Modbus register

Modbus parameter extract

Table of contents

Foreword

This document specifies the Modbus parameters of the "ESL" device from ebm-papst. Knowledge of the general Modbus specifications is assumed:

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

The documents are available on the Internet at modbus.org

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

The source of this document comes from ebm-papst.

1 Protocol frame

The transmission of data by means of the data transmission
Modbus protocol takes place exclusively in an environment defined as a master-slave system. The orderly data flow is determined by the master. A slave must respond to its command request! Therefore, when setting up a system, care must be taken to ensure that a slave address is not assigned twice.

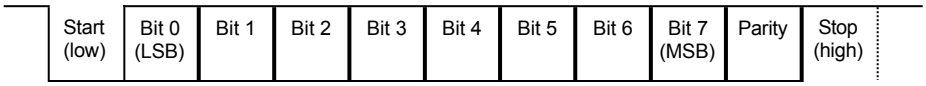
A twisted pair cable with RS485 standard is preferably as the transmission medium.

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

The 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:

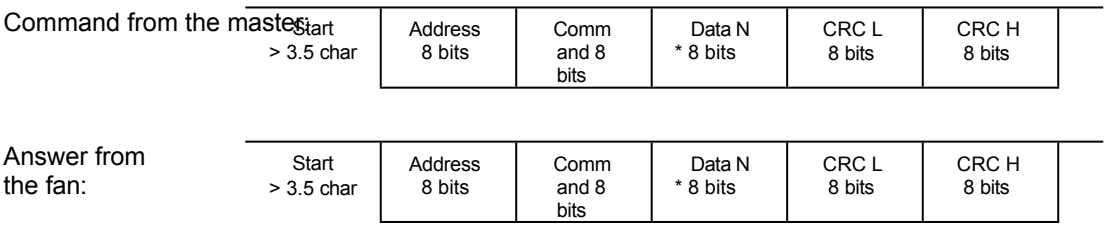


The parity bit is permanently set to "Even".
The transmission speed is fixed at 19200 baud. 8 data bits and one stop bit are used.

19200 baud / 8 data bits / parity "even" / 1 stop bit

1.2 Communication process

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



In deviation from the general specification, the maximum telegram length is 23 bytes!

1.2.1 Command from the 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 next 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..247 are permitted
Address 0 is intended for broadcast commands (i.e. command to all fans in the network).

Command:

The following commands of the general specification "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

The 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:

The number of data bytes and their meaning varies depending on the command. See 1.3 Commands

CRC L / CRC H

A checksum CRC is formed over the entire telegram.
The polynomial for checksum formation is $1+x^2+x^{15}+x^{16}$ (i.e. XOR - link with 0xA001). The start value is 0xFFFF.

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

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

1.2.2 Response from the fan

A fan only responds if

- it was addressed under its own address. No response is sent for the broadcast address.
- the telegram length is a maximum of 23 bytes.
- the correct number of data bytes has been sent in order to be able to interpret the telegram.
- the checksum was recognised correctly.

Initial synchronisation:

After completion of the command from the master, the fan waits a transmission pause of *at least* 3.5 bytes. Depending on the command and processing time, the pause can also be considerably longer (until the fan has all the requested data available)

Address:

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

Command:

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

If the command cannot be processed, the fan responds with an exception. The MSB is set in the command.
Then, for example, the command byte for the "Read Holding Register (0x03)" command is

0x83. Data:

The number of data bytes and their meaning varies depending on the command. See 1.3 Commands.

CRC L / CRC H

A checksum CRC is formed over the entire telegram.
The checksum creation does not differ 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 can be used to read the content of several holding registers. Holding registers are parameters that have both read and write access

Command from the master:

4 data bytes are transmitted:
- 1. holding register address MSB
- 1. holding register address LSB
- Number of addresses to be read MSB
- Number of addresses to be read LSB

The description of the holding registers follows later.

Answer from the fan:

The following data bytes are transmitted:
- Byte count (number of addresses to be read * 2)
- Data 1st Holding Register MSB
- Data 1st holding register LSB

optional:
- Data of the following holding registers (0..n)

Exception codes:

In the event of an error, only one data byte (exception code) is

transmitted Exception codes:

0x02: No permissible holding register address.
0x03: The maximum telegram length for the response (23 bytes) was exceeded i.e. more than 9 holding registers or 0 holding registers were requested.

1.3.2 Read Input Register

Command code: 0x04

This command can used to read the content of several input registers. Input registers are parameters that can only be read.

Command from the master:

4 data bytes are transmitted:
- 1. input register address MSB
- 1. input register address LSB
- Number of addresses to be read MSB
- Number of addresses to be read LSB

The description of the input registers follows later.

Answer from the fan:

The following data bytes are transmitted:
- Byte count (number of addresses to be read * 2)
- Data 1st Holding Register MSB
- Data 1st holding register LSB

optional:
- Data of the following input registers (0..n)

Exception codes:

In the event of an error, only one data byte (exception code) is

transmitted Exception codes:

0x02: The permissible range of the input registers 0xD000 or 0xE201 ... 0xE20B was exceeded 0x03: The maximum telegram length for the response (23 bytes) has been exceeded i.e. more than 9 input registers or 0 input registers were requested.

1.3.3 Write Single Register

Command code: 0x06

This command can used to write the content of *a* holding register. Command from the

master:

4 data bytes are transmitted:
- Holding register address MSB
- Holding Register Address LSB
- Data to be written MSB
- Data to be written LSB

The description of the holding registers follows later. Response from

the fan:

4 data bytes are transmitted:
- Holding register address MSB
- Holding Register Address LSB
- Data to be written MSB
- Data to be written LSB

Exception codes:

In the event of an error, only one data byte (exception code) is

transmitted Exception codes:

0x02: No permissible holding register address.
0x04: There is no write authorisation at this authorisation level (password).

1.3.4 Diagnostics

Command code: 0x08

This command can be used to check the Modbus function Command

from master:

The following data bytes are transmitted:
- Subfunction code MSB
- Subfunction code LSB
- 1 - 17 data bytes

Only the subfunction code 0000 is supported!

Answer from the fan:

The following data bytes are transmitted:
- Subfunction code MSB
- Subfunction code LSB
- 1 - 17 data bytes

Exception codes:

In the event of an error, only one data byte (exception code) is

transmitted Exception codes:

0x01: Subfunction code is not supported (≠ 0000)

1.3.5 Write Multiple Register

Command code: 0x10

This command can used to write to the contents of *several* holding registers. Command from the

master:

The following data bytes are transmitted:
- Holding register address MSB
- Holding Register Address LSB
- Number of addresses to be written MSB
- Number of addresses to be written LSB
- Byte count (number of addresses to be written * 2)
- Data to be written 1. holding register MSB
- Data to be written 1. holding register LSB

optional:
- Data to be written to the following holding registers (0..n) The

description of the holding registers follows later.

Answer from the fan:

4 data bytes are transmitted:
- Holding register address MSB
- Holding Register Address LSB
- Number of addresses to be written MSB
- Number of addresses to be written LSB

Exception codes:

In the event of an error, only one data byte (exception code) is

transmitted 0x02: No permissible holding register address.
0x03: - The maximum number of possible registers has been exceeded
i.e. more than 123 holding register data or 0 holding register data have been defined
- Byte count≠ 2 * Number of registers
- Number of data bytes≠ Byte Count
0x04: There is no write authorisation at this authorisation level (password).

Commands with addressing via serial number

These commands also use the serial number of the fans for addressing:
An identifier for the serial number of the fan (6 bytes) is transmitted in the data area of the commands. Only the fan that has the Modbus address and serial number specified in the command reacts and responds to the command.

The command with addressing via the serial number is derived from the standard command by inserting an identifier for the serial number after the command byte.

Serial number format:

ebm-papst assigns an individual serial number to each fan. The part that can be read by the customer consists of the production date and the serial number. This can also used for addressing. If, contrary to all expectations, problems should arise, please let us know.

Format: JJWW00XXXX

JJ Production year
WW Production calendar week 00
fixed value 00
XXXX : consecutive number

The first 4 characters contain the production date (year / calendar week).
At the start of each production week, the number XXXX starts from zero and is by 1 for each fan. Each character can take values from 0-9 and from A-Z. This results in a maximum encodable number of 36 characters per digit, i.e. $36^4 = 1\,679\,616$ devices / week

Example:
ebm-papst serial number: 09230012GY

Year of production:	09	(2009)
Calendar week:	23	(23)
consecutive no:	12GY	(49525)

The production year and calendar week are each coded as hex values in the command. For the consecutive no., each character is coded as an ASCII value.

In the commands with addressing via the serial number, the serial number must be specified as follows:

Byte no.	Meaning	Example	corresponds to Holding Register
Byte 0:	[address]		
Byte 1:	[Command code]		
Byte 2:	Year of production	0x09 (-> 2009)	E10E MSB
Byte 3:	Calendar week	0x17 (-> 23)	E10E LSB
Byte 4:	consecutive no. 1st character	0x31 (-> 1)	E10D MSB
Byte 5:	consecutive no. 2nd character	0x32 (-> 2)	E10D LSB
Byte 6:	consecutive no. 3rd character	0x47 (-> G)	E10C MSB
Byte 7:	consecutive no. 4th character	0x59 (-> Y)	E10C LSB

The sequence differs from the sequence in which the serial number is stored in the holding registers E10C - E10E (see 2.19)!

