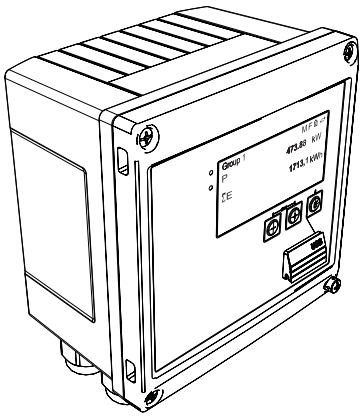
Level | Pressure | Flow | Temperature | Liquid Analysis | Registration | Systems Components | Services | Solutions

Additional Operating Instructions MODBUS

EngyCal® RH33 and RS33 and Batch Controller RA33

BTU meter / Steam calculator / Batch Controller



BA01029K/09/EN/13.10
71120459

Endress+Hauser
People for Process Automation

Table of contents

1 General information

The present MODBUS operating instructions are not a substitute for the general operating instructions for the EngyCal® RH33 and RS33 and the Batch Controller RA33.

These additional operating instructions only present information relevant for the MODBUS settings. For general safety notes, installation, wiring and commissioning, refer to the Operating Instructions of the device on the supplied CD-ROM.

2 Modbus message

2.1 Introduction

The master/slave technique is used for data exchange, whereby only the master can initiate transmission. On receipt of a request, the slave sends the required data to the master in the form of a response message or executes the command requested by the master.

2.2 Message structure

Data is transferred between the master and the slave in a message. A request message from the master contains the following fields:

Message structure

Slave address	Function code	Data	Checksum

- Slave address
  The slave address can be in the range 1 to 247.
  The slave address 0 (broadcast message) is used to transmit a message to be received by all slaves.
- Function code
  The function code defines the read, write or test action that is is to be executed via the Modbus protocol.
- Data
  The values listed below, amongst others, can be transmitted in this data field depending on the function code:
  – Register starting address (from which the data is transmitted)
  – Number of registers
  – Read/write data
  – Data length
  – etc.
- Checksum (CRC or LRC check)
  The message checksum forms the end of the message.

The master can send another message to the slave as soon as it receives a response to the previous message or after the time-out set for the master has elapsed. This time-out can be specified and changed by the user and depends on the slave's response time.

If an error occurs during data transmission or if the slave cannot execute the command requested by the master, the slave sends an error message (exception response) to the master.

The slave's response consists of message fields, which contain the requested data or confirm that the action requested by the master was carried out, as well as a checksum.

3 Modbus function codes

The function code defines the read, write or test action to be executed via the Modbus protocol. The device supports the following function codes:

Function code	Name as per Modbus specification	Description
03	READ HOLDING REGISTER	One or more of the Modbus slave registers are read. Between one and a maximum of 90 consecutive registers (1 register = 2 bytes) can be read with one message. **Application:** Read measured values, e.g. read the volume flow.
04	READ INPUT REGISTER	See READ HOLDING REGISTER
08	DIAGNOSTICS	Check the communication connection between the master and the slave (only for Modbus RTU). The following "diagnostics codes" are supported: Sub-function 00 = Return query data (loopback test)

The device does not make a distinction between function codes 03 and 04; they both yield the same result.

4 Modbus register addresses

4.1 Modbus register address model

The Modbus register addresses of the device are implemented in accordance with the "MODBUS Applications Protocol Specification V1.1".

In addition to the specification mentioned above, systems are also deployed which work with a register address model in accordance with the "Modicon MODBUS Protocol Reference Guide (PI-MBUS-300 Rev. J)" specification.

WIth this specification, the register address is extended depending on the function code used. A "4" is put in front of the register address for "READ HOLDING REGISTER (03)", and a "3" for "READ INPUT REGISTER (04).

Function code	Acess type	Register as per "MODBUS Applications Protocol Specification"		Register as per "Modicon MODBUS Protocol Reference Guide"
03	Read	XXXX Example: Value = 1	➔	4XXXX Example: Value = 40001
04	Read	XXXX Example: Value = 1	➔	3XXXX Example: Value = 30001

4.2 Data types

The following data types are supported by the device:

FLOAT (Floating-point number IEEE 754)

Data length = 4 bytes (2 registers)

Byte 3	Byte 2	Byte 1	Byte 0
SEEEEEEE	EMMMMMMM	MMMMMMMM	MMMMMMMM
S = Sign E = Exponent M = Mantissa			

INTEGER

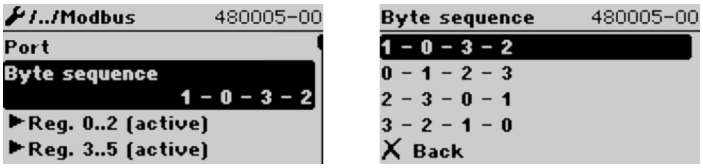
Data length = 2 bytes (1 register)

Byte 1	Byte 0
Most significant byte (MSB)	Least significant byte (LSB)

5 Transmission sequence of the bytes

The addressing of the bytes, i.e. the sequence in which they are transmitted, is not defined in the Modbus specification. It is therefore important to agree on or adjust the addressing mode between the master and slave during commissioning. This can be configured in the device using the "Byte sequence" parameter.

