MODBUS® interface definition DME401 / 440

Camille Bauer AG CH-5610 Wohlen, Switzerland

GOSSEN METRAWATT CAMILLE BAUER

Contents

1. Introduction

The manual in its present form describes in detail all possible functions which may be performed via the RS485 interfacee using protocols defined in the MODBUS® specification. The entirety of the provided information allows to create an independent software solution which uses all possibilities of the transducer.:

- Interrogation of a free selectable number of measurands (till the maximum of available values)
- Interrogation/setting/resetting of programmed internal meters (maximum 4)
- Configuration of all relevant parameters of the transducer including measurands and characteristic of the analog outputs, rated input values, measurands to evaluate for bus interrogation, meter measurands etc.
- Resetting possibly defined slave pointers

Normally this is undesired because existing hardware and / or software solutions are used. The abundance of information misguides to read the manual halfway. Obscurities and time-consuming troubleshooting may be the consequences. Therefore we will give a help to various users which chapters may contain essential information for him.

Hardware installer

2. Connecting devices to the bus

User of a MODBUSÒtool

4. Measurand acquisition

Engineer who wants to program an independent measurand acquisition

3. Realization of interface
4. Measurand acquisition
5. Meter readings
6. Resetting slave pointers

Engineer who wants to program an independent transducer configuration tool

3. Realization of interface
7. Configuration of the transducer

MODBUSÒ - Modbus is a registered trademark of Schneider Automation Inc.

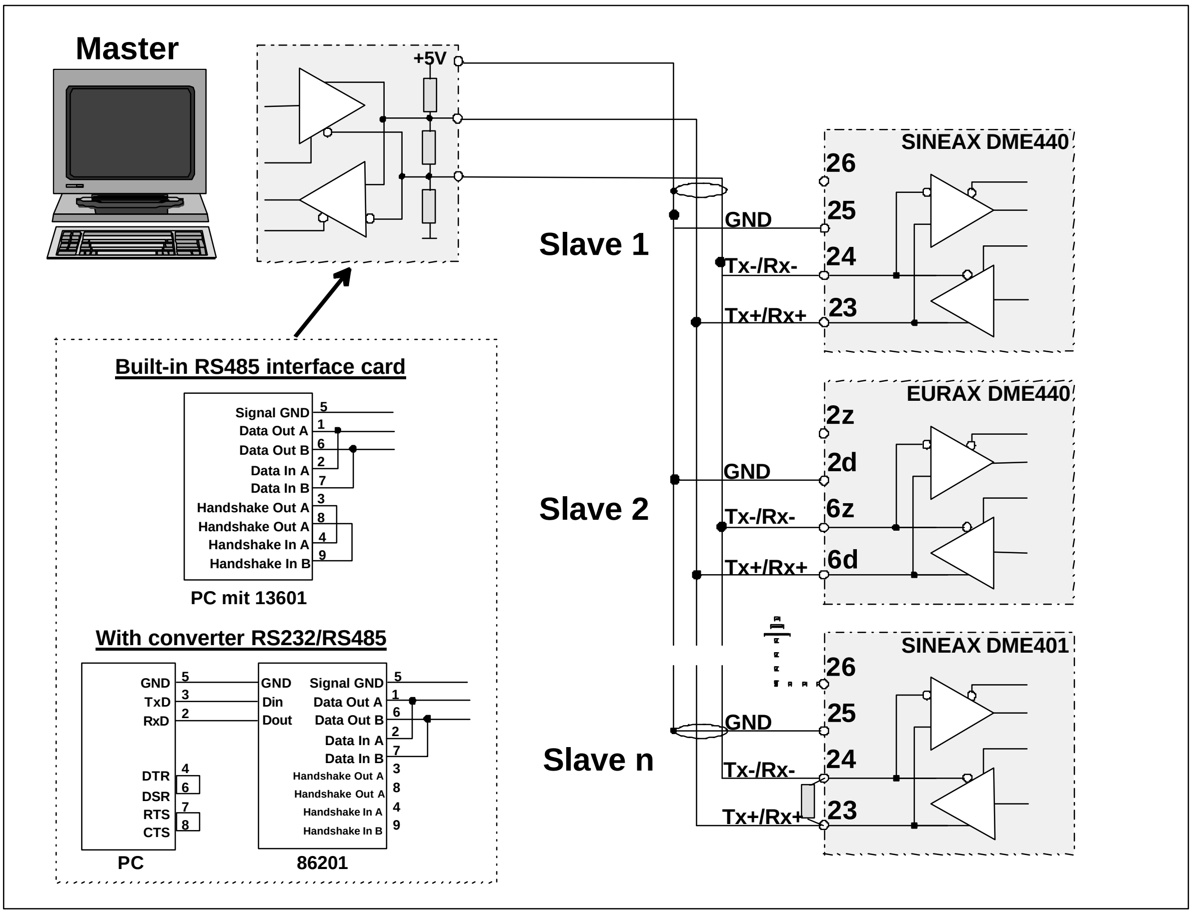
2. Connecting devices to the bus

The RS485 interfaces of the transducers are connected via an isolated cable. The shield both prevents the coupling of external noise to the bus and limits emissions from the bus. The shield must be connected to solid ground.

You can connect up to 32 members to the bus (including master). Basically devices of different manufacturers can be connected to the bus, if they use the standard MODBUS® protocol. Devices without galvanically isolated bus interfaces are not allowed to be connected to the shield.

The optimal topology for the bus is the daisy chain connection from node 1 to node 2 to node n. The bus must form a single continuous path, and the nodes in the middle of the bus must have short stubs. Longer stubs would have a negative impact on signal quality (reflexion at the end). A star or even ring topology is not allowed.

