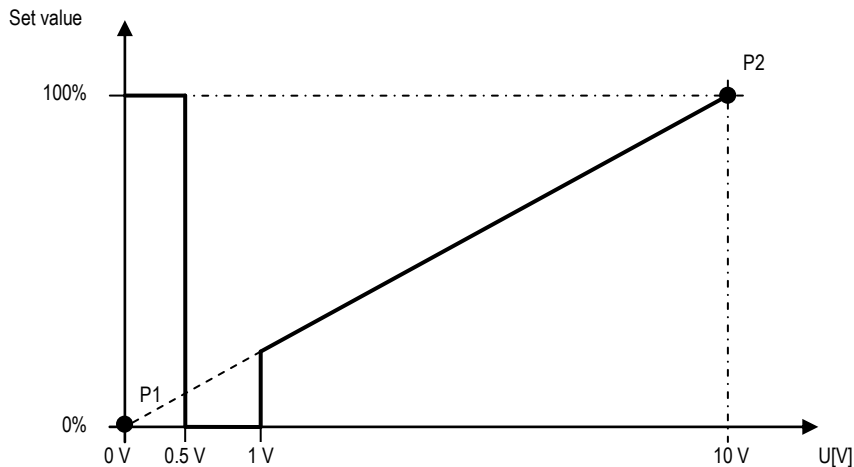
Encoding:

$$U\_cable\ break\ [V] = \frac{Data\ bytes}{65536} \cdot 10V$$

Example:

Fail safe function	= on
Set value for fail safe function	= 100%
Potentiometer characteristic U_cable break	= 0.5V
Potentiometer characteristic P1	= (0 V   0%)
Potentiometer characteristic P2	= (10 V   100%)
Enable motor stop	= 1

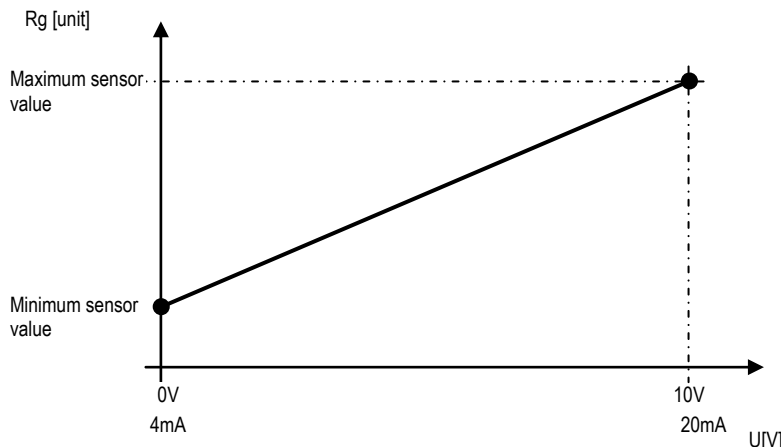
From this, the following curve results:



**2.40 Sensor**

Address for min. sensor value : D160 / D161  
 Address for max. sensor value : D162 / D163  
 Address of sensor unit : D164 - D169  
 Write authorisation : ebm-papst, customer, end customer

These values define the characteristic of the specified sensor as shown in the following diagram.



The necessary data for min. sensor value and max. sensor value can be found in the data sheet for the sensor.

These parameters are only intended for use in the master. The behaviour of the fan is not influenced by these parameters.

Encoding:

a) Minimum sensor value

$$\text{Min. sensor value [phys. unit]} = \text{Data bytes}$$

The minimum sensor value is stored in "IEEE float" format.

b) Maximum sensor value

$$\text{Max. sensor value [phys. unit]} = \text{Data bytes}$$

The maximum sensor value is stored in "IEEE float" format.

c) Sensor unit

$$\text{Sensor unit [ASCII]} = \text{Data bytes}$$

The sensor unit is stored in "ASCII" format.



## 2.41 Customer data

Address : D170 - D17F  
 Write authorisation : ebm-papst, customer

For the customer, a total of 16 parameters (each with 16 bits) is available in this range. Any values can be stored here.

The behaviour of the fan is not influenced by these parameters.

## 2.42 Error history

Address of error indicator : D182  
 Address of 1st error: : D184  
 Address of 1st error time: : D185  
 Address of error history 1 to 13 : D186, D188, D18A, D18C, D18E, D190, D192, D194, D196, D198, D19A, D19C, D19E  
 Address of error history time 1 to 13 : D187, D189, D18B, D18D, D18F, D191, D193, D195, D197, D199, D19B, D19D, D19F  
 Write authorisation : ebm-papst

### A) 1st error

The first error that is detected in the service life of the fan is stored under the "1st error" parameter. At the same time, the reading on the operating hour counter at this time is stored in the parameter "1st error time". The parameters are automatically written to by the fan.

