UBN30
DIN Rail LCD Power Meter

UBN310
DIN 96x96 & ANSI 4” Power Meter

UBN315
DIN 96x96 & ANSI 4” LCD Power Meter

UBN3060
DIN 144x144 & ANSI 4” LCD Power Meter

UBN3080
DIN 144x144 LCD Power Meter

Universal Berg Netzbaustein

MODBUS Protocol Manual
English

1UBMMDB31009

BERG
Energiekontrolle
www.berg-energie.de

Rev. 009 - 05/07/2007

TABLE OF CONTENTS

1. INTRODUCTION

This manual provides information on the MODBUS communication protocol. The publication is not intended for general use, but for qualified technicians.
This term indicates a professional and skilled technician, authorised to act in accordance with the safety standards relating to the dangers posed by electric current.
This person must also have basic first-aid training and be in possession of suitable Personal Protective Equipment.

WARNING!
It is strictly forbidden, for anyone who does not have the above-mentioned features, to install or use the device.

The device is made in compliance with the European Union directives in force, as well as in compliance with the technical standards implementing these requirements, as certified by the CE mark on the device and in this manual.
It is strictly forbidden to use the instrument for purposes other than those intended, which can be deduced from the manual content.

The Manufacturer reserves the right to make changes on the device or in the device specifications identified in this manual without notice.
The Manufacturer declines all liability for any use of the instrument which is different from that described in this manual and in the instrument manual, or for the lack/incorrect application of the reported instructions.

The information contained in this document is believed to be accurate at the time of publication, however, the Manufacturer assumes no responsability for any errors which may appear here and reserves the right to make changes without notice.

2. GRAPHIC SYMBOLS

In the manual and on the device, some instructions are highlighted by symbols to draw the reader's attention to the operational dangers.
These symbols are the following:

 **DANGER!**
This warning indicates the possible presence of voltage exceeding 1kV on the marked terminals (even for short periods).

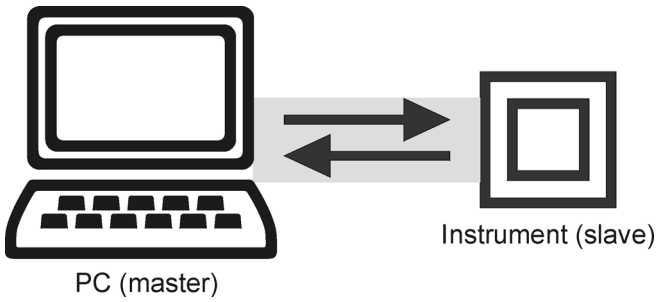
 **WARNING!**
This warning indicates the possible occurrence of an event which may cause a serious accident or considerable damage to the device if suitable precautionary countermeasures are not taken.

 **ATTENTION!**
This warning indicates the possible occurrence of an event which may cause a light accident or damage to the device if suitable precautionary countermeasures are not taken.

 **NOTE**
This warning indicates important information which must be read carefully.

3. DESCRIPTION

MODBUS is a master-slave communication protocol, able to support up to 247 slaves connected in a bus or a star network.
The protocol uses an half duplex connection on a single line. In this way, the communication messages move on a single line in two opposite directions.



MODBUS protocol can communicate in two modes:

1. **ASCII mode**: a limited set of characters is used for communication
2. **RTU mode**: binary system, with time-frame synchronization, faster than ASCII mode; data block is 50% shorter than ASCII mode block

A transmission mode must be define for all the instruments connected in the network.
MODBUS protocol cannot communicate, simultaneously, in two modes (ASCII, RTU).

Master-slave messages can be:

- **Read ($03)**: the communication is between the master and a single slave. It allows to read information about the queried instrument
- **Write ($10)**: the communication is between the master and a single slave. It allows to change the instrument settings
- **Broadcast**: the communication is between the master and all the connected slaves. It is always a write command ($10)

In a multi-point type connection, a code allows to identify each instrument during the communication. This code, the Logical Number, can be assigned by the user. Each instrument is supplied with a $01 Logical Number.

Structure of a generic message

FRAME START	ADDRESS FIELD	FUNCTION CODE	DATA FIELD	ERROR CONTROL	FRAME END

FRAME START Message start marker
ADDRESS FIELD Instrument Logical Number. Use $00 for broadcast commands
FUNCTION CODE Function code: $03=read command; $10=write command
DATA FIELD Data field
ERROR CONTROL Error correction code
FRAME END End message marker

Communication frame structure

ASCII mode

Bit per byte	mode a) 1 Start, 7 Bit, 1 Parity, 1 Stop mode b) 1 Start, 7 Bit, 2 Stop
FRAME START	character ":" (0x3A)
ADDRESS FIELD	2 characters (instrument address identified by 1 byte in hexadecimal format)
FUNCTION CODE	2 characters (function code identified by 1 byte in hexadecimal format)
DATA FIELD	2 x N. characters (N. data byte in hexadecimal format)
ERROR CHECK	2 characters (LRC identified by 1 byte in hexadecimal format)
FRAME END	characters "CR" "LF" (0x0D 0x0A)

RTU mode

Bit per byte	mode a) 1 Start, 8 Bit, 1 Parity, 1 Stop mode b) 1 Start, 8 Bit, 2 Stop
FRAME START	silence on line for time $\geqslant$ 4 characters
ADDRESS FIELD	1 character
FUNCTION CODE	1 character
DATA FIELD	N. characters
ERROR CHECK	CRC 16 bit
FRAME END	silence on line for time $\geqslant$ 4 characters

3.1 LRC and CRC Generation

Example of the LRC generation with 'C' language:

```
byte hexchar_to_byte (char ascii_char)
{
          switch (ascii_char)
          {
                case '1': case '2': case '3': case '4': case '5': case '6': case '7': case '8':
                case '9': case '0':return (a-'0');
                case 'A': case 'a': return (0xa);
                case 'B': case 'b': return (0xb);
                case 'C': case 'c': return (0xc);
                case 'D': case 'd': return (0xd);
                case 'E': case 'e': return (0xe);
                case 'F': case 'f': return (0xf);
                default: return (0x0);
          }
}

byte LRC (byte *ptMsg, word usDataLen)
{
          byte lrc=0;
          word i;

          for (i=0;i<usDataLen;i=i+2)
               lrc += (hexchar_to_byte (ptMsg[i])<<4) +hexchar_to_byte (ptMsg[i+1]);

          return((byte) (-(signed char)lrc));
}
```