There is no bus termination required due to low data rate. If you got problems when using long cables you can terminate the bus at both ends with the characteristic impedance of the cable (normally about 120 Ω). Interface converters RS232⇔RS485 or RS485 interface cards often have a built-in termination network which can be connected to the bus. The second impedance then can be connected directly between the bus terminals of the device far most.



The grafic shows the connection of transducers DME401/440 to the MODBUS®. The RS485 interface can be realized by means of PC built-in interface cards or interface converters. Both is shown using i.e. the interfaces '13601' and '86201' of W&T (Wiesemann & Theis GmbH). They are configured for a 2-wire application with automatic control of data direction.

Important: ***- Each device connected to the bus must have a unique address***

- All devices must be adjusted to the same baudrate

Device address and baudrate can be set via the local RS232 interface only, using the configuration software for SINEAX / EURAX DME4.

3. Realization of interface

3.1 Interface configuration

The bus interface uses the following transmission mode:

- 1 start bit (0), 8 data bits, 1 stop bit (1), no parity
- baudrate 1’200..9’600 baud (programmable via RS-232), presetting 9’600 baud

3.2 Transmission principle

The transmission is fully controlled by the master (PC). No connected device is allowed to send data without prior request by the master. The master as well monitors possibly occuring timeouts (no response from the addressed device). Messages are transmitted using the RTU (remote terminal unit) mode.

The MODBUS® protocol defines a silent interval of 3,5 characters following the last transmitted character as end of the message. After this interval a new message can be sent. A typical message frame is shown below:

Pause	Message 1	→	*Pause*	Message 2	→	*Pause*	Message 3	etc.

The silent interval can be prolongated (configured via RS232 interface).

Remark: *A prolongation of the minimal silent interval may be necessary if the master is not able to send subsequent characters fast enough and therefore causes a transmission break. This effect may occur if a PC with power management is used (especially notebooks). A high baudrate reinforces this effect.*

3.3 General message frame

address (8 bits)	**function** (8 bits)	**data**	**crc16** (16 bits)

address: Address of the device which should perform an action (query message from master) or which is responding (response message from slave). The allowed address range is 1..247.

function: Tells what kind of action to perform. The following function codes are used in communication with the DME401/440:

Code	MODBUS® function	Used for ...
03_H	READ HOLDING REGISTERS	- measurand acquisition - meter reading - reading scaling factors - reading measurand table - reading configuration data
10_H	PRESET MULTIPLE REGISTERS	- programming the transducer - meter setting - selecting measurands to evaluate - resetting slave pointers

data: Contains the information to transmit. This field is divided into register, number of register to transmit and if necessary read data or information to store. Data is normally transmitted as 16 bit registers but there are also 32 bit numbers (double registers) and double bytes used (see chapter 3.5).

crc16: The cyclic redundancy check calculation is performed on the message contents to detect transmission errors.

3.4 Cyclic redundancy check calculation (crc16) *(example using ‘C)’*

The calculation is performed on all message characters, except the check bytes itself. The low-order byte (Crc_LByte) is appended to the message first, followed by the high-order byte (Crc_HByte). The receiver of the message calculates the check bytes again and compares them with the received ones.

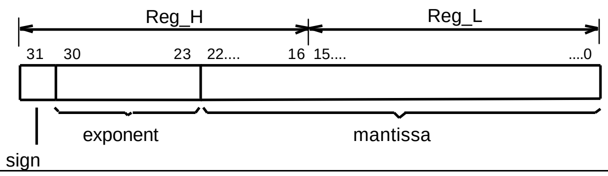
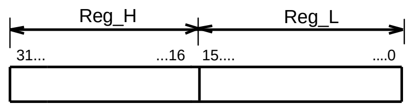
```
void main()
{
   unsigned char data[NUMDATA+2];                    // Message buffer
   unsigned char Crc_HByte,LByte;                    //
   unsigned int Crc;
   ....
   Crc=0xFFFF;
   for (i=0; i<NUMDATA; i++) {
     Crc = CRC16 (Crc, data[i] );
   }
   Crc_LByte = (Crc & 0x00FF);                       // Calculate low-order byte
   Crc_HByte = (Crc & 0xFF00) / 256;                 // Calculate high-order byte
}
// CRC16 calculation
// ----------------
unsigned int CRC16(unsigned int crc, unsigned int data)
{
  const unsigned int Poly16=0xA001;
  unsigned int LSB, i;

  crc = ((crc^data) | 0xFF00) & (crc | 0x00FF);
  for (i=0; i<8; i++) {
    LSB=(crc & 0x0001);
    crc=crc/2;
    if (LSB)
      crc=crc^Poly16;
  }
  return(crc);
}
```

3.5 Special data types

The standard MODBUS® protocol uses 16 bit registers for data transmission. To adapt the transducers data structure and to improve accuracy the following data types are used as well:

- ***32 bit numbers***: 32 bit unsigned integers and 32 bit real numbers are transmitted as two consecutive 16 bit registers. The format of the real number corresponds to the format normally used in PCs.

Type	***32 bit real***	***32 bit unsigned integer***
Format		
Calculation	$\text{Value} = (-1)^{\text{sign}} * 2^{(\text{exponent}-126)} * \frac{\text{mantissa} + 2^{23}}{2^{24}}$	

Transmission order:

Reg_H		*Reg_L*	
HByte	LByte	HByte	LByte

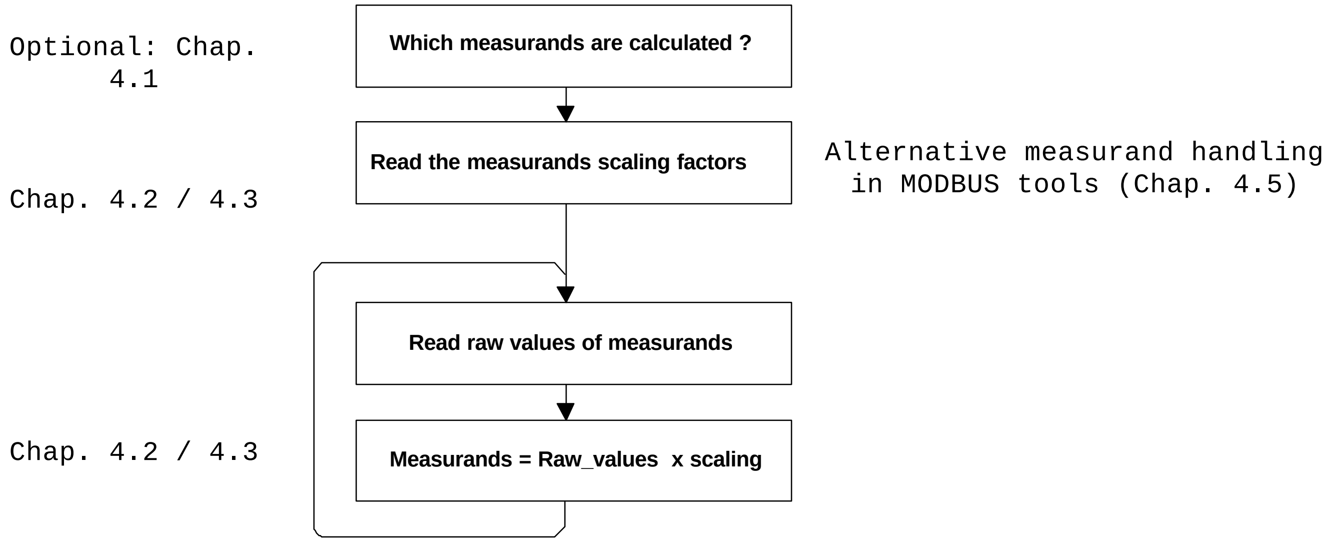
- ***Double bytes***: In PCs a 16 bit register is stored with the low byte on the lower and the high byte on the higher address. For the purpose of data transmission they will be swapped. Using double bytes two 8 bit characters are combined to a 16 bit register. But in opposition to 16 bit registers double bytes are ***not swapped*** for data transmission.

4. Measurand acquisition

The measurands are stored as table-oriented raw values. To calculate the real physical values of the measurands a further table contains appropriate scaling factors for each measurand.

$$\text{Meaurand} = \text{Raw\_value} * \text{Scaling\_factor}$$

Scaling factors are constant, as long as the transducers reference quantities are not reprogrammed. If there are primary transformers defined, their ratings are included in the scaling factors. So the calculated physical values are always primary values.



The DME4 supports up to 47 different measurands. If a measurand is calculable depends on the selected application (system).

Table 1

Measurand	Single phase connection	3-wire unbalanced	4-wire unbalanced	Measurand	Single phase connection	3-wire unbalanced	4-wire unbalanced
1: U	✓	-	-	25: QF1	-	-	✓
2: U1N	-	-	✓	26: QF2	-	-	✓
3: U2N	-	-	✓	27: QF3	-	-	✓
4: U3N	-	-	✓	28: F	✓	✓	✓
5: U12	-	✓	✓	29: S	✓	✓	✓
6: U23	-	✓	✓	30: S1	-	-	✓
7: U31	-	✓	✓	31: S2	-	-	✓
8: I	✓	-	-	32: S3	-	-	✓
9: I1	-	✓	✓	33: IM	-	✓	✓
10: I2	-	✓	✓	34: IMS	-	✓	✓
11: I3	-	✓	✓	35: LF	✓	✓	✓
12: P	✓	✓	✓	36: LF1	-	-	✓
13: P1	-	-	✓	37: LF2	-	-	✓
14: P2	-	-	✓	38: LF3	-	-	✓
15: P3	-	-	✓	39: IB 15 min	✓	-	-
16: Q	✓	✓	✓	40: IB1 15 min	-	✓	✓
17: Q1	-	-	✓	41: IB2 15 min	-	✓	✓
18: Q2	-	-	✓	42: IB3 15 min	-	✓	✓
19: Q3	-	-	✓	43: BS 15 min	✓	-	-
20: PF	✓	✓	✓	44: BS1 15 min	-	✓	✓
21: PF1	-	-	✓	45: BS2 15 min	-	✓	✓
22: PF2	-	-	✓	46: BS3 15 min	-	✓	✓
23: PF3	-	-	✓	47: UM	-	-	✓
24: QF	✓	✓	✓				

(✓=Measurand can be calculated)

4.1 Measurands

Which measurands are calculated within the device can be requested or modified via the registers 220..222. If a bit is set the value will be calculated. The calculable measurands are shown in table 1.

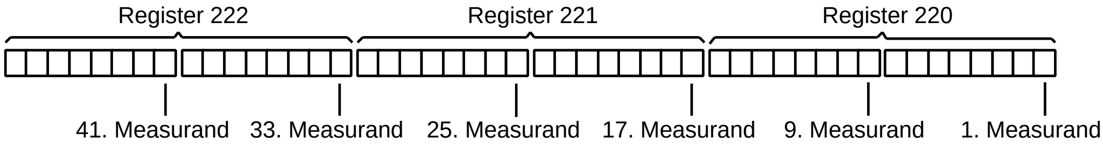


Table 2

Register	Content	Register	Content	Register	Content
220	Measurands 1..8 Measurands 9..16	221	Measurands 17..24 Measurands 25..32	222	Measurands 33..40 Measurands 41..48

4.2 Measurand tables

Table 3: Scaling factors *(32-Bit real numbers, double registers),* ***read only***