Broadcast - Addressing

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

Instead of the broadcast address / serial number 0x00, the own address and serial number are specified in the response.

Options for determining the serial number:

- Imprint on the type plate of the fan
- Reading the serial number from holding register E10C - E10E (see 2.19)
- Broadcast addressing: The fan responds with its serial number

1.3.5.1 Read Holding Register Addressed By Serial No.

Command code: 0x43

This command can be used to read the content of several holding registers. Holding registers are parameters to which there is both read and write access.

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

Command from the master:

10 data bytes are transmitted:

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

The description of the holding registers follows later. Response from

the fan:

The following data bytes are transmitted:

- Series No. Byte 1
- Series No. Byte 2
- Series No. Byte 3
- Series No. Byte 4
- Series No. Byte 5
- Series No. Byte 6
- Byte count (number of addresses to be read * 2)
- Data 1st Holding Register MSB
- Data 1st holding register LSB

optional:

- Data of the following holding registers (0..n)

Exception codes:

In the event of an error, only one data byte (exception code) is

transmitted 0x02: No permissible holding register address.
0x03: The maximum telegram length for the response (23 bytes) was exceeded
i.e. more than 6 holding registers or 0 holding registers were requested.

1.3.5.2 Read Input Register Addressed By Serial No.

Command code: 0x44

This command can used to read the content of several input registers. Input registers are parameters that can only be read.

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

Command from the master:

10 data bytes are transmitted:
- Series No. Byte 1
- Series No. Byte 2
- Series No. Byte 3
- Series No. Byte 4
- Series No. Byte 5
- Series No. Byte 6
- 1. input register address MSB
- 1. input register address LSB
- Number of addresses to be read MSB
- Number of addresses to be read LSB

The description of the input registers follows later.

Answer from the fan:

The following data bytes are transmitted:
- Series No. Byte 1
- Series No. Byte 2
- Series No. Byte 3
- Series No. Byte 4
- Series No. Byte 5
- Series No. Byte 6
- Byte count (number of addresses to be read * 2)
- Data 1st Holding Register MSB
- Data 1st holding register LSB

optional:
- Data of the following input registers (0..n)

Exception codes:

In the event of an error, only one data byte (exception code) is transmitted

0x02: The permissible range of the input registers 0xD000, 0xE201. ....0xE20B has been exceeded
0x03: The maximum telegram length for the response (23 bytes) was exceeded
i.e. more than 6 input registers or 0 input registers were requested.

1.3.5.3 Write Single Register Addressed By Serial No.

Command code: 0x46

This command can used to write the content of *a* holding register.

In deviation from "MODBUS over Serial Line Specification & Implementation guide V1.0", the fan also responds to a broadcast command (Modbus address = 0) if no broadcast identifier (0) has been used in the serial number. As two different fans cannot have the same serial number, it is impossible for several fans to respond to one command.

Command from the master:

10 data bytes are transmitted:
- Series No. Byte 1
- Series No. Byte 2
- Series No. Byte 3
- Series No. Byte 4
- Series No. Byte 5
- Series No. Byte 6
- Holding register address MSB
- Holding Register Address LSB
- Data to be written MSB
- Data to be written LSB

The description of the holding registers follows later.

Answer from the fan:

10 data bytes are transmitted:
- Series No. Byte 1
- Series No. Byte 2
- Series No. Byte 3
- Series No. Byte 4
- Series No. Byte 5
- Series No. Byte 6
- Holding register address MSB
- Holding Register Address LSB
- Data to be written MSB
- Data to be written LSB

In the response, the command is repeated by the master

Exception codes:
In the event of an error, only one data byte (exception code) is

transmitted Exception codes:

0x02: No permissible holding register address.
0x04: There is no write authorisation at this authorisation level (password).

1.3.5.4 Write Multiple Register Addressed By Serial No.

Command code: 0x50

This command can used to write to the contents of *several* holding registers. Command from the

master:

The following data bytes are transmitted:
- Series No. Byte 1
- Series No. Byte 2
- Series No. Byte 3
- Series No. Byte 4
- Series No. Byte 5
- Series No. Byte 6
- Holding register address MSB
- Holding Register Address LSB
- Number of addresses to be written MSB
- Number of addresses to be written LSB
- Byte count (number of addresses to be written * 2)
- Data to be written 1. holding register MSB
- Data to be written 1. holding register LSB

optional:
- Data to be written to the following holding registers (0..n) The

description of the holding registers follows later. Response from the

fan:

10 data bytes are transmitted:
- Series No. Byte 1
- Series No. Byte 2
- Series No. Byte 3
- Series No. Byte 4
- Series No. Byte 5
- Series No. Byte 6
- Holding register address MSB
- Holding Register Address LSB
- Number of addresses to be written MSB
- Number of addresses to be written LSB

exception codes:

In the event of an error, only one data byte (exception code) is transmitted

0x02: No permissible holding register address.
0x03: - The maximum number of possible registers has been exceeded
i.e. more than 123 holding register data or 0 holding register data have been defined
- Byte count≠ 2 * Number of registers
- Number of data bytes≠ Byte Count
0x04: There is no write authorisation at this authorisation level (password).

1.3.5.5 Applications

Initialisation of a system

All fans have the same address 1 ex works. In order to be able to address each fan individually, this address must be changed so that each fan has a different address when setting up a system consisting of several fans.

Conventional method (command 0x06 Write Single Register):
1. Switch on the first fan (all others remain switched off)
2. Changing the address via command 0x06 Write Single Register
3. Switching 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 the type plate)
1. Switch on all fans
2. Changing the addresses via command 0x46 Write Single Register Addressed By Serial No.

Procedure for unknown serial numbers:
1. Switch on all fans
2. Identifying the serial numbers via command 0x43 Read Holding Register Addressed By Serial No. or 0x44 Read Input Register Addressed By Serial No. in broadcast addressing with collision detection
3. If a serial number has been identified:
   Changing the Modbus address via command 0x46 Write Single Register Addressed By Serial No.

The advantage over the conventional method is that all fans can be switched on at the same time when a system is initialised. 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 for switching operations on the system.

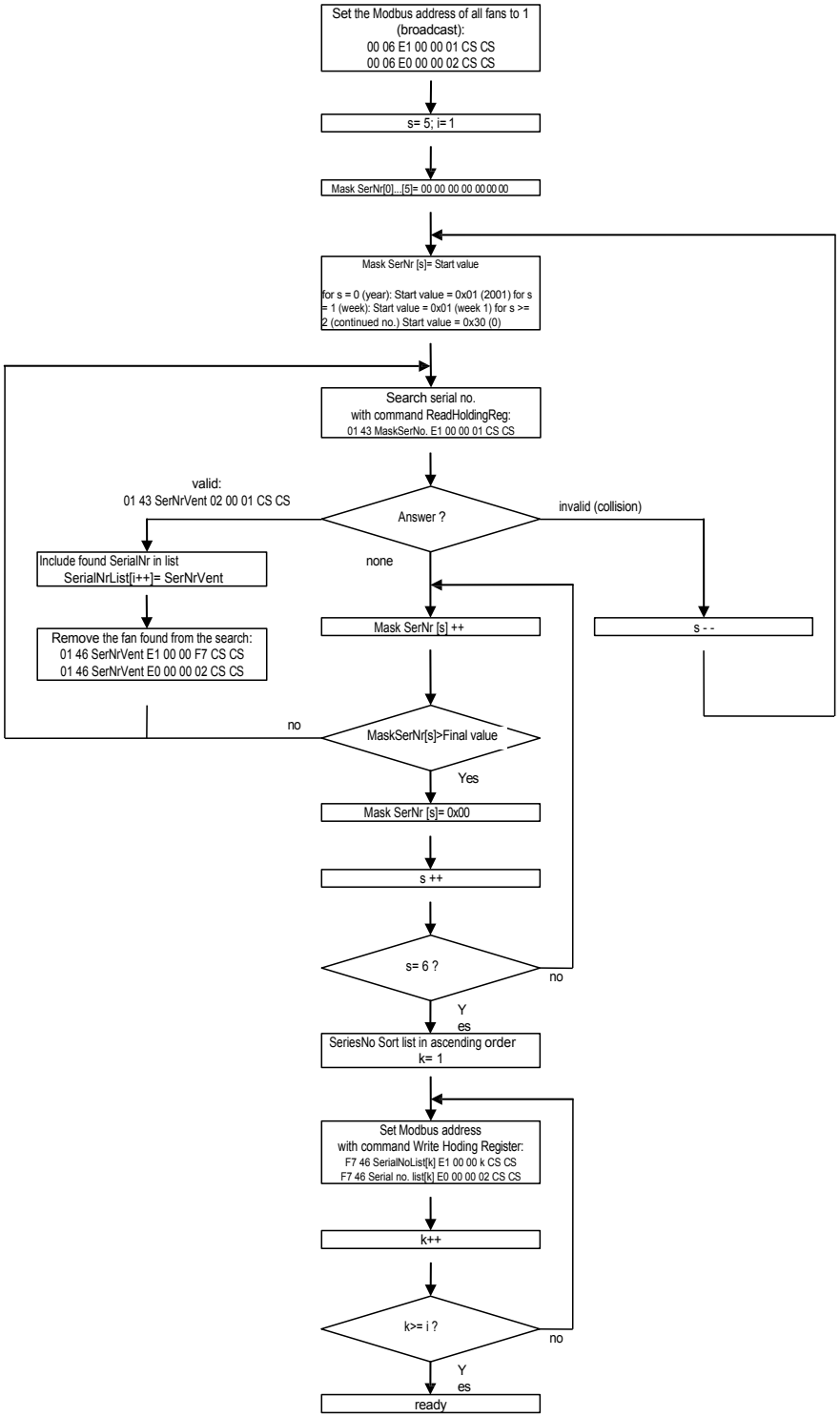
Implementation in the master device (e.g. PC):