Example of the CRC 16 generation with 'C' language:

```
#define byte unsigned char
#define word unsigned short
const byte auchCRCHi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40
};
```

```
const byte auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4,
0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD,
0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,
0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,
0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,
0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,
0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,
0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80,
0x40
};

word CRC16 (byte*ptMsg, word usDataLen)
{
                byte uchCRCHi = 0xFF;       //high byte CRC
                byte uchCRCLo = 0xFF;       //low byte CRC
                word uIndex;

                while (usDataLen--)         //pass through message buffer
                {
                uIndex = uchCRCHi ^ *ptMsg++;
                uchCRCHi = uchCRCLo ^ auchCRCHi [uIndex];
                }
                return (uchCRCHi << 8 | uchCRCLo);
}
```

NOTE

The “Error Check LRC ” field must be calculated on byte pairs which represent one byte in ASCII characters. The start and the end of communication frame are not included in the calculation.
The “Error Check CRC16” field must be calculated on bytes. The start and the end of communication frame are not included in the calculation.

4. READ REGISTERS
(Function code $03)

The master communication device can send, to the slave device, commands to read measured values, instrument status, instrument setup.
Ex. of read commands of measured values: Instantaneous, Harmonics, ...
Ex. of read commands of instrument status&setup: Instrument settings, clock, ...

NOTE
More registers can be read, at the same time, sending a single command, only if the variables registers are consecutive. A single variable cannot be read with two separate commands.

Structure of a query message

FRAME START	ADDRESS FIELD	FUNCTION CODE	START REGISTER	REGISTERS NUMBER	ERROR CONTROL	FRAME END

FRAME START Message start marker
ADDRESS FIELD Instrument Logical Number (1 byte length)
FUNCTION CODE Function code: $03 (1 byte length)
START REGISTER Start register (1 word length)
REGISTERS NUMBER Registers number to read (1 word length)
ERROR CONTROL Error correction code
FRAME END End message marker

In the response message, each register contains data on 1 word.
For each register, the first byte contains the high order bits and the second contains the low order bits.

Structure of a response message

FRAME START	ADDRESS FIELD	FUNCTION CODE	NUMBER OF BYTES	D0	D1	...	Dn	ERROR CONTROL	FRAME END

FRAME START Message start marker
ADDRESS FIELD Instrument Logical Number (1 byte length)
FUNCTION CODE Function code $03 (1 byte length)
NUMBER OF BYTES Number of data bytes (1 byte length)
D0, D1,..., Dn Data bytes (1data word=2data bytes)
ERROR CONTROL Error correction code
FRAME END End message marker

See section 7 for the Registers Table and section 8 for Examples.

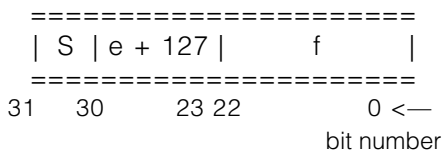
4.1 Floating Point as per IEEE Standard

The basic format allows a IEEE standard floating-point number to be represented in a single-32 bit format, as shown below:

$$N.n = (-1)^S \ 2^{\,e'-127} \ (1.f\,)$$

where **S** is the sign bit, **e'** is the first part of the exponent and **f** is the decimal fraction placed next to 1. Internally the exponent is 8 bits in length and the stored fraction is 23 bits long.

The floating-point format is shown as follows:



where:

	bit length
Sign	1
Exponent	8
Fraction	23 + (1)
Total	m = 32 + (1)
Exponent	
Min e'	0
Max e'	255
Bias	127

NOTE

Fractions (decimals) are always shown while the leading 1 (hidden bit) is not stored.

See section 7 for the Registers Table.

Example of conversion of value shown with floating point

Value read with floating point:
45AACC00($_{16}$)

Value converted in binary format:
0 10001011 01010101100110000000000 ($_2$)
sign | exponent | fraction

sign = 0
exponent = 10001011($_2$) = 139($_{10}$)
fraction = 01010101100110000000000($_2$) / 8388608 ($_{10}$) =
= 2804736 ($_{10}$) / 8388608 ($_{10}$) = 0.334350585 ($_{10}$)

$$N.n = (-1)^S \ 2^{e'-127} \ (1+f) =$$

$$= (-1)^0 \ 2^{139-127} \ (1.334350585) =$$

$$= (+1) \ (4096) \ (1.334350585) =$$

$$= 5465.5$$

5. PARAMETERS SETUP
(Function code $10)

In a network, the master communication device can send control and programming commands to the slave devices.
Example of control commands: Reset recording values, Reset counters, ...
Example of programming commands: set Logical number, set Wiring mode, ...
In broadcast mode, the same register values are set in all the slave devices connected in the same network.

NOTE
More registers can be write, at the same time, sending a single command, only if the variables registers are consecutive. A single variable cannot be write with two separate commands.

Structure of a query message

FRAME START	ADDRESS FIELD	FUNCTION CODE	START REGISTER	REGISTERS NUMBER	NUMBER OF BYTES	D0	D1	...	Dn	ERROR CONTROL	FRAME END

FRAME START Message start marker
ADDRESS FIELD Instrument Logical Number (1 byte length)
FUNCTION CODE Function code $10 (1 byte length)
START REGISTER Start register (1 word length)
REGISTERS NUMBER Registers number to be written (1 word length)
NUMBER OF BYTES Number of data bytes (1 byte length)
D0, D1,..., Dn Data bytes (1data word=2data bytes)
ERROR CONTROL Error correction code
FRAME END End message marker

Structure of a response message

FRAME START	ADDRESS FIELD	FUNCTION CODE	START REGISTER	REGISTERS NUMBER	ERROR CONTROL	FRAME END

FRAME START Message start marker
ADDRESS FIELD Instrument Logical Number (1 byte length)
FUNCTION CODE Function code: $10 (1 byte length)
START REGISTER Start register (1 word length)
REGISTERS NUMBER Written registers number (1 word length)
ERROR CONTROL Error correction code
FRAME END End message marker

See section 7 for the Registers Table and section 8 for the Examples.

6. ERROR MESSAGES

When a slave device receives a not-valid query, an error message is transmitted.