Register	Measurand	Register	Measurand	Register	Measurand
300	U	332	Q1	364	IM
302	U1N	334	Q2	366	IMS
304	U2N	336	Q3	368	LF
306	U3N	338	PF	370	LF1
308	U12	340	PF1	372	LF2
310	U23	342	PF2	374	LF3
312	U31	344	PF3	376	IB 15 min
314	I	346	QF	378	IB1 15 min
316	I1	348	QF1	380	IB2 15 min
318	I2	350	QF2	382	IB3 15 min
320	I3	352	QF3	384	BS 15 min
322	P	354	F	386	BS1 15 min
324	P1	356	S	388	BS2 15 min
326	P2	358	S1	390	BS3 15 min
328	P3	360	S2	392	UM
330	Q	362	S3		

Table 4: Raw values *(16-Bit integer),* ***read only***

Register	Measurand	Register	Measurand	Register	Measurand
100	U	116	Q1	364	IM
101	U1N	117	Q2	366	IMS
102	U2N	118	Q3	368	LF
103	U3N	119	PF	370	LF1
104	U12	120	PF1	372	LF2
105	U23	121	PF2	374	LF3
106	U31	122	PF3	376	IB 15 min
107	I	123	QF	378	IB1 15 min
108	I1	124	QF1	380	IB2 15 min
109	I2	125	QF2	382	IB3 15 min
110	I3	126	QF3	384	BS 15 min
111	P	127	F	386	BS1 15 min
112	P1	128	S	388	BS2 15 min
113	P2	129	S1	390	BS3 15 min
114	P3	130	S2	392	UM
115	Q	131	S3		

Range of values: 10’000	corresponds with	100%	of the rated value of the measurand
0	corresponds with	0 %	of the rated value of the measurand
-10’000	corresponds with	-100%	of the rated value of the measurand

Exception: The frequency is directly stored in [mHz]. The appropriate range of values is 15’300...65’000.

4.3 Optimized data transmission (recommended for independent software solutions only)

These tables contain scaling and raw values for calculated measurands only. Therefore a reduced effort for the data transmission results. However, the position of each measurand in the data block has to be investigated and stored in advance. The sequence of raw values and scaling factors is the same as for the previous defined complete measurand tables (Chap. 4.2), but the non-calculated values are removed. The number of bits set in table 1 (measurands) determines how many registers have to be read.

Table 5 (status: 'read only')

Register	Content
`400..`	Scaling factors for all calculated measurands

Table 6 (status: 'read only')

Register	Content
`150..`	All calculated measurands

Restriction: The tables have to be read starting at register address 150 or 400 respectively.

4.4 Examples for measurands acquisition

Request for single measurand 'P' (table 1: measurand 12) (Example for device 7):

- Optional request for all measurands really calculated

(Check if the measurand is indeed calculated)

Device	Code	Register		Number of registers		Checksum	
		HByte	LByte	HByte	LByte	LByte	HByte
07_H	03_H	00_H	DC_H	00_H	03_H	CRC16	

Device response:

Device	Code	Number of bytes	Data	Checksum	
				LByte	HByte
07_H	03_H	06_H	*6 data bytes*	CRC16	

- Request: Complete scaling table *(only once)*

Device	Code	Register		Number of registers		Checksum	
		HByte	LByte	HByte	LByte	LByte	HByte
07_H	03_H	01_H	$2C_H$	00_H	$5E_H$	CRC16	

Device response:

Device	Code	Number of bytes	Data	Checksum	
				LByte	HByte
07_H	03_H	BC_H	*188 bytes*	CRC16	

- Periodical request: Single measurand 'P' from table 3

Device	Code	Register		Number of registers		Checksum	
		HByte	LByte	HByte	LByte	LByte	HByte
07_H	03_H	00_H	$6F_H$	00_H	01_H	CRC16	

Device response:

Device	Code	Number of bytes	Data		Checksum	
			HByte	LByte	LByte	HByte
07_H	03_H	02_H	27_H	10_H	CRC16	

The received measurand in this example is 10'000 (2710_H). For conversion to physical value it has to be multiplied with the appropriate scaling factor (register 322 and 323).

Optimized data transmission (Example for device 7)

- *Optional:* ***One-time request for scaling factors of all calculated measurands****, table 5:*

Device	Code	Register		Number of registers		Checksum	
		HByte	LByte	HByte	LByte	LByte	HByte
07_H	03_H	01_H	90_H	00_H	10_H	CRC16	

Device response:

Device	Code	Number of bytes	Data		Checksum	
			HByte	LByte	LByte	HByte
07_H	03_H	20_H	27_H	10_H	CRC16	

This example shows the interrogation of the scaling factors of 8 measurands. It assumes that the number of really calculated measurands (chapter 4.1) has 8 bits set in the table.

- ***Periodical request for all calculated measurands****, table 6*

Device	Code	Register		Number of registers		Checksum	
		HByte	LByte	HByte	LByte	LByte	HByte
07_H	03_H	00_H	96_H	00_H	08_H	CRC16	

Device response:

Device	Code	Number of bytes	Data	Checksum	
				LByte	HByte
07_H	03_H	10_H	*16 bytes*	CRC16	

This example shows the acquisition of 8 measurands.

4.5 Alternative measurands handling in MODBUS tools

If tool can handle single register requests only

If the acquisition tool cannot handle requests for multiple register (blocks) you have to use the measurand tables descripted in chapter 4.2. The tables for optimized data transmission can't be used because the starting address for the register request is fixed.

If tool can't combine multiple registers to one measurand

Some MODBUS tools don't allow to calculate a measurand out of multiple acquisited register values. But normally there's the possibility to multiply acquisited data with constants.

The way to handle this best is to do it without a scaling factor request. The factors are entered as constants into the acquisition tool. However, modifications of the tranducer configuration can't be adopted automatically this way.

On the next page the calculation of the scaling factors for each measurand, depending on the selected application, is shown. The necessary parameters for the factor calculation may be requested using the DME4 configuration software (via the local RS232 interface). The shown calculation is valid for the optimized measurand tables as well.