For known serial numbers, the following procedure is recommended for implementation in a master device:

1. All serial numbers are made known to the master device by manual entry

2. A Modbus address is assigned to each fan via the serial number. This requires 2 broadcast commands per fan:
   a. Set holding register fan address (E100):
      00 46 SNr SNr SNr SNr SNr SNr SNr E1 00 00 Addr CS CS
      (CS= CRC checksum)
   b. Holding Register Reset (E000) to 0x02 Accept parameter: 00 46 SNr SNr SNr SNr SNr SNr SNr E0 00 00 02 CS CS
      SNr= Determined serial number of the fan; Adr= Assigned Modbus address

For unknown serial numbers, the following algorithm is recommended for implementation in a master device:



Functionality:

When setting up the system, the fans must be arranged in ascending order by serial number. This is the prerequisite for ensuring that the automatically assigned Modbus addresses can be easily assigned to the fans in the system.

Initially, all fans are to the Modbus address 0x01. This requires 2 broadcast commands:

- Set holding register fan address (E100) to 0x01: 00 06 E1 00 00 00 01 CS CS (CS = CRC checksum)
- Set Holding Register Reset (E000) to 0x02 Accept parameter: 00 06 E0 00 00 00 02 CS CS

All serial numbers of the system are determined in a loop:
The mask for serial number addressing is initially set to broadcast addressing (00 00 00 00 00 00 00). The last byte is set to the start value 0x30 (0).
The SerNr mask is therefore 00 00 00 00 00 00 30.
This screen is used to search for fans that have the value 0x30 (0) in the last digit of the serial number.
Any command can used to read a holding register or input register,
e.g. Read Holding Register Fan address: 01 43 00 00 00 00 00 00 30 E1 00 00 00 01 CS CS

There are several possibilities for the answer:

- There is a clear answer:
  The fan responds in the address field with its serial number (SerNrVent). This serial number is saved in a list.
  The fan is then blocked for further requests by setting the Modbus address to 0xF7 (247) is set.
  The same serial number must then be queried again, as it cannot be ruled out that another fan was also addressed but not respond because it had already recognised the start of the response from the other fan due to runtime differences.
- There is an invalid response due to the overlapping of responses from several fans: The mask for the serial number must then be further restricted by also setting the penultimate byte to the start value 0x30 (0). For the next enquiry, only fans with the value 0x30 (0) in the last two digits of the serial number will respond.
- there is no answer:
  then all serial numbers with the value 0x30 in the last digit can be excluded. The last byte of the mask is incremented to the value 0x31 (1). All fans with the value 0x31 (1) in the last digit of the serial number will now respond to the next query.

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

- If the response is valid, the serial number of the fan is saved in the list
- If the response is invalid, the serial number range further restricted by masking another byte, starting with the start value. The start value for the last 4 bytes is 0x30 (0). The first two bytes have the start value 0x01, as the year (2001) and calendar week (CW1) are coded here.
- If no answer is given, the relevant digit is increased by 1 until the final value is reached.
  The final value for the last 4 bytes is 0x5A (Z). For the first byte ((year), the end value is 0x63 (2099); for the second byte (CW), the end value is 0x35 (CW53).
  If the final value is reached, the relevant digit is addressed again via broadcast (0x00) and the next digit is by 1. If the addressing for the last byte reaches the end value 0x5A (Z), all serial numbers have been checked. The request for serial numbers can thus be ended.

All serial numbers found in the master are then sorted in ascending order. A Modbus address is assigned to each fan via the serial number. This requires 2 commands per fan:

- Set holding register fan address (E100): F7 46 SNr SNr SNr SNr SNr SNr E1 00 00 Adr CS CS
- Set Holding Register Reset (E000) to 0x02 Accept parameter: F7 46 SNr SNr SNr SNr SNr SNr E0 00 00 02 CS CS
  SNr= Determined serial number of the fan; Adr= Assigned Modbus address

Extension of the address space

A maximum of 247 fans can be addressed with standard Modbus commands.
The commands with addressing via the serial number offer the option of addressing any number of ebm-papst fans with a Modbus interface. In this case, the address of the fan does not consist of just one byte, but of 7 bytes (Modbus address + 6 bytes serial number). This address is set at the factory and cannot be changed.

The serial number in the master device (e.g. PC) must be known for addressing, as it is not possible to scan all serial numbers due to the large number. To determine the serial number, see above (initialisation of a system).

With an extended address space, only the commands with addressing via the serial number can be used. If standard commands are used, collisions will inevitably occur as several fans have the same Modbus address.

Determination of 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 identifier are specified with 0x00
- The connected fan responds with its Modbus address and serial number

1.3.6 Other commands

All other commands are not supported.
An unsupported command is always answered with the 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

areas are defined:

Address	Range
E000 ... E0FF	RAM
E100 ... E3FF	EEPROM

The following list provides an overview of all parameters.
In addition to the Modbus address and the name, the authorisation level required to write a parameter is shown, as well as the address of the memory location for the factory setting and customer setting (if available).

The function of the parameters is described in the following chapters

Modbus address	Designation	Letter ebm-papst	Customer letter	Letter end customer	Factory setting Address	Customer login Address
E000	Reset	x	x	x	-	-
E001	MODBUS setpoint	x	x	x	-	-
E002	password	x	x	x	-	-
E003	password	x	x	x	-	-
E004	password	x	x	x	-	-
E005	Factory setting Control	x	x *)	-	-	-
E006	Customer setting Control	x	x	x *)	-	-
E100	Fan address	x	x	x	E400	E600
E101	Save setpoint	x	x	x	E401	E601
E102	Setpoint last saved	x	x	x	E402	E602
E103	Reference speed	x	x	x	E403	E603
E104	Reference speed Max	x			E404	E604
E105	Setpoint reduction	x	x	x	E405	E605
E106	Setpoint level 3	x	x	x	E406	E606
E107	Direction of rotation setpoint stage3 inverted	x	x	x	E407	E607
E108	Emergency running Running direction	x	x		E408	E608
E109	Emergency running on/off	x	x		E409	E609
E10A	Emergency setpoint	x	x		E40A	E60A
E10B	Emergency run time delay	x	x		E40B	E60B
E10C	Serial number HH	x			E40C	E60C
E10D	Serial number LH	x			E40D	E60D
E10E	Serial number LL	x			E40E	E60E
E10F	Setpoint level 1	x	x	x	E40F	E60F
E110	Setpoint level 2	x	x	x	E410	E610
E111	Reserved	x	x	x	E411	E611
E112	Reserved	x	x	x	E412	E612
E113	Reserved	x	x	x	E413	E613

Modbus address	Designation	Letter ebm-papst	Customer letter	Letter end customer	Factory setting Address	Customer login Address
E114	Function1	x	x	x	E414	E614
E115	Operating mode	x	x	x	E415	E615
E116	Number of start-up attempts	x	x	x	E416	E616
E117	KP controller	x	x	x	E417	E617
E118	AI controller	x	x	x	E418	E618
E119	Direction of rotation default	x	x	x	E419	E619
E121	Minimum dowry rate	x	x	x	E421	E621
E122	Maximum dowry rate	x	x		E422	E622
E123	Motor stop enable	x	x	x	E423	E623
E127	Operating hours 1 (high)	x			E427	E627
E128	Operating hours 2 (low)	x			E428	E628
E129	Run monitoring tolerance	x	x		E429	E629
E12A	Run monitoring Time offset	x	x		E42A	E62A
E12B	Fan type	x			E42B	E62B
E12C	Fan type	x			E42C	E62C
E12D	Fan type	x			E42D	E62D
E12E	Fan type	x			E42E	E62E
E12F	Fan type	x			E42F	E62F
E130	Fan type	x			E430	E630
E133	Reserved	x			E433	E633
E134	Reserved	x			E434	E634
E135	Reserved	x			E435	E635
E136	Reserved	x			E436	E636
E138	Setpoint ramp slope	x	x	x	E438	E638
E139	Service time	x	x		E439	E639
E13A - E149	Customer data	x	x		E43A - E449	E63A - E649
E14A	KP Speed limiter	x			E44A	E64A
E14B	AI speed limiter	x			E44B	E64B
E14C	KP Power limiter	x			E44C	E64C
E14D	AI power limiter	x			E44D	E64D
E14E	Speed limiting value	x			E44E	E64E
E14F	Power limitation value	x			E44F	E64F
E150	Power reference value	x			E450	E650
E151	Enable limiter	x			E451	E651
E152	Characteristic curve analogue value motor start	x	x	x	E452	E652
E153	Characteristic curve analogue value motor stop	x	x	x	E453	E653
E154	Characteristic curve analogue value motor max	x	x	x	E454	E654
E155	Setpoint speed Minimum value	x	x	x	E455	E655
E156	Setpoint Speed Maximum value	x	x	x	E456	E656
E157	Setpoint power Minimum value	x	x	x	E457	E657
E158	Setpoint Power Maximum value	x	x	x	E458	E658
E159	Setpoint Controlled Minimum value	x	x	x	E459	E659
E15A	Setpoint Controlled Maximum value	x	x	x	E45A	E65A
E15B	Level 1 analogue control voltage	x	x	x	E45B	E65B
E15C	Stage 2 analogue control voltage	x	x	x	E45C	E65C
E15D	Level 3 analogue control voltage	x	x	x	E45D	E65D
E15E	Stages analogue hysteresis	x	x	x	E45E	E65E

*) only partially

Coding of the parameters:

Unless otherwise specified, parameters are coded in "big endian" format, i.e. the byte with the most significant bits comes first. This applies in particular to parameters that comprise several holding registers.