Structure of a response message

FRAME START	ADDRESS FIELD	FUNCTION CODE	ERROR CODE	ERROR CONTROL	FRAME END

FRAME START Message start marker
ADDRESS FIELD Instrument Logical Number (1 byte length)
FUNCTION CODE Function code $03 / $10 (1 byte length)
ERROR CODE Message containing the error code (1byte length)
ERROR CONTROL Error correction code
FRAME END End message marker

6.1 Error Example

QUERY

Field name	Example (HEX)
Slave address	01
Function code	03
Starting address Hi	80
Starting address Lo	00
Number of word Hi	00
Number of word Lo	05
Error check (CRC)	AC
	09

RESPONSE

Field name	Example (HEX)
Slave address	01
Function code	83 (1)
Error code	02 (2)
Error check (CRC)	C0
	F1

(1): Function code transmitted by master with bit 7 high.

(2): Error code 01=illegal function
02=illegal data address
03=illegal data value

7. REGISTERS TABLE

7.1 Important Notes for Registers

NOTE
The highest number of registers, which can be read with a single command, are:
- **ASCII mode: 63 registers (words)**
- **RTU mode: 127 registers (words)**

NOTE
The highest number of registers, which can be programmed with a single command, are:
- **ASCII mode: 13 registers (words)**
- **RTU mode: 29 registers (words)**

NOTE
The variable indicated by YES in RESET column, resets the instrument. Wait at least 3 seconds before sending a new query.

NOTE
In this manual, the register words are expressed in hexadecimal.

NOTE
The * symbol indicates a parameter available only when the instrument includes the relevant option.

7.2 Measured Values (Function code $03)

Reg	Word	Reg. (IEEE)	Word	Description	U. M.	Sign
$0000	4	$1000	2	SYSTEM VOLTAGE	[mV]	(Unsigned, MSB=0)
$0004	4	$1002	2	L-N VOLTAGE PHASE 1	[mV]	(Unsigned)
$0008	4	$1004	2	L-N VOLTAGE PHASE 2	[mV]	(Unsigned)
$000C	4	$1006	2	L-N VOLTAGE PHASE 3	[mV]	(Unsigned)
$0010	4	$1008	2	L-L VOLTAGE LINE 12	[mV]	(Unsigned)
$0014	4	$100A	2	L-L VOLTAGE LINE 23	[mV]	(Unsigned)
$0018	4	$100C	2	L-L VOLTAGE LINE 31	[mV]	(Unsigned)
$001C	4	$100E	2	SYSTEM CURRENT	[mA]	(Signed, negative MSB=1)
$0020	4	$1010	2	PHASE 1 CURRENT	[mA]	(Signed)
$0024	4	$1012	2	PHASE 2 CURRENT	[mA]	(Signed)
$0028	4	$1014	2	PHASE 3 CURRENT	[mA]	(Signed)
$002C	4	$1016	2	SYSTEM POWER FACTOR	[-]	(Signed)
$0030	4	$1018	2	PHASE 1 POWER FACTOR	[-]	(Signed)
$0034	4	$101A	2	PHASE 2 POWER FACTOR	[-]	(Signed)
$0038	4	$101C	2	PHASE 3 POWER FACTOR	[-]	(Signed)
$003C	4	$101E	2	PHASE ORDER	[-]	(Unsigned)
$0040	4	$1020	2	PHASE 1 COSØ *	[-]	(Signed)
$0044	4	$1022	2	PHASE 2 COSØ *	[-]	(Signed)
$0048	4	$1024	2	PHASE 3 COSØ *	[-]	(Signed)
$004C	4	$1026	2	SYSTEM APPARENT POWER	[mVA]	(Signed)
$0050	4	$1028	2	PHASE 1 APPARENT POWER	[mVA]	(Signed)
$0054	4	$102A	2	PHASE 2 APPARENT POWER	[mVA]	(Signed)
$0058	4	$102C	2	PHASE 3 APPARENT POWER	[mVA]	(Signed)
$005C	4	$102E	2	SYSTEM ACTIVE POWER	[mW]	(Signed)
$0060	4	$1030	2	PHASE 1 ACTIVE POWER	[mW]	(Signed)
$0064	4	$1032	2	PHASE 2 ACTIVE POWER	[mW]	(Signed)
$0068	4	$1034	2	PHASE 3 ACTIVE POWER	[mW]	(Signed)
$006C	4	$1036	2	SYSTEM REACTIVE POWER	[mvar]	(Signed)
$0070	4	$1038	2	PHASE 1 REACTIVE POWER	[mvar]	(Signed)
$0074	4	$103A	2	PHASE 2 REACTIVE POWER	[mvar]	(Signed)
$0078	4	$103C	2	PHASE 3 REACTIVE POWER	[mvar]	(Signed)
$007C	4	$103E	2	IMPORTED ACTIVE ENERGY	[mWh]	(Unsigned)
$0080	4	$1040	2	IMPORTED REACTIVE INDUCTIVE ENERGY	[mvarh]	(Unsigned)
$0084	4	$1042	2	EXPORTED ACTIVE ENERGY	[mWh]	(Unsigned)
$0088	4	$1044	2	EXPORTED REACTIVE INDUCTIVE ENERGY	[mvarh]	(Unsigned)
$008C	4	$1046	2	LINE FREQUENCY	[mHz]	(Unsigned)
$0090	4	$1048	2	PHASE 1 VOLTAGE THD	[m%]	(Unsigned)
$0094	4	$104A	2	PHASE 2 VOLTAGE THD	[m%]	(Unsigned)
$0098	4	$104C	2	PHASE 3 VOLTAGE THD	[m%]	(Unsigned)
$009C	4	$104E	2	PHASE 1 CURRENT THD	[m%]	(Unsigned)
$00A0	4	$1050	2	PHASE 2 CURRENT THD	[m%]	(Unsigned)
$00A4	4	$1052	2	PHASE 3 CURRENT THD	[m%]	(Unsigned)
$00A8	4	$1054	2	VOID	[-]	(Unsigned)
$00AC	4	$1056	2	VOID	[-]	(Unsigned)
$00B0	4	$1058	2	IMPORTED REACTIVE CAPACITIVE ENERGY	[mvarh]	(Unsigned)
$00B4	4	$105A	2	EXPORTED REACTIVE CAPACITIVE ENERGY	[mvarh]	(Unsigned)
$00B8	4	$105C	2	IMPORTED APPARENT ENERGY	[mVAh]	(Unsigned)
$00BC	4	$105E	2	EXPORTED APPARENT ENERGY	[mVAh]	(Unsigned)
$00C0	4	$1060	2	NEUTRAL CURRENT	[mA]	(Unsigned)
DEMAND AND MAXIMUM VALUES						
$00C4	4	$1062	2	SYSTEM CURRENT DEMAND	[mA]	(Signed)
$00C8	4	$1064	2	SYSTEM ACTIVE POWER DEMAND	[mW]	(Signed)
$00CC	4	$1066	2	SYSTEM APPARENT POWER DEMAND	[mVA]	(Signed)
$00D0	4	$1068	2	MAXIMUM PHASE 1 CURRENT	[mA]	(Signed)
$00D4	4	$106A	2	MAXIMUM PHASE 2 CURRENT	[mA]	(Signed)
$00D8	4	$106C	2	MAXIMUM PHASE 3 CURRENT	[mA]	(Signed)
$00DC	4	$106E	2	MAXIMUM SYSTEM CURRENT DMD	[mA]	(Signed)
$00E0	4	$1070	2	MAXIMUM SYSTEM ACTIVE POWER DMD	[mW]	(Signed)
$00E4	4	$1072	2	MAXIMUM SYSTEM APPARENT POWER DMD	[mVA]	(Signed)