Table 7: Scaling factors calculation

Register	Measurand	Single-phase system	4-wire balanced load	3-wire balanced load	3-wire unbalanced	4-wire unbalanced
100	1: U	Un	Un	Un	-	-
101	2: U1N	-	-	-	-	Un/10000/√3
102	3: U2N	-	-	-	-	Un/10000/√3
103	4: U3N	-	-	-	-	Un/10000/√3
104	5: U12	-	-	-	Un/10000	Un/10000
105	6: U23	-	-	-	Un/10000	Un/10000
106	7: U31	-	-	-	Un/10000	Un/10000
107	8: I	In	In	In	-	-
108	9: I1	-	-	-	In/10000	In/10000
109	10: I2	-	-	-	In/10000	In/10000
110	11: I3	-	-	-	In/10000	In/10000
111	12: P	Un*In/10000	Un*In*3/10000	Un*In*√3/10000	Un*In*√3/1000	Un*In*√3/1000
112	13: P1	-	-	-	-	Un*In/√3/1000
113	14: P2	-	-	-	-	Un*In/√3/1000
114	15: P3	-	-	-	-	Un*In/√3/1000
115	16: Q	Un*In/10000	Un*In*3/10000	Un*In*√3/10000	Un*In*√3/1000	Un*In*√3/1000
116	17: Q1	-	-	-	-	Un*In/√3/1000
117	18: Q2	-	-	-	-	Un*In/√3/1000
118	19: Q3	-	-	-	-	Un*In/√3/1000
119	20: PF	0.0001	0.0001	0.0001	0.0001	0.0001
120	21: PF1	-	-	-	-	0.0001
121	22: PF2	-	-	-	-	0.0001
122	23: PF3	-	-	-	-	0.0001
123	24: QF	0.0001	0.0001	0.0001	0.0001	0.0001
124	25: QF1	-	-	-	-	0.0001
125	26: QF2	-	-	-	-	0.0001
126	27: QF3	-	-	-	-	0.0001
127	28: F				0.0001	0.0001
128	29: S	Un*In/10000	Un*In*3/10000	Un*In*√3/10000	Un*In*√3/1000	Un*In*√3/1000
129	30: S1	-	-	-	-	Un*In/√3/1000
130	31: S2	-	-	-	-	Un*In/√3/1000
131	32: S3	-	-	-	-	Un*In/√3/1000
132	33: IM	-	-	-	In/10000	In/10000
133	34: IMS	-	-	-	In/10000	In/10000
134	35: LF	0.0001	0.0001	0.0001	0.0001	0.0001
135	36: LF1	-	-	-	-	0.0001
136	37: LF2	-	-	-	-	0.0001
137	38: LF3	-	-	-	-	0.0001
138	39: IB 15 min	In/10000	In/10000	In/10000	-	-
139	40: IB1 15 min	-	-	-	In/10000	In/10000
140	41: IB2 15 min	-	-	-	In/10000	In/10000
141	42: IB3 15 min	-	-	-	In/10000	In/10000
142	43: BS 15 min	In/10000	In/10000	In/10000	-	-
143	44: BS1 15 min	-	-	-	In/10000	In/10000
144	45: BS2 15 min	-	-	-	In/10000	In/10000
145	46: BS3 15 min	-	-	-	In/10000	In/10000
146	47: UM	-	-	-	-	Un/10000/√3

In = Ir according to the configuration software (primary value if available, secondary value otherways)
Un = Ur according to the configuration software (primary value if available, secondary value otherways)

5. Meter readings

Up to four internal counters can be configured at the same time. The meter readings are stored as unsigned 32 bit integers. For every meter there is also a scaling factor available (32 bit real number), which represent the conversion from secondary to primary values if transformers are used. The current meter standings are already scaled in the basic units [Wh, Varh, Vah, mAh].

Table 8

Register	Content	Register	Content
200	Internal meter 1	204	Internal meter 3
202	Internal meter 2	206	Internal meter 4

Table 9 (status: 'read only')

Register	Content	Register	Content
500	Scaling for meter 1	504	Scaling for meter 3
502	Scaling for meter 2	506	Scaling for meter 4

Meter readings (shown for device 7)

- *Single request for the meters scaling factors*

Device	Code	Register		Number of registers		Checksum	
		HByte	LByte	HByte	LByte	LByte	HByte
07_H	03_H	01_H	$F4_H$	00_H	08_H	CRC16	

Device response:

Device	Code	Number of bytes	Data	Checksum	
				LByte	HByte
07_H	03_H	10_H	*16 data bytes*	CRC16	

- *Periodical request for current meter standings*

Device	Code	Register		Number of registers		Checksum	
		HByte	LByte	HByte	LByte	LByte	HByte
07_H	03_H	00_H	$C8_H$	00_H	08_H	CRC16	

Device response:

Device	Code	Number of bytes	Data	Checksum	
				LByte	HByte
07_H	03_H	10_H	*16 data bytes*	CRC16	

Messages settings (shown for device 7)

You have to store secondary values for the meters (conversion maybe required).

- *Request: Setting meter 1 to a specific value*

Device	Code	Register		Number of registers		Number of bytes	Data	Checksum	
		HByte	LByte	HByte	LByte			LByte	HByte
07_H	10_H	00_H	$C8_H$	00_H	02_H	04_H	*4 data bytes*	CRC16	

Device response:

Device	Code	Register		Number of registers		Checksum	
		HByte	LByte	HByte	LByte	LByte	HByte
07_H	10_H	00_H	$C8_H$	00_H	02_H	CRC16	

6. Resetting slave pointers

A maximum of four slave pointers can be defined for the analog outputs A to D (DME440 only) and for the bus measurands 43..46 (with fixed response time of 15 min.). They can be resetted by setting the appropriate bit(s) in the following register.