2.2 Reset

Address : E000
Write authorisation : ebm-papst, customer, end

customer Coding:

MSB	0	0	0	0	0	0	0	0
LSB	0	0	0	0	0	Error	Param.	Reset

A bit triggers the following action in the fan if it is set:

Reset Software reset ("Accept parameters") Software is restarted.

Param : All parameters are transferred from the EEPROM to the RAM.
This bit must set to make changed parameters valid.

Error Errors are reset, e.g. max. number of start-up attempts reached. After the

action has been carried out, the bit is reset automatically by the fan.

2.3 Default setpoint

Address : E001
Write authorisation : ebm-papst, customer, end customer

The default setpoint parameter can used to specify a setpoint for each operating mode via Modbus. The prerequisite is that MODBUS is specified as "Function 1". Otherwise, the parameter has no function.

If the "Save" function is activated in the "Save setpoint" parameter (see 2.8 Save setpoint), the value in the Setpoint last saved parameter is saved each time the preset setpoint is written (see 2.9 Setpoint last saved

The "Save setpoint" function must not be activated if write access to the preset setpoint takes place cyclically at short intervals!
Otherwise there is a risk that the memory will be permanently destroyed.

When this function is activated, the maximum number of write accesses to the preset setpoint is 100,000 over the entire service life of the fan.
For more details, see the fan data sheet.

After a reset, the motor then starts again with this value. Coding:

a) in speed control mode

The setpoint designates a speed:

Default ***sol*** *value* 1/ min

The value zero means motor standstill

b) in power control mode

The setpoint designates a performance:

Default setpoint [W]= Data bytes

The value zero means motor standstill

c) in control mode

The default setpoint denotes a level of control:

$$Target\ \textbf{\textit{sol}}\ value\ \% \quad = \quad \frac{Data\ bytes}{65536} 100\%$$

The value zero means motor standstill

2.4 password

Address E002 - E004 Write authorisation
: ebm-papst, customer, end
customer

Coding:

password = Data bytes

To unauthorised writing to certain parameters, these are only written to if the correct password has been entered for the required authorisation at this point.
a user forgets to reset the password, it is automatically to 0x000000000000 after 4 minutes of inactivity.

When reading the "Password" parameter, the value 0 is always output to prevent users with lower authorisation from gaining knowledge of a higher level password.

2.5 Factory setting Control

Address : E005
Write authorisation : ebm-papst, customer (partial)

Coding:

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

Setting the D -> W bit causes all parameters in the data range (E100..E149) to be copied to the factory setting (range E400..E449).
The "ebm-papst" authorisation is required to set this bit!

Setting the W -> D bit causes all parameters of the factory setting (range E400..E449) to be copied to the data range (E100..E149).
The "Customer" authorisation is sufficient for setting this bit.
Once the copying process is complete, the bit is automatically reset by the fan.

2.6 Customer setting Control

Address : E006
Write authorisation : ebm-papst, customer, end customer

(partial) Coding:

MSB	0	0	0	0	0	0	0	0
LSB	0	0	0	0	0	0	D -> K	K -> D

Setting the D -> K bit causes all parameters of the data range (E100..E149) to be copied to the customer setting (range E600..E649). At least "Customer" authorisation is required to set this bit!

Setting the K -> D bit causes all parameters of the customer setting (range E600..E649) to be copied to the data range (E100..E149).
The "End customer" authorisation is sufficient for setting this bit.

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

2.7 Fan address

Address : E100
Write authorisation : ebm-papst, customer, end

customer When addressing via the MODBUS register

(E100):

Coding:

$$Fan\ address = Data\ byte\ (LSB)$$

The MSB is irrelevant! Permissible

value range: 1...247 Automatic

limitation:

Value> 247 results in 1
Value= 0 results in 1

2.8 Save setpoint

Address : E101
Write authorisation : ebm-papst, customer, end

customer Coding:

This parameter specifies whether an incoming preset setpoint (E001) is also saved in the EEPROM under setpoint last saved (E102) or not.

Value	Function
0	Setpoint is not saved After reset, the fan stops
1	Setpoint is saved in the EEPROM After reset, the fan runs with the saved setpoint value

The MSB has no meaning! Permissible

value range: 0..1

The function must not be activated if a write access to the preset setpoint (E001, see 2.3 Preset setpoint) takes place cyclically at short intervals! Otherwise there is a risk that the memory will be permanently destroyed.

When this function is activated, the maximum number of write accesses to the preset setpoint is 100,000 over the entire service life of the fan.
For more details, see the fan data sheet.

2.9 Setpoint last saved

Address : E102
Write authorisation : ebm-papst, customer, end customer

The parameter is only valid if the "Save setpoint" function is activated (see 2.8 Save setpoint). Otherwise, the parameter has no function.

The parameter contains the last saved setpoint value. If the "Save setpoint" option is active, each "Default setpoint" is also saved in this register and used as the default setpoint.

If "Save setpoint" is active, the fan with "Saved setpoint" as the setpoint after starting. Note: To change this value,

"Save setpoint" should be activated and a new setpoint set.
"Default setpoint" can be written. Coding: as

E001 ("Default setpoint")

2.10 Reference speed

Address : E103
Write authorisation : ebm-papst, customer, end customer

This information is only valid for "Speed control" (E115 operating mode).
The specified value is the maximum speed that can be set for the default setpoint E001. The value is limited to the maximum reference speed (E104 Reference speed max).

Coding:

$$Reference\ speed\ 1/\min$$

Permissible value range: [0 ... Reference speed max]

2.11 Reference speed max

Address : E104
Write authorisation : ebm-papst

This specification is only valid for "Speed control" operating mode (E115 Operating mode). The specified value limits the specification in (E103 Reference speed).

Coding:

$$\text{Reference } speedMax1/\min$$

2.12 Setpoint reduction

Address : E105
Write authorisation : ebm-papst, customer, end customer

The MODBUS setpoint can be reduced by this value. The emergency operation value (E10A emergency operation setpoint) is not affected by this.
The reduction is used for speed control, power control and control. Scaling: 0-255 = 0-100%

Coding: $= \frac{data\ byte}{256} \cdot MODBUS\ setpoint(E001)$

One possible application would ~~be a night~~ setback to X%, for example.

Permissible value range: [0 ... 255]

2.13 Setpoint level 3

Address : E106
Write authorisation : ebm-papst, customer, end customer

Setpoint for (E114 function 1) "STUFOSTOP, STUFMSTOP".
With (E114 function 1) "MODBUS", this setpoint level has no effect. Coding: As E001

Default setpoint

2.14 Direction of rotation stage3 inverted

Address : E107
Write authorisation : ebm-papst, customer, end customer

Direction of rotation for setpoint stage 3 for (E114 function 1) "STUFMSTOP" or "STUFOSTOP".
With (E114 function 1) "MODBUS" and "Analogue linear", this direction of rotation specification has no effect.

Coding:

Value	Running direction
0	Direction of rotation as in "Direction of rotation default" (E119)
1	Direction of rotation inverted as in "Direction of rotation default" (E119)

The MSB has no meaning! Permissible

value range: 0..1

2.15 Running direction Emergency running

Address : E108
Write authorisation : ebm-papst, customer

If the emergency function is activated and an interruption in the MODBUS connection is detected, the direction of rotation is set to the value defined here. The setting of the "Default direction of rotation" parameter (see 2.29 Default direction of rotation) is then not relevant.

Coding:

Value	Running direction Emergency running
0	Preferred direction of rotation
1	Counter rotation direction
2	Maintain direction of rotation (straight active running direction is retained)

The MSB is ! Permitted value range:

0..2

2.16 Emergency function on / off

Address : E109
Write authorisation : ebm-papst, customer

This parameter specifies whether the emergency function is active or inactive.

Emergency operation can only activated for setpoint source (HR E114) MODBUS. Emergency operation is inactive for other setpoint sources.

Coding:

Value	Emergency function
0	inactive
1	active

The MSB is ! Permitted value range:

0..1

2.17 Emergency setpoint

Address E10A
Write authorisation : ebm-papst, customer

If the failure of the setpoint signal is recognised, the fan automatically switches to the emergency run setpoint specified here.

Coding:

As E001 Default setpoint, depending on the active operating mode.