7.3 Recordings

MINIMUM/AVERAGE/MAXIMUM VALUES (Function code $03)

Reg	Word	Reg. (IEEE)	Word	Description	U. M.	Sign
$E050	8			Variables selected for recording (see section 7.6)	[-]	(Unsigned)
$E058	1			Recording mode (0=NONE, 1=FILL, 2=RING)	[-]	(Unsigned)
$E059	1			Integration time (1, 5, 10, 15, 30, 60 minutes)	[-]	(Unsigned)
$E05A	2			Recording start time (HHMM, period1)	[-]	(Unsigned)
$E05C	2			Recording stop time (HHMM, period1)	[-]	(Unsigned)
$E05E	2			Recording start time (HHMM, period2)	[-]	(Unsigned)
$E060	2			Recording stop time (HHMM, period2)	[-]	(Unsigned)
$E062	1			Recorded data available (0=not present, 1=present)	[-]	(Unsigned)
$0A00	3	$8A00	6	YYMMDD	[-]	(Unsigned)
$0A03	3	$8A06	6	HHMMSS	[-]	(Unsigned)
$0A06	4	$8A0C	2	MINIMUM VALUE OF 1st VARIABLE	[?]	(Signed)
$0A0A	4	$8A0E	2	DEMAND VALUE OF 1st VARIABLE	[?]	(Signed)
$0A0E	4	$8A10	2	MAXIMUM VALUE OF 1st VARIABLE	[?]	(Signed)
$0A12	4	$8A12	2	MINIMUM VALUE OF 2nd VARIABLE	[?]	(Signed)
$0A16	4	$8A14	2	DEMAND VALUE OF 2nd VARIABLE	[?]	(Signed)
$0A1A	4	$8A16	2	MAXIMUM VALUE OF 2nd VARIABLE	[?]	(Signed)
$0A1E	4	$8A18	2	MINIMUM VALUE OF 3rd VARIABLE	[?]	(Signed)
$0A22	4	$8A1A	2	DEMAND VALUE OF 3rd VARIABLE	[?]	(Signed)
$0A26	4	$8A1C	2	MAXIMUM VALUE OF 3rd VARIABLE	[?]	(Signed)
$0A2A	4	$8A1E	2	MINIMUM VALUE OF 4th VARIABLE	[?]	(Signed)
$0A2E	4	$8A20	2	DEMAND VALUE OF 4th VARIABLE	[?]	(Signed)
$0A32	4	$8A22	2	MAXIMUM VALUE OF 4th VARIABLE	[?]	(Signed)
$0A36	4	$8A24	2	MINIMUM VALUE OF 5th VARIABLE	[?]	(Signed)
$0A3A	4	$8A26	2	DEMAND VALUE OF 5th VARIABLE	[?]	(Signed)
$0A3E	4	$8A28	2	MAXIMUM VALUE OF 5th VARIABLE	[?]	(Signed)
$0A42	4	$8A2A	2	MINIMUM VALUE OF 6th VARIABLE	[?]	(Signed)
$0A46	4	$8A2C	2	DEMAND VALUE OF 6th VARIABLE	[?]	(Signed)
$0A4A	4	$8A2E	2	MAXIMUM VALUE OF 6th VARIABLE	[?]	(Signed)
$0A4E	4	$8A30	2	MINIMUM VALUE OF 7th VARIABLE	[?]	(Signed)
$0A52	4	$8A32	2	DEMAND VALUE OF 7th VARIABLE	[?]	(Signed)
$0A56	4	$8A34	2	MAXIMUM VALUE OF 7th VARIABLE	[?]	(Signed)
$0A5A	4	$8A36	2	MINIMUM VALUE OF 8th VARIABLE	[?]	(Signed)
$0A5E	4	$8A38	2	DEMAND VALUE OF 8th VARIABLE	[?]	(Signed)
$0A62	4	$8A3A	2	MAXIMUM VALUE OF 8th VARIABLE	[?]	(Signed)

The number of words depends on the number of variables selected for recording.
[?] (Signed): the unit of measurement and the sign depend on the selected variable.