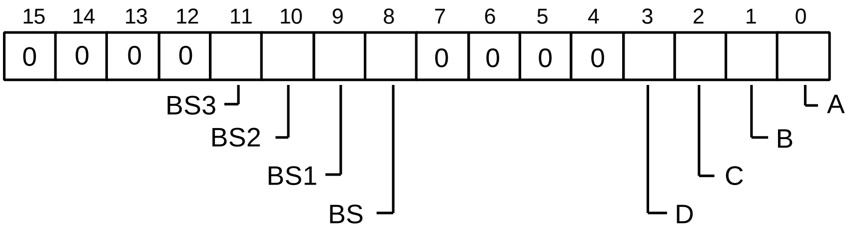


Table 8

Register	Content
230	slave pointers reset

After resetting the slave pointers the transducer will initialize the register to all zeroes.

Resetting a slave pointer (example for device 7):

- *Request: Reset slave pointer of analog output B*

Device	Code	Register		Number of registers		Number of bytes	Daten	Checksum	
		HByte	LByte	HByte	LByte			LByte	HByte
07_H	10_H	00_H	$E6_H$	00_H	01_H	02_H	00_H, 02_H	CRC16	

Device response:

Device	Code	Register		Number of registers		Checksum	
		HByte	LByte	HByte	LByte	LByte	HByte
07_H	10_H	00_H	$E6_H$	00_H	01_H	CRC16	

7. Configuration of the transducer

By changing the parameters the transducer can be suit to changing conditions. It’s possible to change individual registers or multiple contiguous registers at once.

Following each step changing parameters the transducer starts up to initialize the new configuration. For about 3 seconds the transducer cannot handle communication requests.

(Used data types (C): BYTE=unsigned char, WORD=unsigned int, LONG=unsigned long)

Table 9.1

Register	Type	Variable name		Meaning
700	WORD	ProgDate	*	Date of last **configuration**
701	BYTE	Application		Connected system
	BYTE	Freqmeas		Frequency measurement U/I, rated frequency
702	WORD	Ir		Rated current [mA]
703	LONG	PrimaryIr		Primary rated value of input current transformer [A]
705	WORD	Ur		Rated voltage
706	LONG	PrimaryUr		Primary rated value of input voltage transformer [V]
708	BYTE	Meas[12]		Measurands (used: 0..6 and 9)
714	int	XA[4][3]		Characteristic of analog measurands: X0;X1;X2
726	int	YA[4][3]		Characteristic of analog outputs:Y0;Y1;Y2Software
738	WORD	Ydel[12]		Response time of measurands
750	WORD	NotUsed1[22]		*Not used for DME401/440*
772	int	Y0Limit[4]		Lower range limit of analog outputs
776	int	Y2Limit[4]		Upper range limit of analog outputs
780	char	FileInfo[94]	*	User defined text for device description
Informal data (not used for device function)			☞	

Table 9.2 (status: ‘read only’, to be used for verification purpose only)

Register	Type	Variable name		Meaning
600	BYTE	OutpType[4]	*	***HW-Info***: type of analog outputs
602	WORD	Y2HW[4]	*	hardware range limit of analog outputs
606	BYTE	DeviceType	*	device type (constant)
	BYTE	CalFreq	*	calibration frequency
607	char	Password[8]	*	User defined password
611	char	PasswValidity	*	Validity of password protection
	BYTE	CalVers	*	Calibration version
612	BYTE	EPROMVers[2]	*	EPROM version, [0]=high_byte, [1]=low_Byte
Informal data (not used for device function)			☞	

Password protection*:* A possible defined password and its validity represent a protection mechanism against unpermitted modification of transducer data which is used in the standard software of DME4 for programming via RS232 interface. If you use this protection mechanism for the bus interface as well depends on your own requirements.

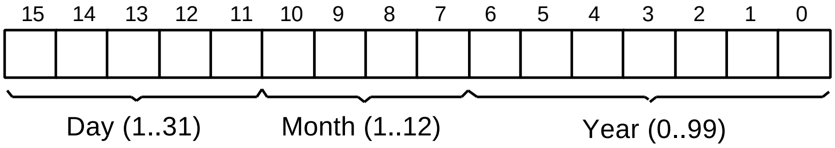
ATTENTION:

The individual parameters have to match with each other. So i.e. if you want to change the application (system) you have to check if all the configured measurands for the analog outputs are still measurable. If you perform a non balanced configuration the transducer may not work correctly.

The individual parameters will be descripted on the following pages.

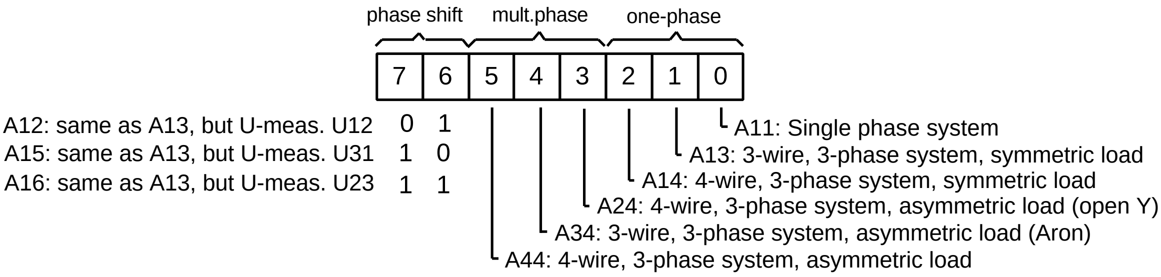
WORD ProgDate

Date of the last configuration of the transducer (informal).