2.18 Time delay for emergency operation

Address E10B
Write authorisation : ebm-papst, customer

If no command is sent to the fan for the time specified here, the fan automatically switches to the emergency run setpoint, provided the emergency run function is activated.

Coding:

$$\textit{Time delay for emergency operation s} = \textit{Data byte} \cdot 1s$$

Permissible value range: 0..65535

2.19 Serial number and production date of fan

Address Serial number E10C / E10D
Address Production date E10E Write
authorisation : ebm-papst

ebm-papst assigns an individual serial number to each fan. The part that can be read by the customer consists of the production date and the serial number. This can also used for addressing. If, contrary to all expectations, problems should arise, please let us know.

Format: JJWW00XXXX

JJ Production year
WW Production calendar week 00
fixed value 00
XXXX : consecutive number

The first 4 characters contain the production date (year / calendar week).
At the start of each production week, the number XXXX starts from zero and is by 1 for each fan. Each character can take values from 0-9 and from A-Z. This results in a maximum encodable number of 36 characters per digit, i.e. $36^4 = 1\,679\,616$ devices / week

a) Serial number (E10C/ E10D)

Only the XXXX part (consecutive number) is in the "Serial number" parameter. Each character is encoded as an ASCII value.

Coding:

Serial number ASCII = Data bytes

b) Production date (E10E)

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

Coding:

Year of production = Data byte(MSB)

Production week = Data byte(LSB)

Example:

ebm-papst serial number: **09230012GY**
-> Contents E10C: 0x4759 (GY)
Contents E10D: 0x3132 (12)
Contents E10E: 0x0917 (09/23)

2.20 Setpoint level 1

Address E10F
Write authorisation : ebm-papst, customer, end customer

Setpoint for (E114 function 1) "STUFOSTOP, STUFMSTOP".
With (E114 function 1) "MODBUS", this setpoint level has no effect. Coding: As E001

Default setpoint

2.21 Setpoint level 2

Address : E110
Write authorisation : ebm-papst, customer, end customer

Setpoint for (E114 function 1) "STUFOSTOP, STUFMSTOP".
With (E114 function 1) "MODBUS", this setpoint level has no effect. Coding: As E001

Default setpoint

2.22 Function 1 (setpoint source)

Address : E114
Write authorisation : ebm-papst, customer, end customer

Selection of the setpoint specification.
Setpoint from MODBUS or via analogue value.

Value (decimal)	Name Function 1
38	MODBUS
0	STUFOSTOP
1	Analogue linear
2	STUFMSTOP
Default	MODBUS

2.23 Operating mode

Address : E115
Write authorisation : ebm-papst, customer, end

customer Coding:

Value	Operating mode
0	Control system
1	Speed control
2	Power control
Default	Control system

The MSB has no meaning! Permissible

value range: 0..1

2.24 Number of start-up attempts

Address : E116
Write authorisation : ebm-papst, customer, end

customer Coding:

Value	
255	Unlimited start-up attempts
1 - 254	According to the value

If the number of start-up attempts carried out exceeds the value set here, the following is displayed in addition to the "Motor blocked" error signalling the "Number of start attempts reached" error (see 3.11 Error status). The MSB

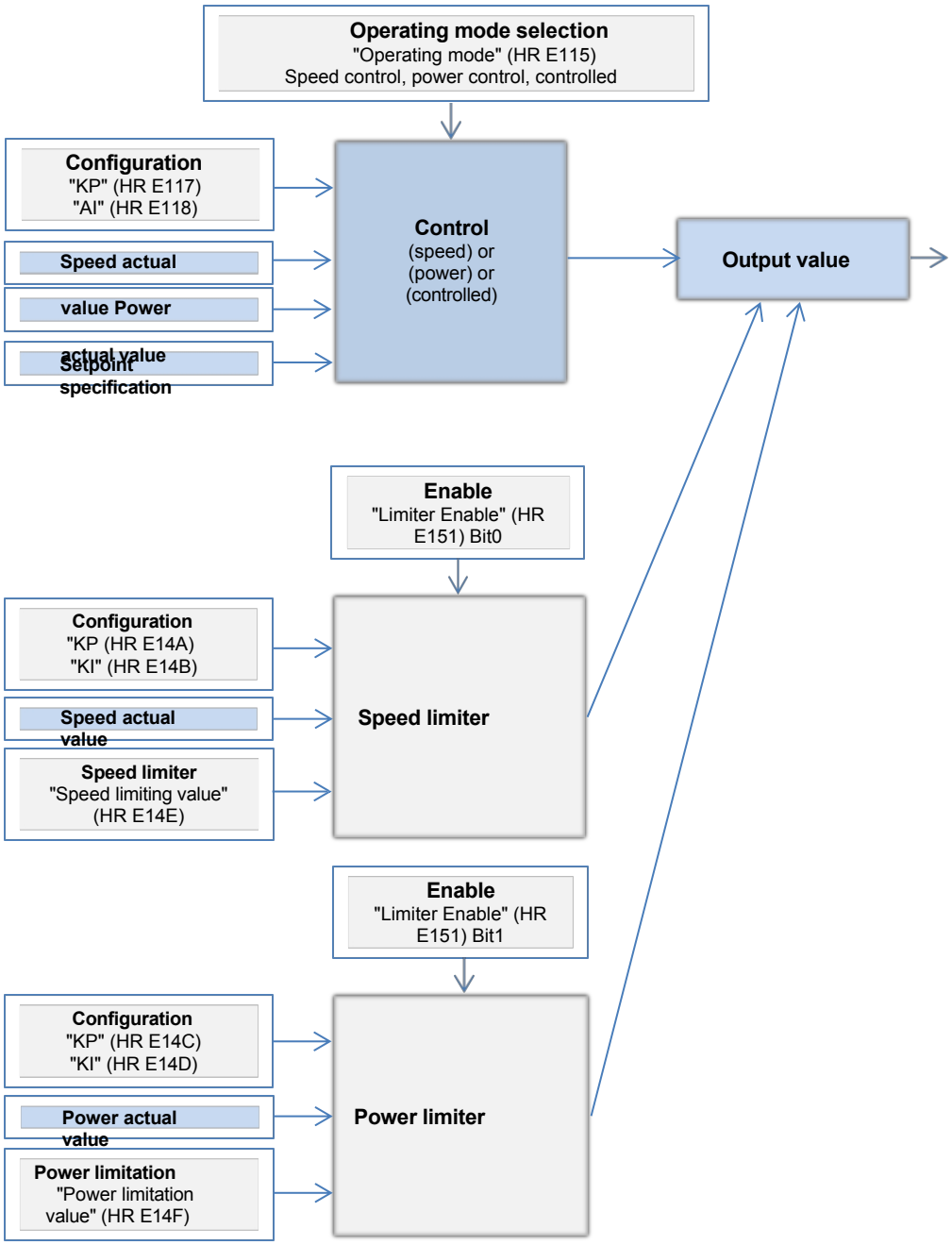
has no meaning!

Permissible value range: 1..255

2.25 Controller / limiter parameters

Controller:
Speed control or power control can be selected for control (E115 operating mode). The controller can also be completely deactivated in the "Control" operating mode.

Limiter:
There is a separate limiter for speed limitation and power limitation. These run independently of the operating mode (E115 operating mode). The speed limiter can also be activated during speed control.
The limiters have the parameters (E14A - E14D) to adjust the properties. The value to be limited is set with (E14E - E14F).
Each limiter can activated or deactivated separately with (E151).



2.25.1 Controller parameters KI/KP

Address P - Factor controller	: E117
Address I - Factor controller	: E118
Address P - Speed limiter factor	E14A : E14A
Address I - Speed limiter factor	: E14B
Address P - Power limiter factor	E14C : E14C
Address I - Power limiter factor	E14D : E14D
Write authorisation in each case	: ebm-papst

Coding:

Not speaking.

It is recommended to use the default values as a guide. Permissible

value range: 0..65535

2.25.2 Speed limiting value

Address Speed limiting value	E14E
Write authorisation	: ebm-papst

coding:

Limiting speed [1/min]= Data byte

Permissible value range: 0..reference speed

2.25.3 Power limitation value

Address Power limitation value	E14F
Write authorisation	: ebm-papst

coding:

Limiting power [W]= Data byte

Permissible value range: 0..reference power

2.25.4 Enable limiter

Address limiter Enable : E151
Write authorisation : ebm-papst

There is one controller and two limiters.

The limiters can be activated/deactivated individually.

LSB								
	0	0	0	0	0	0	PWR_EN	RPM_EN

If a bit is set, the corresponding limiter is active:

RPM_EN: Speed limitation enable
PWR_EN: Power limitation enable

2.26 Direction of rotation default

Address : E119
Write authorisation : ebm-papst, customer, end

customer Coding:

Value	Running direction
0	Preferred direction of rotation
1	Counter rotation direction

The MSB has no meaning! Permissible

value range: 0..1

2.27 Minimum dowry rate

Address : E121
Write authorisation : ebm-papst, customer, end customer

This value can used to limit the PWM setpoint to a minimum. Regardless of the active operating mode (E115 operating mode), this value is not undershot as a default value.
The limitation is applied to the setpoint current specification if, for example, a speed controller would output a smaller value than specified via the data byte.

The limitation is not applied directly to the PWM for the output stages.