MINIMUM/AVERAGE/MAXIMUM VALUES (Function code $10)

Reg.	Word	Description	Range	Reset
$E0A2	1	DOWNLOAD RECORDED DATA	0=none 1=start to download 1° data block 2=download the next data block	NO
$E0A3	1	DELETE MIN/AVG/MAX RECORDING	0=none 1=delete Min/Avg/Max recordings 2=delete Min/Avg/Max (recordings & setup)	YES

DAILY ENERGY COUNTERS RECORDING (Function code $03)

Reg	Word	Reg. (IEEE)	Word	Description	U. M.	Sign
$0A70	3	$8A40	6	YYMMDD	[-]	(Unsigned)
$0A73	4	$8A46	2	IMPORTED ACTIVE ENERGY	[mWh]	(Unsigned)
$0A77	4	$8A48	2	IMPORTED REACTIVE INDUCTIVE ENERGY	[mvarh]	(Unsigned)
$0A7B	4	$8A4A	2	IMPORTED REACTIVE CAPACITIVE ENERGY	[mvarh]	(Unsigned)
$0A7F	4	$8A4C	2	IMPORTED APPARENT ENERGY	[mVAh]	(Unsigned)
$0A83	4	$8A4E	2	EXPORTED ACTIVE ENERGY	[mWh]	(Unsigned)
$0A87	4	$8A50	2	EXPORTED REACTIVE INDUCTIVE ENERGY	[mvarh]	(Unsigned)
$0A8B	4	$8A52	2	EXPORTED REACTIVE CAPACITIVE ENERGY	[mvarh]	(Unsigned)
$0A8F	4	$8A54	2	EXPORTED APPARENT ENERGY	[mVAh]	(Unsigned)

DAILY ENERGY COUNTERS RECORDING (Function code $10)

Reg.	Word	Description	Range	Reset
$E0A2	1	DOWNLOAD RECORDINGS	0=none 3=start to download 1° data block 4=download the next data block	NO
$E0A3	1	DELETE RECORDINGS	0=none 3=delete daily energy counters rec. F=delete all recordings	YES

7.4 Read Only Parameters (Function code $03)

Reg.	Word	Description	Range
$E000	5	SERIAL NUMBER	
$E005	7	VERSION NUMBER	
$E00C	1	INSTRUMENT TYPE	0B=UBN310 0C=UBN3060 0D=UBN3080 15=UBN315 43=UBN30
$E00D	1	MEMORY	0=no memory 1=128kB
$E00F	1	NUMBER OF DIGITAL OUTPUTS	0=none 2=2 4=4 * 6=6 * 8=8 *
$E010	1	NUMBER OF ANALOG OUTPUTS *	0=none 1=1 2=2 4=4
$E011	1	NUMBER OF DIGITAL INPUTS *	0=none 4=4
$E012	1	NUMBER OF ANALOG INPUTS *	0=none 1=1 2=2 4=4
$E013	1	NUMBER OF EXTERNAL MODULES *	0=none 1=EM80A (8 analog outputs)
$E015	1	NUMBER OF HARMONICS	
$0A66	2	MAX MEMORY SIZE OF MIN/AVG/MAX RECORDING	
$0A68	2	MEMORY SIZE OF THE CURRENT MIN/AVG/MAX RECORDING	
$0A9B	2	MAX MEMORY SIZE OF DAILY ENERGY COUNTERS RECORDING	
$0A9D	2	MEMORY SIZE OF THE CURRENT DAILY ENERGY COUNTERS RECORDING	

7.5 Read & Write Parameters (Function code $03 & $10)

Reg.	Word	Description	Range	Reset
$E020	1	LOGICAL NUMBER (1-247)	$0001-$00F7	YES
$E021	3	DATE	YYMMDD	NO
$E024	1	DAY OF THE WEEK	1=Monday 2=Tuesday 3=Wednesday 4=Thursday 5=Friday 6=Saturday 7=Sunday	NO
$E025	3	TIME	HHMMSS	NO
$E028	1	BAUDRATE(COM 1)	0=300 baud 1=600 baud 2=1200 baud 3=2400 baud 4=4800 baud 5=9600 baud 6=19200 baud 7=38400 baud 8=57600 baud	YES
$E029	1	PARITY (COM 1)	0=none 1=even 2=odd	YES
$E02A	1	DATA BITS (COM 1)	0=7 bit (ASCII) 1=8 bit (RTU) 2=switching on STANDARD protocol (8databit, parity none, 1stopbit, no change for baudrate)	YES
$E02B	1	BAUDRATE (COM 2)	see COM1	YES
$E02C	1	PARITY (COM 2)	see COM1	YES
$E02D	1	DATA BITS (COM 2)	see COM1	YES
$E02E	1	STOP BITS (COM 1) (only 3080)	0=1 bit 1=2 bit	YES
$E02F	1	STOP BITS (COM 2) (only 3080)	see COM1	YES
$E031	1	CT RATIO (1-9999)	$0001-$270F	YES
$E032	1	CURRENT INPUT	0=1A 1=5A	YES
$E034	2	PT RATIO (1.000-9999.999)	$0001-$270F(INTEGER) $0000-$03E7(DECIMAL)	YES
$E038	1	WIRING MODE	1=Single-phase 2=1 phase 3Wr/2CT (only 310) 3=3 phase 3Wr/2CT 4=3 phase 4Wr/3CT	YES

Reg.	Word	Description	Range	Reset
$E03C	1	DMD INTEGRATION TIME (min.)		YES
		01	$0001	
		05	$0005	
		10	$000A	
		15	$000F	
		30	$001E	
		60	$003C	
$E03D	1	BACKLIGHT ON TIME (seconds) (not available for 310 and 3060)		NO
		(000-999)	$0000-$03E7	

MIN/AVG/MAX RECORDING PROGRAMMING

$E050	8	VARIABLES (see section 7.6)	$00XX	NO
$E058	1	RECORDING MODE	0=none 1=fill 2=ring	NO
$E059	1	AVG INTEGRATION TIME (min.)		NO
		01	$0001	
		05	$0005	
		10	$000A	
		15	$000F	
		30	$001E	
		60	$003C	
$E05A	2	RECORDING START TIME OF THE FIRST PERIOD	HHMM	NO
$E05C	2	RECORDING STOP TIME OF THE FIRST PERIOD	HHMM	NO
$E05E	2	RECORDING START TIME OF THE SECOND PERIOD	HHMM	NO
$E060	2	RECORDING STOP TIME OF THE SECOND PERIOD	HHMM	NO
$E0A0	1	ENERGY COUNTERS RESET	0=none 1=reset energy counters 3=reset digital input counters * 4=reset all counters	YES
$E0A1	1	MAX DMD VALUES RESET	0=none 1=reset all 2=reset MAX DMD power 3=reset MAX DMD current	NO
$E0A2	1	DOWNLOAD RECORDED DATA	0=none 1=1° data block of Min/Avg/Max values 2=next data block of Min/Avg/Max values 3=first data block of daily counters 4=next data block of daily counters	NO
$E0A3	1	DELETE RECORDING	0=none 1=delete Min/Avg/Max recordings 2=delete Min/Avg/Max (recordings & setup) 3=delete daily energy counters F=delete all counters	YES