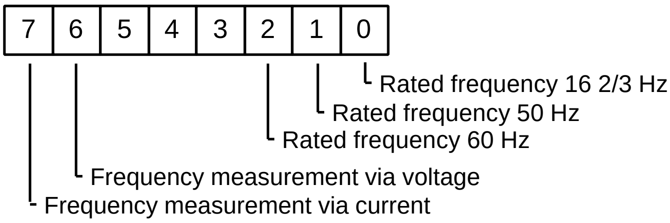
BYTE Application

Contains the application of the transducer. This information allows to derive which measurands in fact can be calculated respectively which of them are valid for measurand display (see table 1).



BYTE Freqmeas

Defines the rated frequency and the kind of frequency measurement. Frequency is measured via voltage path normally. If there is no voltage input or in cases of instability frequency measurement can be performed via current path. When programming, rated frequency and calibration frequency should be checked against each other. They should be the same, otherwise permissible variations will increase.



WORD Ir

Rated value of input current. Ir may be 0 (no current input) or 1000...6000 [mA].

LONG PrimaryIr

Defines the primary rated current of a possible current transformer in [A]. If there is no current transformer, this value should be zero otherwise maximum 200’000 A.

WORD Ur

Rated value of input voltage. You have to store **always** the phase-to-phase voltage, even if i.e. you measure in a single phase system where no phase-to-phase voltage exists. Ur contains the voltage [V]*50, so the allowed range is 5’000 ... 34’641 (100 ... 692.8V). Without input voltage you can set Ur to zero but you have to switch frequency measurement to ‘via current’ (see ‘Freqmeas’).

LONG PrimaryUr

Primary rated voltage of a possible voltage transformer in [V]. If there is no voltage transformer, this value should be zero otherwise maximum 2’000’000 V.

BYTE Meas[12]

Here the measurands for the analog outputs and the internal meters are defined.

Messgr	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]
DME440	**A**	**B**	**C**	**D**	$\mathbf{Z_1}$	$\mathbf{Z_2}$	$\mathbf{Z_3}$	FF_H	FF_H	$\mathbf{Z_4}$	FF_H	FF_H
DME401	FF_H	FF_H	FF_H	FF_H	$\mathbf{Z_1}$	$\mathbf{Z_2}$	$\mathbf{Z_3}$	FF_H	FF_H	$\mathbf{Z_4}$	FF_H	FF_H

[0]...[3]: Analog outputs **[4],[5],[6],[9]**: Internal meters

The measurand for unused outputs is 00_H and FF_H if the output doesn't exists. Normally you have to use the last two numbers of the measurand code in the order sheet, i.e. 12 for 'P', 25 for 'QF1' etc. (see also table 2). There are a few exceptions:

- For meters you have to use the same measurand as for the appropriate analog output, i.e the apparent power of the system 'S' is 29 not 54.
- Measurands for outgoing active power: 48..51
- Measurands for reactive power capacitive: 52..55
- For applications A24, A34 and A44 the measurand 1 (U12 with X0=0, X2=Ur) must be changed to 5.

int XA[4][3] (DME440 only)

Here you can define the lower and the upper range limits and if necessary the break points of the measurands which should appear at the analog outputs.

XA[A,B,C,D][X0,X1,X2].

The range of values for X0..X2 is as follows:

10'000	corresponds with	100%	of the rated value of the appropriate measurand
0	corresponds with	0 %	of the rated value of the appropriate measurand
-10'000	corresponds with	-100%	of the rated value of the appropriate measurand
15'291	($3BBB_H$) is the value for X1 if no break point is defined		

Examples for XA[ ][ ]: Rated input voltage Ur=100 V (phase-to-phase), rated input current Ir=1A

- Output A, Measurand I_1 (9), linear 0..1 [A]

XA[0][0] = 0 (0 % of the input)
XA[0][1] = 15'291 (no break point, linear)
XA[0][2] = 10'000 (100 % of the input)

- Output B, Measurand U_{1N} (2), linear 40..60 [V], 100 % of the input is 100/√3=57.74 V

XA[1][0] = 6'928 (69.28 % of the input)
XA[1][1] = 15'291 (no break point, linear)
XA[1][2] = 10'391 (103.91 % of the input)

- Output C, Measurand P_1 (13), -50..50 W, break point at 10 W, 100 % of the input is Ur*Ir/√3=57.74 W

XA[2][0] =-8'660 (-86.60 % of the input)
XA[2][1] = 1'732 (17.32 % of the input, independent of what analog output you got for it)
XA[2][2] = 8'660 (86.60 % of the input)

The programming of frequency measurement is an exception:

The values are directly in [mHz].

15,3 Hz -> 15'300 [mHz]
65,0 Hz -> 65'000 [mHz]

!!! ATTENTION: Here an 'unsigned int' will be stored in an 'int'.

int YA[4][3] **(DME440 only)**

Here you can define the lower and the upper range limits and if necessary the break points of the analog outputs. The upper range limit Y2 is always 10'000 (100% of the output) and therefore don't have to be programmed.

YA[A,B,C,D][Y0,Y1,Y2SW]

- Lower range limit Y0 and break point Y1are proportional in the range -10'000..10'000
- If there is no break point Y1 is 15'291 ($3BBB_H$)
- Using Y2SW you can reduce the output value by software:

YA[A,B,C,D][Y2SW] = 10'000 — if Y2 corresponds to the rated value of hardware output

YA[A,B,C,D][Y2SW] = $\frac{\text{desired range limit}}{\text{hardware range limit}} * 10'000$ — if Y2 doesn't correspond to the rated value of hardware output

- For unused or non existing analog outputs all values should be set to zero

Examples for YA[ ][ ]:

- Output A, configured 4..20 mA linear, hardware 20 mA
  - YA[0][0] = 2'000 (20 % of the configured upper range limit)
  - YA[0][1] = 15'291 (no break point defined, linear)
  - YA[0][2] = 10'000 (configured upper range limit = hardware range limit)
- Output B, configured -16..16 mA linear, hardware 20 mA
  - YA[1][0] =-10'000 (-100 % of the configured upper range limit)
  - YA[1][1] = 15'291 (no break point defined, linear)
  - YA[1][2] = 8'000 (configured upper range limit = 8'000/10'000 * 20mA = 16mA)
- Output C, configured 2..10V linear, hardware 10V, break point at 2V
  - YA[2][0] = 2'000 (20 % of the configured upper range limit)
  - YA[2][1] = 2'000 (break point at 20 % of the configured upper range limit)
  - YA[2][2] = 10'000 (configured upper range limit = hardware range limit)

WORD Ydel[12]

There's the possibility to define a response time for each measurand. You have to distinguish between measurands with short response time (1..30 s) and those with long response time (1..30 min). Unused elements should be set to zero. The assignment of the elements to the outputs is the following:

Tein	[0]	[1]	[2]	[3]	[4]..[11]
DME440	**A**	**B**	**C**	**D**	**not used**
DME401	**not used**				

[0]...[3]: Analog outputs **[4]...[11]**: not used

The range of values is 1'000..30'000 [ms] for those measurands with short response time. To choose minimal response time you can set the value to zero.

For bimetal measuring functions and slave pointers the range is 1..30 min (*1000), zero is not allowed.

int Y0Limit[4] **(DME440 only)**

Lower range limit of the analog outputs. The range of values is {YA[i][0]-2500} ... YA[i][0]. So the range limit can be maximum 25 % below the configured lower range limit of the output (Y0). For non existing analog outputs set the appropriate element to zero.

int Y2Limit[4] **(DME440 only)**

Upper range limit of the analog outputs. The range of values is 10'000..12'500 representing 100..125 % of the configured upper range limit. For non existing analog outputs set the appropriate element to zero.

Example: Output A, output range 4..20mA, desired range limits at 2.5 and 22 mA

-> Y0Limit[0] = 1'250

-> Y2Limit[0] = 11'000

char FileInfo[94]
User defined text for a device description. You can use characters with ASCII code higher than 32 or the value 0 for unused characters at the end of the text. In the standard software DME4 this text is normally divided into 3 lines. The position of the linefeed is realized with the characters <CR> and <LF> (ASCII code 13 and 10). These characters are part of *'FileInfo'* and therefore the maximum of usable characters is reduced to 94 characters.

<u>The following parameters are</u> ***<u>'read only':</u>***

BYTE OutpType[4] **(DME440 only)**
Specifies the kind of analog outputs A..D. Valid arguments are 1 (current output) and 2 (voltage output). All other values should be interpreted as non existing outputs. These parameters have no influence on the device function and are used for display functions only.

WORD Y2HW[4] **(DME440 only)**
The hardware range limits of the analog outputs are fixed by the PCB assembly. The format is the following:

- Voltage output: $\text{Upper\_range\_limit}[V] * \frac{10000}{10[V]}$
- Current output: $\text{Upper\_range\_limit}[mA] * \frac{10000}{20[mA]}$

BYTE DeviceType
40 (DME440) or 1 (DME401).

BYTE CalFreq
Contains the information for what rated frequency the transducer is calibrated. This frequency should be the same as defined in '*Freqmeas*' otherwise permissible variations will increase.

7	6	5	4	3	2	1	0

- Bit 3: Calibration frequency 55 Hz
- Bit 2: Calibration frequency 60 Hz
- Bit 1: Calibration frequency 50 Hz
- Bit 0: Calibration frequency 16 2/3 Hz

char Password[8]
User defined password. Allowed characters are those with ASCII code 32 to 127. To set password protection inactive all characters should be set to 32 (space). Use the password protection to prevent from unpermitted modifications of transducer data. See 'PasswValidity' to know for what operations password protection will be active.

BYTE PasswValidity
Defines for what operations the password protection mechanism will be valid:

7	6	5	4	3	2	1	0

- Bit 0: Storing a new configuration
- Bit 1: Setting meters
- Bit 2: Resetting slave pointers
- Bit 3: Performing a customer output calibration
- Bit 4: Simulation of output values

BYTE CalVers
Version of used calibration program (0..255)

BYTE EPROMVers[2]
EPROM version, [0]=high_byte, [1]=low_byte

Messages for the configuration:

- *Request: Changing the upper range limit of analog output A to 105 % of the rated value*

Device	Code	Register		Number of registers		Number of bytes	Daten		Checksum	
		HByte	LByte	HByte	LByte		HByte	LByte	LByte	HByte
07_H	10_H	00_H	20_H	00_H	01_H	02_H	29_H	04_H	CRC16	

Device response:

Device	Code	Register		Number of registers		Checksum	
		HByte	LByte	HByte	LByte	LByte	HByte
07_H	10_H	00_H	20_H	00_H	01_H	CRC16	

ATTENTION:

Following each step changing parameters the transducer starts up to initialize the new configuration. During about 3 seconds the transducer cannot handle communication requests.

8. Error messages

If the receiver of a message detects an error he sends back an appropriate error message to the master.

Device response:

Device	Code	Data	Checksum	
			LByte	HByte
07_H	Code+80_H	***error code***	CRC16	

The device sends the received function code back. However, to signal the error the most significant bit (MSB) of the function code byte will be set. The error code always signals a programming or operating error (never a transmission error). The following error codes will be used:

Error code	Meaning
01_H	Use of a unsupported function code
02_H	Use of an invalid memory register address: Use of invalid register number or attempt to write to a memory protected register.
03_H	Use of invalid data, i.e. an invalid number of registers
06_H	Device is busy. This code signals that the transducer is occupied with functions performed via the local RS232 interface. These may be: changing configuration, simulation or calibration of analog outputs.

If a transmission error occurs of that kind that the receivers crc16 calculation does not match the received one, by no means a response is sent to the master therefore provoking a timeout. The same happens if a non existing or switched-off device will be addressed.