Coding:

$$\min . \underset{\%}{Degree\ of\ dowry} = \frac{Data\ byte}{65536} 100\%$$

Permissible value range: 0..65535

2.28 Maximum dowry rate

Address : E122
Write authorisation : ebm-papst, customer

Coding:

$$Default\ \mathbf{sol}\ value\ \% \quad = \quad \frac{Data\ bytes}{65536} \cdot 100\%$$

Limitation of the preset setpoint (E001 Preset setpoint).

Only valid in operating mode (E115 operating mode) "Control", otherwise this value has no effect. Value range:

0..65535

2.29 Motor stop enable

Address : E123
Write authorisation : ebm-papst, customer, end customer

The motor stop enable is used for all setpoint sources (set in HR E114). Coding:

Value	Motor Stop
0	Motor always runs (even with setpoint= 0)
1	Motor stops at setpoint= 0

The MSB has no meaning! Value

range: 0..1

2.30 Operating hours counter

Address E127 (high) - E128(low)

Write authorisation : ebm-papst

Coding:

$$Operating\ time\ [h] = Data\ bytes * 30min$$

The value is limited to 16777215

For every 30 minutes that the fan is energised, the operating hours counter is increased by 1. A maximum of

8388607 hours (approx. 939 years) can be achieved with 16777215.
If the operating hours counter overflows, it is no longer written to, i.e. it remains at 16777215.

2.31 Run monitoring tolerance

Address : E129
Write authorisation : ebm-papst, customer

Tolerance band for the currently applied setpoint (E208 Setpoint currently applied). Only applies to "Speed control" operating mode (E115 Operating mode).

Scaling: Deviation setpoint +- X 1/min

$$X = \frac{Data\ \text{byte}^* Setpoint}{256}$$

Data byte 0-255= 0-100%

The MSB is irrelevant!

Permissible value range: 5..200. This corresponds to 2% - 78%

2.32 Run monitoring Time offset

Address E12A
Write authorisation : ebm-papst, customer

The speed value (E205 Speed actual value) must violate the tolerance band (E129 Run monitoring tolerance) around the setpoint (E208 Setpoint currently applied) for the time set here in order to generate a warning.

$$Time\ offset\ s = Data\ bytes$$

Value range: 10..65535

2.33 Fan type

Address E12B - E130
Write authorisation : ebm-papst

Coding:
The fan type is stored here in ASCII characters E12B
contains the first two characters.
E130 contains the last two characters.

2.34 Setpoint ramp slope

Address : E138
Write authorisation : ebm-papst, customer, end

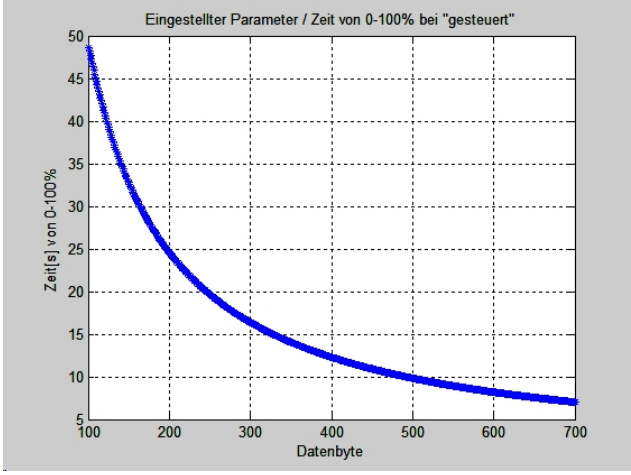
customer Coding:

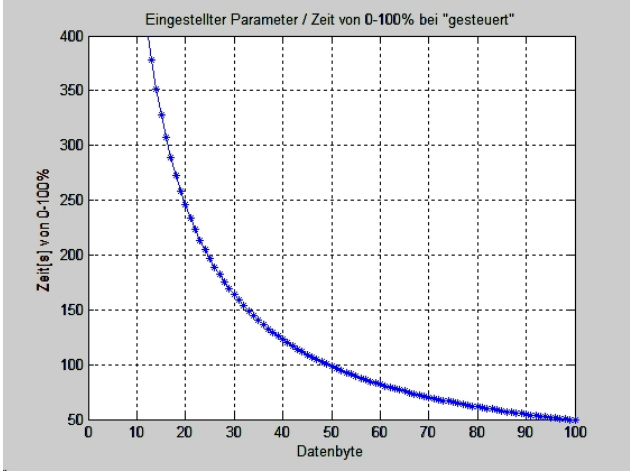
Control operating mode:

$$Datenbytes = \frac{\text{Änderung } [\%] * 32767 * 0.15s}{100\% * Zeit[s]}$$

Data bytes= 0 deactivates the setpoint ramp.

Data bytes Value set in HR E138 for the speed of the setpoint change. Change [%] By how much the setpoint should change
Time [s] How long this change should take





Example:

HR E138 (setpoint slope)= **50**

The time required to realise a setpoint change of 50% to be calculated. Calculated time (s) = ((50% * 32767) / (100% * 50)) * 0.15s = 49.15s

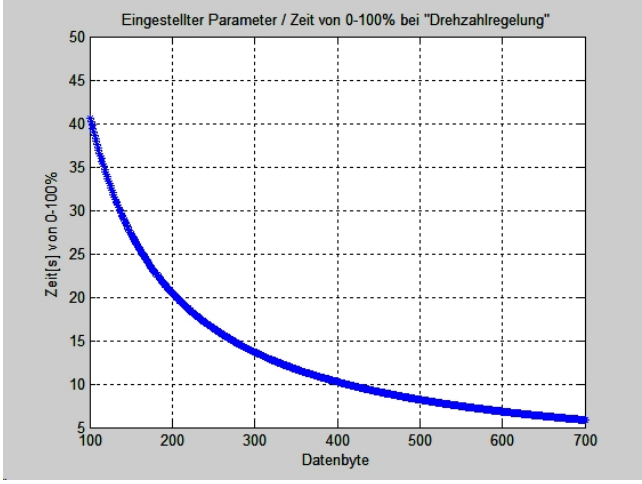
=> The setpoint ramp requires 49.15 seconds until the setpoint changes by 50%. Value range: 0..32767

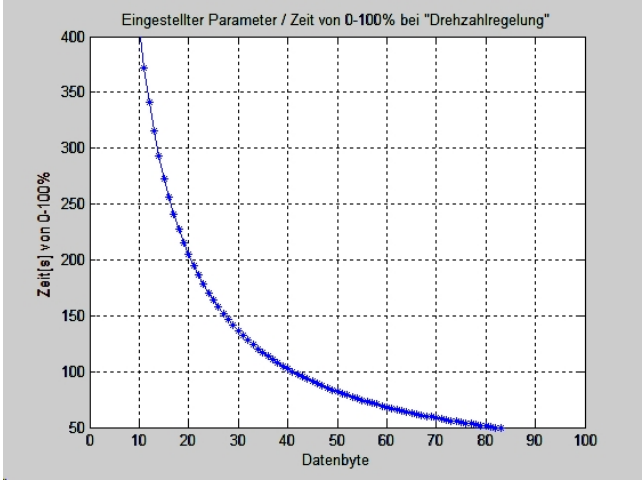
Speed control operating mode:

$$Datenbyte = \frac{Änderung\ [\frac{}{min}] * 27305 * 0.15s}{Bezugsdrehzahl\ [\frac{1}{min}] * Zeit[s]}$$

Data bytes= 0 deactivates the setpoint ramp.

Data byte Value set in HR E138 for the speed of the setpoint change. Change [1/min] By how much the setpoint should change
Time [s] How long this change should take





Example:

HR E114 (function1)= Analogue linear
HR E138 (setpoint slope) = **40**
HR E103 (reference speed)= 3000 rpm

The time required to realise a setpoint change of 500 rpm to be calculated.

Calculated time (s)= ((500 rpm * 27305) / (3000 rpm * 40)) * 0.15s= 17.06s

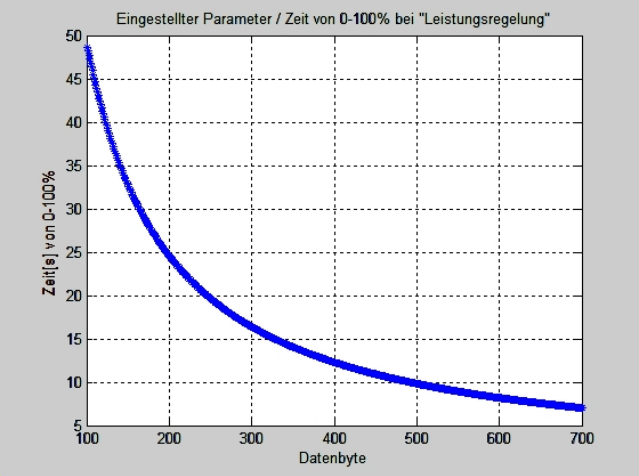
=> The setpoint ramp requires 17.06 seconds until the setpoint by 500 rpm. Value range: 0..32767

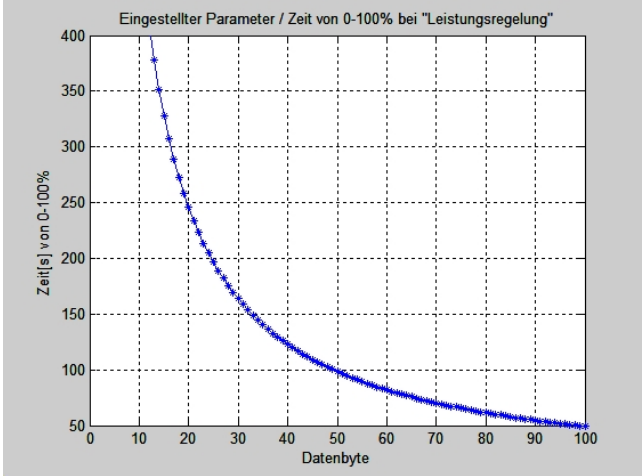
Power control operating mode:

$$Datenbyte = \frac{Änderung g\ [\ ]W * (32767) * 0.15s}{Bezugsleistung\ [\ ]W * Zeit[s]}$$

Data bytes= 0 deactivates the setpoint ramp.