### B) Error history

The error history contains the last 13 errors that were detected in the fan. The record of each error includes the respective operating hours counter reading in the parameter "Error history timing". The parameters are automatically written to by the fan.

The error indicator (D182) specifies the address of the last error to be detected in the error history. The previous error is then at the preceding address.

Example:

Error indicator = D196

Then:

D196	Last error (error n)	D197	Time of last error (error n)
D194	Error n-1	D195	Time of error n-1
D192	Error n-2	D193	Time of error n-2
D190	Error n-3	D191	Time of error n-3
D18E	Error n-4	D18F	Time of error n-4
D18C	Error n-5	D18D	Time of error n-5
D18A	Error n-6	D18B	Time of error n-6

D188	Error n-7	D189	Time of error n-7
D186	Error n-8	D18D	Time of error n-8
D19E	Error n-9	D19F	Time of error n-9
D19C	Error n-10	D19D	Time of error n-10
D19A	Error n-11	D19B	Time of error n-11
D198	Error n-12	D199	Time of error n-12

Every time a new error is detected, the error indicator increases by 2 and the error and its timing are stored at the address shown on the error indicator. The addresses are automatically written to by the fan. When the error indicator reaches the final value D19E, it will be reset to the initial value D186 when the next error is detected.

Encoding:

Error indicator:

*Address of error indicator = Data bytes*

Error:

MSB	0	0	0	0 <sup>*)</sup>	0	0	0	0
LSB	BLK	HLL	TFM	0 <sup>*)</sup>	0	TFE	0	0 <sup>*)</sup>

If bit = 1, the error described below has been detected:

- BLK: Locked motor
- HLL: Hall sensor error
- TFM: Motor overheated
- TFE: Power mod overheated

<sup>\*)</sup> The errors "DC-link undervoltage" and "Phase failure" and the general fault detection "Fan bad" are not stored, even if they are set in the motor status (see Motor status)!

Error timing:

*Time of error [h] = Data bytes*

## 2.43 Reference value of DC-link voltage

Address : D1A0  
Write authorisation : ebm-papst

To keep the resolution variable, all values for the DC-link voltage are based on this reference value.

Encoding:

$$\text{Reference } U_z [\text{mV}] = \text{Data bytes} \cdot 20\text{mV}$$

## 2.44 Reference value of DC-link current

Address : D1A1  
Write authorisation : ebm-papst

To keep the resolution variable, all values for the DC-link current are based on this reference value.

Encoding:

$$\text{Reference } I_z [\text{mA}] = \text{Data bytes} \cdot 2\text{mA}$$

## 2.45 Production data

Production data is information to enable a device to be traced. The behaviour of the fan is not influenced by these parameters.

### 2.45.1 Fan serial number and production date

Address of serial number : D1A2 / D1A3  
 Address of production date : D1A4  
 Write authorisation : ebm-papst

ebm-papst assigns an individual serial number to each fan. The part that our customers can extract/read consists of the production date and the consecutive number. Addressing can be easily done using this data. In the unlikely event that you encounter any problem doing so, please let us know.

Format: YYWW00XXXX

YY : Production year  
 WW : Calendar week of production  
 00 : Fixed value 00  
 XXXX : Sequential number

The first 4 digits contain the production date (year/calendar week).  
 With the beginning of each production week, the number XXXX starts counting up from zero and is increased by 1 for each fan. Each character can represent values from 0-9 and from A-Z. The maximum number of characters that can be encoded for each digit is thus 36, i.e.  $36^4 = 1\,679\,616$  devices/week.

#### a) Serial number (D1A2 / D1A3)

In the parameter "Serial number", only the part XXXX (sequential number) is stored.  
 Each digit is encoded as an ASCII value.

Encoding:

*Serial number [ASCII] = Data bytes*

#### b) Production date (D1A4)

The YYWW part is stored in the parameter "Production date".  
 Each digit is encoded as a hex value.

Encoding:

*Production year = Data bytes (MSB)*

*Production week = Data bytes (LSB)*

#### Example:

ebm-papst serial number: 092300**12GY**

-> Content D1A2: 0x4759 (GY)

Content D1A3: 0x3132 (12)

Content D1A4: 0x0917 (09/23)

### 2.45.2 Fan type

Address : D1A5 - D1AA  
Write authorisation : ebm-papst

Encoding:

The fan type is stored here in ASCII code

D1A5 contains the first two characters.

D1AA contains the last two characters.