DIGITAL OUTPUT 1 PROGRAMMING

Reg.	Word	Description	Range	Reset
Threshold programming				
$E100	5	VARIABLE (see section 7.6)	$00XX	YES
		MODE	1= threshold with lower limit 2= threshold with upper limit	
		DELAY (0-999 sec.)	$0000-$03E7	
		HYSTERESIS (0-99%)	$0000-$0063	
		INTERVENTION VALUE (SET) (0-150.00% of FS)	$0000-$3A98 (Ex. 100.00% -> 10000 = $2710)	
Pulse programming				
$E100	5	VARIABLE (see section 7.6)	$00XX	YES
		MODE	3= pulse	
		PULSE DURATION (0-999 msec)	$0000-$03E7	
		PULSE COEFFICIENT (0-1000)	$0000-$03E8 (Ex. 125 = $007D)	
		HYSTERESIS	$0000 (fixed)	
$E105	1	PULSE COEFFICIENT (decimal point position) read only	0=0.1 1=0.01 2=0.001	NO
$E106	1	PULSE COEFFICIENT (multiplier) read only	0=none 1=K 2=M 3=G 4=T	NO

The content of registers $E105 and $E106 depends on the set CT and PT ratios.

DIGITAL OUTPUT 2 PROGRAMMING (see DIGITAL OUTPUT 1)

$E108	5	VARIABLE, MODE, DELAY, HYSTERESIS, SET
$E10D	1	PULSE COEFFICIENT (decimal point position) read only
$E10E	1	PULSE COEFFICIENT (multiplier) read only

DIGITAL OUTPUT 3 PROGRAMMING (see DIGITAL OUTPUT 1)*

$E110	5	VARIABLE, MODE, DELAY, HYSTERESIS, SET
$E115	1	PULSE COEFFICIENT (decimal point position) read only
$E116	1	PULSE COEFFICIENT (multiplier) read only

DIGITAL OUTPUT 4 PROGRAMMING (see DIGITAL OUTPUT 1)*

$E118	5	VARIABLE, MODE, DELAY, HYSTERESIS, SET
$E11D	1	PULSE COEFFICIENT (decimal point position) read only
$E11E	1	PULSE COEFFICIENT (multiplier) read only

DIGITAL OUTPUT 5 PROGRAMMING (see DIGITAL OUTPUT 1)*

$E120	5	VARIABLE, MODE, DELAY, HYSTERESIS, SET
$E125	1	PULSE COEFFICIENT (decimal point position) read only
$E126	1	PULSE COEFFICIENT (multiplier) read only

DIGITAL OUTPUT 6 PROGRAMMING (see DIGITAL OUTPUT 1)*

$E128	5	VARIABLE, MODE, DELAY, HYSTERESIS, SET
$E12D	1	PULSE COEFFICIENT (decimal point position) read only
$E12E	1	PULSE COEFFICIENT (multiplier) read only

ANALOG OUTPUT 1 PROGRAMMING*

Reg.	Word	Description	Range	Reset
$E150	4	VARIABLE (see section 7.6)	$00XX	YES
		MODE	0 = 0-20mA monodir. 1 = 4-20mA monodir. 2 = 0-20mA bidir. 3 = 4-20mA bidir.	
		MIN. THRESHOLD VALUE PERCENTAGE (0-9999)	$0000-$270F (Ex. 50.00% -> 5000 = $1388)	
		MAX. THRESHOLD VALUE PERCENTAGE (0-9999)	$0000-$270F (Ex. 90.00% -> 9000 = $2328)	

ANALOG OUTPUT 2 PROGRAMMING (see ANALOG OUTPUT 1)*

$E154	4	VARIABLE		
		MODE		
		MIN. THRESHOLD PERCENTAGE VALUE		
		MAX. THRESHOLD PERCENTAGE VALUE		

DIGITAL INPUT 1 PROGRAMMING*

$E160	1	MODE		YES
$E161	2	PULSE COEFFICIENT (00.00-99.99)	$0000-$0063 (INTEGER) $0000-$0063 (DECIMAL)	YES
$E163	1	MULTIPLIER	0=none 1=kWh 2=kVAh 3=kvarh 4=MWh 5=MVAh 6=Mvarh 7=GWh 8=GVAh 9=Gvarh 10=m3 11=l 12=N, h'F8 13=yd3 14=ft3 15=in3 16=gal 17=oz	NO
$E164	4	VOID (read only)		

DIGITAL INPUT 2 PROGRAMMING (see DIGITAL INPUT 1)*

$E168	1	MODE
$E169	2	PULSE COEFFICIENT
$E16B	1	MULTIPLIER
$E16C	4	VOID (read only)

DIGITAL INPUT 3 PROGRAMMING (see DIGITAL INPUT 1)*

$E170	1	MODE
$E171	2	PULSE COEFFICIENT
$E173	1	MULTIPLIER
$E174	4	VOID (read only)

DIGITAL INPUT 4 PROGRAMMING (see DIGITAL INPUT 1)*

$E178	1	MODE
$E179	2	PULSE COEFFICIENT
$E17B	1	MULTIPLIER
$E17C	4	VOID (read only)

7.6 Variables List

The code assigned to each variable is described in the following list.