The bytes are transmitted depending on the option selected in the "Byte sequence" parameter under **../Setup/Advanced setup/System/Modbus**:



1 Setup/Advanced setup/System/Modbus menu

FLOAT:

Data length = 4 bytes (2 registers)

	Sequence			
Option	**1**	**2**	**3**	**4**
1 – 0 – 3 – 2 *	Byte 1 (MMMMMMMM)	Byte 0 (MMMMMMMM)	Byte 3 (SEEEEEEE)	Byte 2 (EMMMMMMM)
0 – 1 – 2 – 3	Byte 0 (MMMMMMMM)	Byte 1 (MMMMMMMM)	Byte 2 (EMMMMMMM)	Byte 3 (SEEEEEEE)
2 – 3 – 0 – 1	Byte 2 (EMMMMMMM)	Byte 3 (SEEEEEEE)	Byte 0 (MMMMMMMM)	Byte 1 (MMMMMMMM)
3 – 2 – 1 – 0	Byte 3 (SEEEEEEE)	Byte 2 (EMMMMMMM)	Byte 1 (MMMMMMMM)	Byte 0 (MMMMMMMM)
*) Default setting S = Sign E = Exponent M = Mantissa				

INTEGER: (Status)

	Sequence	
Option	**1**	**2**
1 – 0 – 3 – 2 * 3 – 2 – **1 – 0**	Byte 1 (MSB)	Byte 0 (LSB)
0 – 1 – 2 – 3 2 – 3 – **0 – 1**	Byte 0 (LSB)	Byte 1 (MSB)
* = Default setting MSB = Most significant byte LSB = Least significant byte		

6 Modbus error messages

If the Modbus slave detects an error in the master's request message, it sends an error message to the master as response. The error message consists of the slave address, function code, error code (exception code) and checksum. To indicate that this is an error message, the lead bit of the returned function code is used. The cause of the error is transmitted to the master by means of the exception code.

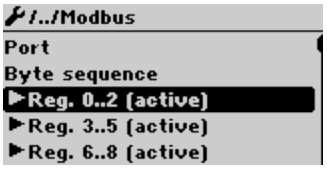
The following exception codes are supported by the device:

Exception codes	Description
01	ILLEGAL_FUNCTION The function code sent by the master is not supported by the device (slave). For a description of the function codes supported by the device, see (→ 5).
02	ILLEGAL_DATA_ADDRESS The register addressed by the master is not assigned (i.e. it does not exist), or the requested data is too long.
03	ILLEGAL_DATA_VALUE The value that appears in the data field is not permitted: e.g. range limits exceeded or incorrect data format.

7 Modbus register list

7.1 Functional description

In the device, up to 30 parameters can be flexibly assigned to registers 0 to 89 under **../Setup/ Advanced setup/System/Modbus/Reg 0..2 to Reg 87..89**. The master can address this entire data block with a single request message (register address (base 1) 1 to 90, 1001-1060, 2001-2030).



A0014382-EN

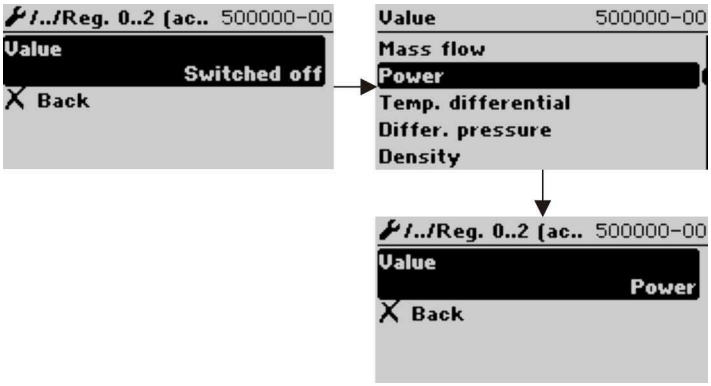
2 Setup / Advanced setup / System / Modbus

Example (RH33):

The values which can be assigned for RS33 and RA33 in the setup differ from the ones shown here.