Data byte Value set in HR E138 for the speed of the setpoint change. Change [W] By how much the setpoint should change
Time [s] How long this change should take





Example:

HR E114 (function1)= Analogue linear
HR E138 (setpoint slope) = **15**
HR E150 (reference power)= 1500 Watt

The time required to realise a setpoint change of 100 watts to be calculated. Calculated time (s) = ((100 watts *

32767) / (1500 watts * 15)) * 0.15s = 21.84s

=> The setpoint ramp requires 21.84 seconds until the setpoint changes by 100 watts. Value range:

0..32767

2.35 Service time

Address : E139
Write authorisation : ebm-papst, customer

If the operating hours counter of the fan exceeds this value, a "Service due" warning is displayed in the warning register (2.52 Warnings). A value of 0 deactivates the function.

Coding:

Service time [h= *Data bytes*

Value range: 0..65535

2.36 Customer data

Address E13A - E149 Write
authorisation : ebm-papst,
customer

A total of 16 parameters (16 bits each) are available to the customer in this area. Any values can be stored here.

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

2.37 Power reference value

Address : E150
Write authorisation : ebm-papst

The power reference value is the maximum DC link voltage multiplied by the maximum measurable DC link current.

P_max= Uzk_max * Izk_max.

The value is therefore dependent on the shunt value and the voltage divider for the DC link voltage. Coding:

Power reference value [W]= Data bytes

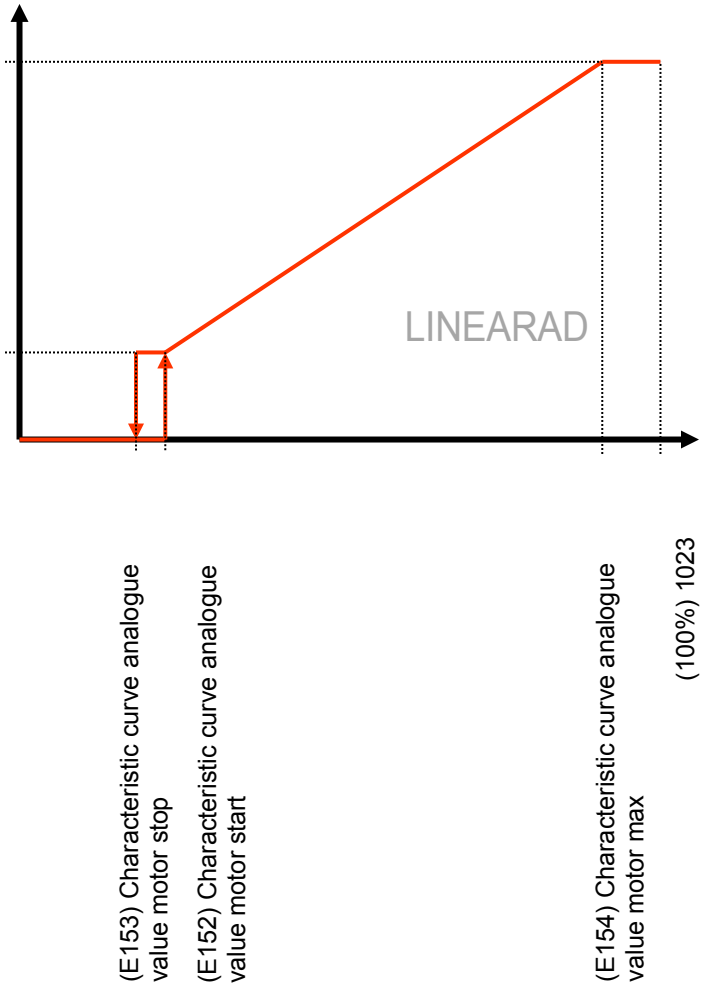
Value range: 1..65535

2.38 Analogue value Input characteristic Linear

Characteristic curve analogue value motor start	: E152	ebm-papst, customer, end customer
Characteristic curve analogue value motor stop	: E153	ebm-papst, customer, end customer
Characteristic curve analogue value motor max	: E154	ebm-papst, customer, end customer
Setpoint speed Minimum value	: E155	ebm-papst, customer, end customer
Setpoint Speed Maximum value	: E156	ebm-papst, customer, end customer
Setpoint power Minimum value	: E157	ebm-papst, customer, end customer
Setpoint Power Maximum value	: E158	ebm-papst, customer, end customer
Setpoint controlled Minimum value	: E159	ebm-papst, customer, end customer
Setpoint controlled Maximum value	E15A	ebm-papst, customer, end customer

(E156) Setpoint speed maximum value
(E158) Setpoint power maximum value
(E15A) Setpoint controlled maximum value

(E155) Setpoint speed minimum value
(E157) Setpoint power minimum value
(E159) Setpoint controlled minimum value



The setpoint max/min values are selected depending on the operating mode (E115).

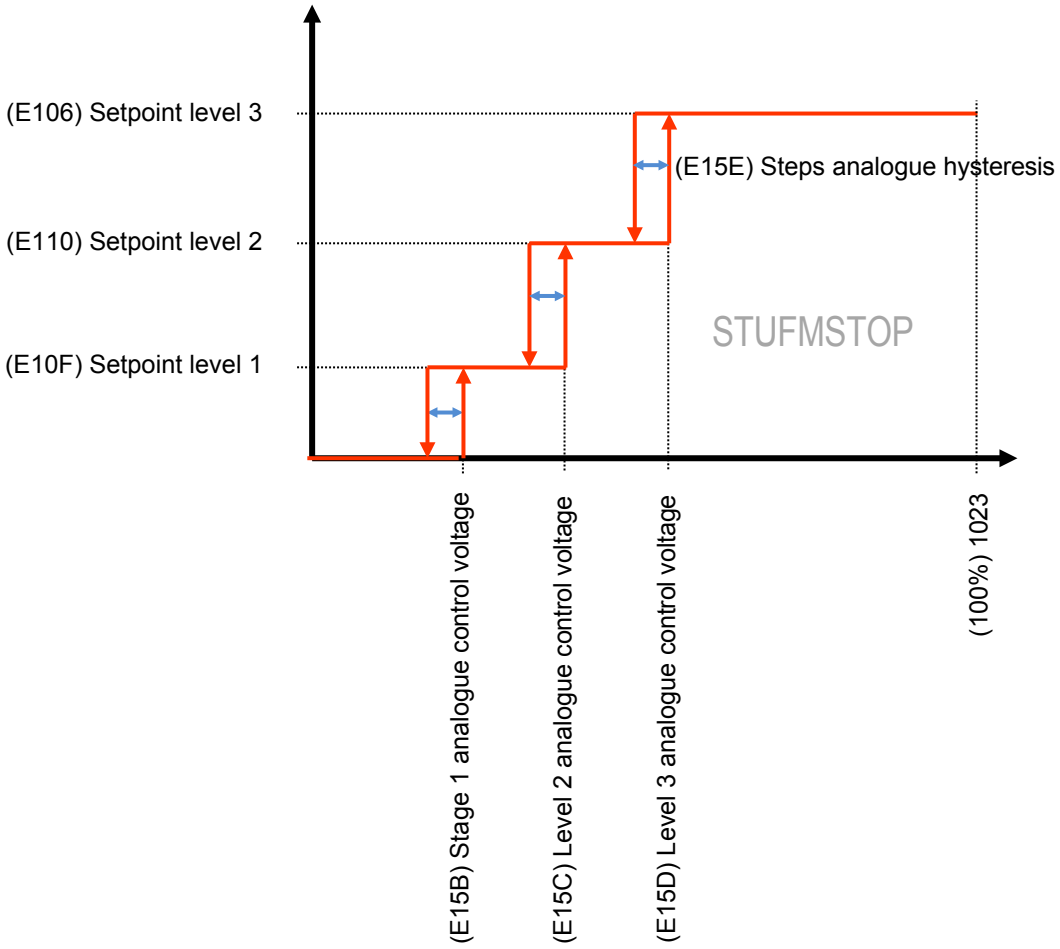
Coding:

E155, E156	*Speed [rpm]= Data bytes*	Value range: 1...reference speed
E157, E158	*Power [Watt]= Data bytes*	Value range: 1...reference power
E159, E15A	*Controlled= Data bytes*	Value range: 1...65535
E152 ... E154	*Analogue value [%]= Data bytes * 100% / 1023*	Value range: 1...1023

2.39 Analogue value 3-stage STUFMSTOP

(stepped with stop)

Characteristic curve analogue value motor start	E15B	ebm-papst, customer, end customer
Characteristic curve analogue value motor stop	E15C	ebm-papst, customer, end customer
Characteristic curve analogue value motor max	E15D : E15D	ebm-papst, customer, end customer
Stages analogue hysteresis	E15E : E15E	ebm-papst, customer, end customer



The setpoint min/max values are selected depending on the operating mode (E115).