Code	Description
00	System voltage
01	L-N voltage phase 1
02	L-N voltage phase 2
03	L-N voltage phase 3
04	L-L voltage line 12
05	L-L voltage line 23
06	L-L voltage line 31
08	System current
09	Phase 1 current
0A	Phase 2 current
0B	Phase 3 current
0C	Phase 1 current THD
0D	Phase 2 current THD
0E	Phase 3 current THD
0F	Neutral current
10	System power factor
11	Phase 1 power factor
12	Phase 2 power factor
13	Phase 3 power factor
15	Phase 1 COSØ *
16	Phase 2 COSØ *
17	Phase 3 COSØ *
18	System apparent power
19	Phase 1 apparent power
1A	Phase 2 apparent power
1B	Phase 3 apparent power
20	System active power
21	Phase 1 active power
22	Phase 2 active power
23	Phase 3 active power
28	System reactive power
29	Phase 1 reactive power
2A	Phase 2 reactive power
2B	Phase 3 reactive power
34	Line frequency
36	Phase 1 voltage THD
37	Phase 2 voltage THD
38	Phase 3 voltage THD
39	System DMD active power
3B	Phase order
FF	None variable selected
40 … 5E	Harmonic (H1 … H31) L-N voltage phase 1 *
5F … 7D	Harmonic (H1 … H31) L-N voltage phase 2 *
7E … 9C	Harmonic (H1 … H31) L-N voltage phase 3 *
9D … BB	Harmonic (H1 … H31) phase 1 current *
BC … DA	Harmonic (H1 … H31) phase 2 current *
DB … F9	Harmonic (H1 … H31) phase 3 current *
30	Imported active energy
31	Imported reactive inductive energy
33	Imported reactive capacitive energy
1C	Imported apparent energy
32	Exported active energy
2E	Exported reactive inductive energy
2F	Exported reactive capacitive energy
1D	Exported apparent energy

8. EXAMPLES

This chapter contains some examples of reading and setting registers.

8.1 Currents Values Reading (Function code $03)

QUERY

Field Name	Example (Hex)
Slave Address	01
Function code	03
Starting Address Hi	00
Starting Address Lo	1C
Number of Word Hi	00
Number of Word Lo	10
Error check (CRC)	85
	C0

RESPONSE

Field Name	Example (Hex)	
Slave Address	01	
Function code	03	
Byte count	20	
Data Hi	00	|
Data Lo	00	|
Data Hi	00	|
Data Lo	00	|
Data Hi	00	|==> AΣ value (4 words)
Data Lo	00	| E.g.: 2802 mA
Data Hi	0A	|
Data Lo	F2	|
Data Hi	00	|
Data Lo	00	|
Data Hi	00	|
Data Lo	00	|
Data Hi	00	|==> A1 value (4 words)
Data Lo	00	|
Data Hi	0A	|
Data Lo	F2	|
Data Hi	00	|
Data Lo	00	|
Data Hi	00	|
Data Lo	00	|
Data Hi	00	|==> A2 value (4 words)
Data Lo	00	|
Data Hi	0A	|
Data Lo	F2	|

Data Hi	00	|
Data Lo	00	|
Data Hi	00	|
Data Lo	00	|
Data Hi	00	|==> A3 value (4 words)
Data Lo	00	|
Data Hi	0A	|
Data Lo	F2	|
Error check (CRC)	7A	
	20	

8.2 Logical Number Setup (Function code $10)

QUERY

Field Name	Example (Hex)
Slave Address	01
Function code	10
Starting Address Hi	E0
Starting Address Lo	20
Number of Word Hi	00
Number of Word Lo	01
Byte count	02
Logical number Hi	00 |
Logical number Lo	01 |==> Logical Number = 01
Error check (CRC)	81
	3E

RESPONSE

Field Name	Example (Hex)
Slave Address	01
Function code	10
Starting Address Hi	E0
Starting Address Lo	20
Number of Word Hi	00
Number of Word Lo	01
Error check (CRC)	37
	C3

8.3 Communication Baud Rate Setup (Function code $10)

QUERY

Field Name	Example (Hex)	
Slave Address	01	
Function code	10	
Starting Address Hi	E0	
Starting Address Lo	28	
Number of Word Hi	00	
Number of Word Lo	01	
Byte count	02	
Baud Rate Hi	00	|
Baud Rate Lo	07	|==> Baud Rate = 38400 baud
Error check (CRC)	00	
	74	

RESPONSE

Field Name	Example (Hex)
Slave Address	01
Function code	10
Starting Address Hi	E0
Starting Address Lo	28
Number of Word Hi	00
Number of Word Lo	01
Error check (CRC)	B6
	01

8.4 Date/Day Setup (Function code $10)

QUERY

Field Name	Example (Hex)	
Slave Address	01	
Function code	10	
Starting Address Hi	E0	
Starting Address Lo	21	
Number of Word Hi	00	
Number of Word Lo	04	
Byte count	08	
Year Hi	00	|
Year Lo	02	|==> Year = 2002
Month Hi	00	|
Month Lo	0A	|==> Month = 10
Day Hi	00	|
Day Lo	04	|==> Day = 04
Day of the week Hi	00	|
Day of the week Lo	05	|==> Day of the week = FRIDAY
Error check (CRC)	8E	
	64	

RESPONSE

Field Name	Example (Hex)
Slave Address	01
Function code	10
Starting Address Hi	E0
Starting Address Lo	21
Number of Word Hi	00
Number of Word Lo	04
Error check (CRC)	A6
	00

8.5 PT Setup (Function code $10)

QUERY

Field Name	Example (Hex)		
Slave Address	01		
Function code	10		
Starting Address Hi	E0		
Starting Address Lo	34		
Number of Word Hi	00		
Number of Word Lo	02		
Byte count	04		
Data Hi	00	|	
Data Lo	01	|==>	Value with 2 words (E.g. 0001,0000)
Data Hi	00	|	First word = integer part
Data Lo	00	|	Second word = decimal part
Error check (CRC)	69		
	4C		

RESPONSE

Field Name	Example (Hex)
Slave Address	01
Function code	10
Starting Address Hi	E0
Starting Address Lo	34
Number of Word Hi	00
Number of Word Lo	02
Error check (CRC)	37
	C6

8.6 Min/Avg/Max Recording Setup (Function code $10)

QUERY

Field Name	Example (Hex)	
Slave Address	01	
Function code	10	
Starting Address Hi	E0	
Starting Address Lo	50	
Number of Word Hi	00	
Number of Word Lo	12	
Byte count	24	|==> 36 bytes in RTU mode; in ASCII mode it is necessary to split the command
1st selected variable Hi	00	|
1st selected variable Lo	00	|==> System voltage
2nd selected variable Hi	00	|
2nd selected variable Lo	08	|==> System current
3rd selected variable Hi	00	|
3rd selected variable Lo	0F	|==> Neutral current
4th selected variable Hi	00	|
4thselected variable Lo	10	|==> System power factor
5th selected variable Hi	00	|
5th selected variable Lo	20	|==> System active power
6th selected variable Hi	00	|
6th selected variable Lo	18	|==> System apparent power
7th selected variable Hi	00	|
7th selected variable Lo	34	|==> Line frequency
8th selected variable Hi	00	|
8th selected variable Lo	FF	|==> No variable selected
Recording mode Hi	00	|
Recording mode Lo	01	|==> Fill mode recording
Integration time Hi	00	|
Integration time Lo	01	|==> 01 minute
Start period 1 (hh) Hi	00	|
Start period 1 (hh) Lo	08	|
Start period 1 (mm) Hi	00	|
Start period 1 (mm) Lo	00	|==> 1° Start at 08:00
Stop period 1 (hh) Hi	00	|
Stop period 1 (hh) Lo	0C	|
Stop period 1 (mm) Hi	00	|
Stop period 1 (mm) Lo	00	|==> 1° Stop at 12:00