The following device parameters should be grouped with the register list and read with one request message from the master:

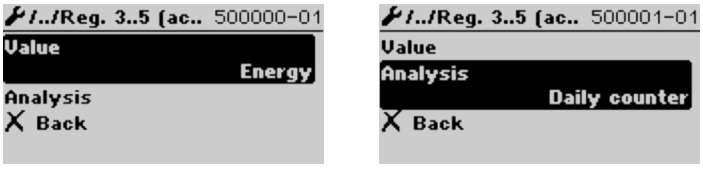
1. Power (address 0)



A0014383-EN

3 Modbus setup, assign parameters

2. Heat (Energy), Daily counter (address 3)



A0014384-EN

4 Modbus setup, assign parameters

From address 1, the status and the value are each provided in 3 registers (this corresponds to assignment in the device).

From address 1001, only the values are provided in 2 registers each.

From address 2001, only the statuses are provided in 1 register each.

No.	Value	Register address (base 1)	Contents	Register address (base 1)	Contents	Register address (base 1)	Contents
		(No.-1)*3+1		(No.-1)*2+1001		(No.-1)+2001	
1	Reg 0..2	0001-0003	Status+Float	1001-1002	Float	2001	Status
2	Reg 3..5	0004-0006	Status+Float	1003-1004	Float	2002	Status
3	Reg 6..8	0007-0009	Status+Float	1005-1006	Float	2003	Status
4	Reg 9..11	0010-0012	Status+Float	1007-1008	Float	2004	Status
5	Reg 12..14	0013-0015	Status+Float	1009-1010	Float	2005	Status
6	Reg 15..17	0016-0018	Status+Float	1011-1012	Float	2006	Status
7	Reg 18..20	0019-0021	Status+Float	1013-1014	Float	2007	Status
8	Reg 21..23	0022-0024	Status+Float	1015-1016	Float	2008	Status
9	Reg 24..26	0025-0027	Status+Float	1017-1018	Float	2009	Status
10	Reg 27..29	0028-0030	Status+Float	1019-1020	Float	2010	Status
11	Reg 30..32	0031-0033	Status+Float	1021-1022	Float	2011	Status
12	Reg 33..35	0034-0036	Status+Float	1023-1024	Float	2012	Status
13	Reg 36..38	0037-0039	Status+Float	1025-1026	Float	2013	Status
14	Reg 39..41	0040-0042	Status+Float	1027-1028	Float	2014	Status
15	Reg 42..44	0043-0045	Status+Float	1029-1030	Float	2015	Status
16	Reg 45..47	0046-0048	Status+Float	1031-1032	Float	2016	Status
17	Reg 48..50	0049-0051	Status+Float	1033-1034	Float	2017	Status
18	Reg 51..53	0052-0054	Status+Float	1035-1036	Float	2018	Status
19	Reg 54..56	0055-0057	Status+Float	1037-1038	Float	2019	Status
20	Reg 57..59	0058-0060	Status+Float	1039-1040	Float	2020	Status
21	Reg 60..62	0061-0063	Status+Float	1041-1042	Float	2021	Status
22	Reg 63..65	0064-0066	Status+Float	1043-1044	Float	2022	Status
23	Reg 66..68	0067-0069	Status+Float	1045-1046	Float	2023	Status
24	Reg 69..71	0070-0072	Status+Float	1047-1048	Float	2024	Status
25	Reg 72..74	0073-0075	Status+Float	1049-1050	Float	2025	Status
26	Reg 75..77	0076-0078	Status+Float	1051-1052	Float	2026	Status
27	Reg 78..80	0079-0081	Status+Float	1053-1054	Float	2027	Status
28	Reg 81..83	0082-0084	Status+Float	1055-1056	Float	2028	Status
29	Reg 84..86	0085-0087	Status+Float	1057-1058	Float	2029	Status
30	Reg 87..89	0088-0090	Status+Float	1059-1060	Float	2030	Status

Status (Integer):

Bits	Description
Bit 0..3	
0x0000	OK
0x0001	Cable open circuit
0x0002	Over range
0x0003	Under range
0x0004	Invalid measured value
0x0006	Error value, i.e. not the calculated value

Bits	Description
0x0007	Sensor error
Bit 4..7	
0x0010	Lower limit value
0x0020	Upper limit value
Bit 15	
0x8000	Counter overflow

www.endress.com/worldwide

Endress+Hauser

People for Process Automation

BA01029K/09/EN/13.10
71120459