### 3 Input Register

#### 3.1 Overview

Input registers are stored in the RAM of the fan.  
Input registers only have read access

The following list gives an overview of all parameters.  
The function of the parameters is described in the following chapters

MODBUS address	Designation
D000	Identification
D001	Maximum number of bytes
D002	Software name of bus controller
D003	Software version of bus controller
D004	Software name of commutation controller
D005	Software version of commutation controller
D010	Actual speed
D011	Motor status
D012	Warning
D013	DC-link voltage
D014	DC-link current
D015	Module temperature
D016	Motor temperature
D017	Electronics temperature
D018	Current direction of rotation
D019	Current modulation level
D01A	Current set value
D01B	Sensor actual value
D01C	No function
D01D	Current parameter set
D01E - D020	No function
D021	Current power
D022	Reserved
D023	Sensor actual value 1
D024 - D026	No function

"No function" parameters have no influence on functionality in the Lite version.  
These parameters are only defined for the non-Lite version.

### 3.2 Identification

Address : D000

Identification specifies the type of electronics and protocol concerned.

Encoding:

Value	Device	Specification version
00 01	ebm-papst series 84 / 112 / 150	1.02 <sup>*)</sup>
00 02	ebm-papst series 84 / 112 / 150	2.01, 3.00 - 3.01 <sup>*)</sup>
00 03	Customer application	)
00 04	Customer application	)
00 05	Customer application	)
00 06	ebm-papst series 84 / 112 / 150	3.02 <sup>*)</sup>
00 07	ebm-papst series 84 / 112 / 150	4.00 <sup>*)</sup>
00 08	ebm-papst series 84 / 112 / 150 / 200	5.00 <sup>*)</sup>
00 09	Customer application	)
00 0A	ebm-papst series 84 / 112 / 150 / 200 Lite	5.00 Lite <sup>*)</sup>
00 0B	ebm-papst series 84 / 112 / 150 / 200 Lite + addon module	5.00 <sup>*)</sup>
00 0C	Special applications for ebm-papst series 84 / 112 / 150 / 200	5.02 <sup>*)</sup>
00 0D	ebm-papst series 84 / 112 / 150 / 200 Lite	5.01 Lite
04 00	Customer application	)
05 00	Customer application	)
06 00	Customer application	)
07 00	Customer application	)
08 00	Customer application	)

<sup>\*)</sup> Devices with identification ≠ 0x000D do not correspond to this specification. In such cases, the corresponding document should be used.

### 3.3 Maximum number of bytes

Address : D001

This parameter specifies the maximum number of bytes that a telegram sent via MODBUS may contain.

Encoding:

*Max. number of bytes = Data bytes*

### 3.4 Software name of bus controller

Address : D002

This parameter specifies the number of the bus controller software (without the version).

Encoding:

*Software Name* = *Data bytes*

### 3.5 Software version of bus controller

Address : D003

This parameter specifies the software version of the bus controller.

Encoding:

*SoftwareVersion* = *Data bytes LSB*

The MSB is always 0.

### 3.6 Software name of commutation controller

Address : D004

This parameter specifies the number of the software of the commutation controller (without the version).

Encoding:

*Software Name* = *Data bytes*

### 3.7 Software version of commutation controller

Address : D005

This parameter specifies the software version of the commutation controller.

Encoding:

*SoftwareVersion* = *Data bytes LSB*

The MSB is always 0.



### 3.8 Actual speed

Address : D010

Encoding:

The actual speed is made up of the speed parameter and the maximum speed:

$$Actual\ speed\ [rpm] = \frac{Data\ bytes}{64000} \cdot nMax\ [rpm]$$

nMax [rpm] to maximum speed (in rpm) - see 2.23 Maximum speed

Note:

If the actual speed exceeds the value "1.02 \* maximum speed", the display will be limited to the value "1.02 \* maximum speed" (0xFFFF0)

### 3.9 Motor status

Address : D011

The motor status specifies errors currently detected in the fan.

Encoding:

MSB	0	0	0	UzLow	0	0	0	0
-----	---	---	---	-------	---	---	---	---

LSB	BLK	HLL	TFM	FB	0	TFE	0	PHA
-----	-----	-----	-----	----	---	-----	---	-----

If a bit is set, the error described below has been detected:

UzLow: DC-link undervoltage

BLK: Locked motor

HLL: Hall sensor error

TFM: Motor overheated

FB: Fan bad (general error) \*)

TFE: Power mod overheated

PHA: Phase failure (3-phase devices) or mains undervoltage (1-phase devices)

\*) "Fan bad" is set for every error

### 3.10 Warning

Address : D012

A warning is a stage preliminary to an error message, i.e. the limit value for the error message has almost been reached.