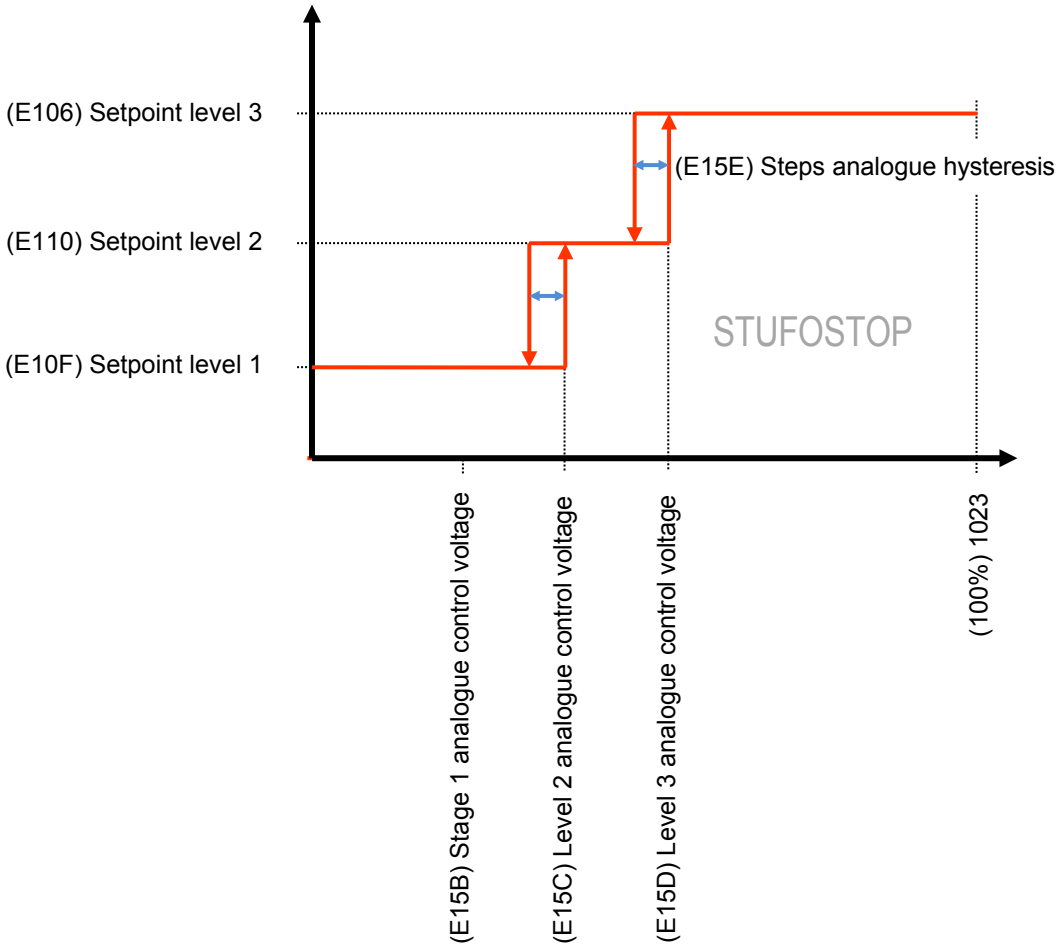
Coding:

E106, E110, E10F	*Same as default setpoint (E001)*	
E15B ... E15D	*Analogue value [%]= (data bytes * 4) * 100% / 1023*	Value range: 1...255
E15E	*Analogue value [%]= Data bytes * 100% / 1023*	Value range: 0...255 (0-25%)

2.40 Analogue value 3-stage STUFOSTOP

(stepped without stop)

Characteristic curve analogue value motor start	E15B	ebm-papst, customer, end customer
Characteristic curve analogue value motor stop	E15C	ebm-papst, customer, end customer
Characteristic curve analogue value motor max	E15D : E15D	ebm-papst, customer, end customer
Stages analogue hysteresis	E15E : E15E	ebm-papst, customer, end customer



The setpoint min/max values are selected depending on the operating mode (E115).

Coding:

E106, E110, E10F	*Same as default setpoint (E001)*	
E15B ... E15D	*Analogue value [%]= (data bytes * 4) * 100% / 1023*	Value range: 1...255
E15	*Analogue value [%]= Data bytes * 100% / 1023*	Value range: 0...255 (0-25%)

Input register

2.41 Overview

There is read-only access to input registers.

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

Modbus address	Designation
D000	Identification
E201	Max. Number of bytes
E202	SW Name Low
E203	SW Name High
E204	SW version
E205	Speed actual value
E206	Direction of rotation
E207	Degree of dowry (current)
E208	Setpoint currently applied
E209	Error status
E20A	Warnings
E20B	Version Parameter set
E20C	Power actual value
E20D	DB Index

2.42 Identification

Address : D000

The identification indicates the type of electronics or protocol . Coding:

Value	Device	Specification version
00 01	ebm-papst 84 / 112 / 150 series	1.02 *)
00 02	ebm-papst 84 / 112 / 150 series	2.01, 3.00 - 3.01 *)
00 06	ebm-papst 84 / 112 / 150 series	3.02 *)
00 07	ebm-papst 84 / 112 / 150 series	4.00 *)
00 08	ebm-papst 84 / 112 / 150 / 200 series	5.00 *)
00 0A	ebm-papst 84 / 112 / 150 / 200 Lite series	5.00 *)
00 0B	ebm-papst 84 / 112 / 150 / 200 Lite series+ Plug-on module	5.00 *)
00 0C	Special applications for ebm-papst 84/ 112/ 150/ 200 series	5.02 *)
00 0D	ebm-papst 84 / 112 / 150 / 200 Lite series	5.01 Lite *)
0A 00	ESM	1.00 *)
0A 10	ESL	1.00

*) Devices with identification≠ 0x0A10 do not comply with this specification! In this case, the corresponding document must be used!

2.43 Maximum number of bytes

Address : E201

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

Coding:

Max. Number of bytes = *Data bytes*

2.44 Software Name Low

Address : E202

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

Coding:

Display in HEX !

$$Software\ NameLowWord = Data\ bytes$$

Example:
Software Name Low= 0x0011
Software Name High= 0x4567
Software Version= 0x0002
results in: 114567V02

2.45 Software Name High

address : E203

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

Coding:

Display in HEX !

$$Software\ NameHighWord = Data\ byte$$

2.46 Software version

Address : E204

This parameter the number of the software version. Coding:

Software version= Data byte

2.47 Speed actual value

Address : E205

Coding:

$$Actual\ speed\ [1/\min] = Data\ bytes$$

2.48 Current direction of rotation

Address : E206

Current direction of

rotation Coding:

Value	Running direction
0	Preferred direction of rotation
1	Counter rotation direction

2.49 Output level (Ref PWM)

Address : E207

Output level at which the motor currently being operated.

Coding:

$$Deductible\,\% = \frac{Data\ byte}{65536} \cdot 100\%$$

2.50 Setpoint currently applied

Address : E208

Setpoint that currently applied during a setpoint ramp. Coding:

For "Speed control" in 1/min For

"Power control" in watts For

"Control":

$$Setpoint\,\% = \frac{Data\ byte}{65536} \cdot 100\%$$

2.51 Error status

Address : E209

The motor status indicates errors currently present on the fan. Coding:

MSB

LSB

0	0	0	UA	BLK	ASE	UeLow	UeHigh

If a bit is set, the corresponding error described below has occurred:

UeHigh: Overvoltage
UeLow: Undervoltage
ASE: Number of start attempts
reached BLK: Engine blocked
UA: Hardware overcurrent switch-off (HW reset required)

2.52 Warnings

Address E20A

Motor warnings warnings currently present on the fan. Coding:

MSB

0	0	0	0	0	0	0	0

LSB

0	0	0	PWR_Limit	RPM_Limit	SF	0	F_RPM

If a bit is set, the corresponding warning described below has occurred: F_RPM: The actual

speed deviates from the specified speed more than tolerated.
The tolerance can be set in (E129 Running monitoring tolerance).
The tolerated time until a deviation is signalled can be specified in (E12A Run monitoring time offset).
The warning is only active in "Speed control" operating mode (E115 operating mode).
SF: The operating hours of the fan have exceeded the value set in (E139 Service time).
RPM_Limit: The desired setpoint for the fan is by the speed limiter.
The limiting value can be set in HR (E14E Speed limiting value). The limiter must be activated in (E151 Limiter Enable).
PWR_Limit: The desired setpoint for the fan is limited by the power limitation.
The limiting value can be set in HR (E14F Power limiting value). The limiter must be activated in (E151 Limiter Enable).

2.53 Version Parameter set

Address : E20B

Contains the parameter set version

2.54 Power actual

value Address : E20C

Coding:

Actual power [Watt]= Data bytes

2.55 DB Index

Address : E20D

Specifies the firmware name from E202-E204 in more detail for test versions. Coding:

Index= Data bytes