Encoding:

A set bit makes the warning active:

MSB	LRF	UeHigh	nGrenz	UzHigh	0	Cable break	n_Low	0
-----	-----	--------	--------	--------	---	-------------	-------	---

LSB	Brake	UzLow	TEI_high	TM_high	TE_high	P_Limit	0	I_Limit
-----	-------	-------	----------	---------	---------	---------	---	---------

- LRF : Shedding function active (see 2.33 Shedding function)
- UeHigh : Supply voltage high
- nGrenz : Limiting speed exceeded
- UzHigh : DC-link voltage high
- Cable break : cable break at set value analogue input  
(Voltage at the analogue input < Limit value for cable break - see 2.39)
- n\_Low : Actual speed is less than limit speed for running monitor  
(see 2.31 Limiting speed for running monitor)
- Brake : Brake operation: set if exterior drive is applied in opposite direction with high speed for prolonged period
- UzLow : DC-link voltage low
- TEI\_high : Electronics interior temperature high
- TM\_high : Motor temperature high
- TE\_high : Output stage temperature high
- P\_Limit : Power limitation in mesh
- I\_Limit : Current limitation in mesh

### 3.11 DC-link voltage

Address : D013

Encoding:

$$U_z [V] = \frac{\text{Data bytes}}{256} \cdot \text{Reference } U_z [V]$$

Reference Uz .....DC-link voltage reference variable (see 2.43 Reference value of DC-link voltage).

### 3.12 DC-link current

Address : D014

Encoding:

$$I_z [A] = \frac{\text{Data bytes}}{256} \cdot \text{Reference } I_z [A]$$

Reference Iz ..... DC-link current reference variable (see 2.44 Reference value of DC-link current)

### 3.13 Module temperature

Address : D015

Encoding:

$$T\_Modul [^{\circ}C] = \text{Data bytes}$$

### 3.14 Motor temperature

Address : D016

Encoding:

$$T\_Motor [^{\circ}C] = \text{Data bytes}$$

Format: signed integer

### 3.15 Interior electronics temperature

Address : D017

Encoding:

$$T\_EI [^{\circ}C] = \text{Data bytes}$$

Format: signed integer

### 3.16 Current direction of rotation

Address : D018

Encoding:

Value	Current direction of rotation
0	anti-clockwise
1	clockwise

### 3.17 Current modulation level

Address : D019

Encoding:

$$\text{Modulation level} = \frac{\text{Data bytes}}{65536} \cdot 100\%$$

### 3.18 Current set value

Address : D01A

Encoding:

#### a) in closed-loop speed control

The set value denotes a speed:

$$\text{Set value [rpm]} = \frac{\text{Data bytes}}{64000} \cdot n_{Max} \text{ [rpm]}$$

$n_{Max}$  [rpm] to maximum speed in revolutions per minute (see 2.23 Maximum speed)

The value zero means motor standstill

#### b) in open-loop PWM control

The set value denotes a modulation level:

$$\text{Set value [\%]} = \frac{\text{Data bytes}}{65536} \cdot 100\%$$

The value zero means motor standstill

### 3.19 Sensor actual values

Address of actual sensor value : D01B  
 Address of actual sensor value 1 : D023

One sensor can be connected to the fan.  
 Actual sensor value and actual sensor value 1 outputs the actual value of the external sensor at analog input ("set value input").

Encoding:

$$Actual\ value\ [V] = \frac{Data\ byte}{65536} \cdot 10V$$

$$Actual\ value\ [unit(Rg)] = Rg\ (U) = Rg\ \left(\frac{Data\ byte}{65536} \cdot 10V\right)$$

Actual value [V] = output voltage of sensor  
 Rg (U) = characteristic of sensor depending on voltage

A sensor can only be connected to measure and display the sensor actual value. Closed-loop control based on the sensor signal is not possible.. Control mode „Closed-loop sensor control“ is not supported (see 2.14)

### 3.20 Current parameter set

Address : D01D

This parameter specifies which parameter set (1 or 2) is currently in use.

The value of the "Internal parameter set" is always shown here (see 2.13 Internal parameter set).

Encoding:

Value	Parameter set
0	Parameter set 1
1	Parameter set 2

### 3.21 Current power

Address : D021

Encoding:

$$P [W] = \frac{\text{Data bytes}}{65536} \cdot \text{References } U_z [V] \cdot \text{References } I_z [A]$$

Reference Uz ..... DC-link voltage reference variable (see 2.43 Reference value of DC-link voltage)

Reference Iz ..... DC-link current reference variable (see 2.44 Reference value of DC-link current)