Start period 2 (hh) Hi	00	|
Start period 2 (hh) Lo	0D	|
Start period 2 (mm) Hi	00	|
Start period 2 (mm) Lo	1E	|==> 2° Start at 13:30
Stop period 2 (hh) Hi	00	|
Stop period 2 (hh) Lo	12	|
Stop period 2 (mm) Hi	00	|
Stop period 2 (mm) Lo	1E	|==> 2° Stop at 18:30
Error check (CRC)	D1	
	F0	

RESPONSE

Field Name	**Example (Hex)**
Slave Address	01
Function code	10
Starting Address Hi	E0
Starting Address Lo	50
Number of Word Hi	00
Number of Word Lo	12
Error check (CRC)	77
	D5

8.7 Min/Avg/Max Recording Download
(Function code $03 & $10)

QUERY A (read the first data block of MIN/AVG/MAX recording)

Field Name	Example (Hex)
Slave Address	01
Function code	03
Starting Address Hi	E0
Starting Address Lo	50
Number of Word Hi	00
Number of Word Lo	13
Error check (CRC)	33
	D6

RESPONSE A

Field Name	Example (Hex)	
Slave Address	01	
Function code	03	
No. of transmitted bytes	26	
1st stored variable Hi	00	|
1st stored variable Lo	00	|==> System voltage
2nd stored variable Hi	00	|
2nd stored variable Lo	08	|==> System current
3rd stored variable Hi	00	|
3rd stored variable Lo	0F	|==> Neutral current
4th stored variable Hi	00	|
4th stored variable Lo	10	|==> System power factor
5th stored variable Hi	00	|
5th stored variable Lo	20	|==> System active power
6th stored variable Hi	00	|
6th stored variable Lo	18	|==> System apparent power
7th stored variable Hi	00	|
7th stored variable Lo	34	|==> Line frequency
8th stored variable Hi	00	|
8th stored variable Lo	FF	|==> No variable selected
Recording mode Hi	00	|
Recording mode Lo	01	|==> Fill mode recording
Integration time Hi	00	|
Integration time Lo	01	|==> 01 minute

Start period 1 (hh) Hi	00	|
Start period 1 (hh) Lo	08	|
Start period 1 (mm) Hi	00	|
Start period 1 (mm) Lo	00	|==> 1° Start at 08:00
Stop period 1 (hh) Hi	00	|
Stop period 1 (hh) Lo	0C	|
Stop period 1 (mm) Hi	00	|
Stop period 1 (mm) Lo	00	|==> 1° Stop at 12:00
Start period 2 (hh) Hi	00	|
Start period 2 (hh) Lo	0D	|
Start period 2 (mm) Hi	00	|
Start period 2 (mm) Lo	1E	|==> 2° Start at 13:30
Stop period 2 (hh) Hi	00	|
Stop period 2 (hh) Lo	12	|
Stop period 2 (mm) Hi	00	|
Stop period 2 (mm) Lo	1E	|==> 2° Stop at 18:30
Data in RAM Hi	00	|
Data in RAM Lo	01	|==> Data in RAM available for downloading
Error check (CRC)	92	
	E8	

QUERY B (prepare the first data block for downloading)

Field Name	Example (Hex)	
Slave Address	01	
Function code	10	
Starting Address Hi	E0	
Starting Address Lo	A2	
Number of Word Hi	00	
Number of Word Lo	01	
Byte count	02	
First data Hi	00	|
First data Lo	01	|==> first block
Error check (CRC)	9F	
	1C	

RESPONSE B

Field Name	Example (Hex)
Slave Address	01
Function code	10
Starting Address Hi	E0
Starting Address Lo	A2
Number of Word Hi	00
Number of Word Lo	01
Error check (CRC)	97
	EB

QUERY C (download the first data block)

Field Name	Example (Hex)	
Slave Address	01	
Function code	03	
Starting Address Hi	0A	
Starting Address Lo	00	
Number of Word Hi	00	|
Number of Word Lo	66	|==> (102 words – depends on the number of enabled variables)
Error check (CRC)	C6	
	38	

RESPONSE C

Field Name	Example (Hex)	
Slave Address	01	
Function code	03	
No. of transmitted bytes	CC	
Year Hi	00	|
Year Lo	02	|====> 2002
Month Hi	00	|
Month Lo	04	|====> April
Day Hi	00	|
Day Lo	08	|====> 08
Hour Hi	00	|
Hour Lo	10	|====> 16
Minute Hi	00	|
Minute Lo	00	|====> 00
Second Hi	00	|
Second Lo	00	|====> 00
		|
min/avg/max values		|
(of 4 words) of the variables		|====> 96 words ($60) if 8 variables are selected
selected for recording		|
		|
Error check (CRC)	94	
	89	

QUERY D (prepare the next data block for downloading)

Field Name	Example (Hex)
Slave Address	01
Function code	10
Starting Address Hi	E0
Starting Address Lo	A2
Number of Word Hi	00
Number of Word Lo	01
Byte count	02
Next data Hi	00 |
Next data Lo	02 |==> next block
Error check (CRC)	DF
	1D

Repeat QUERY D and then QUERY C until the instrument replies with the following response:

RESPONSE (if the device has no recorded values in the memory)

Field Name	Example (Hex)
Slave Address	01
Function code	90
Error Code	03
Error check (CRC)	0C
	01

BERG
Energiekontrolle

Energieservice - Optimierungssysteme - Betriebsdatenerfassung
Fraunhofer Straße 22 - D - 82152 Martinsried - Telefon +49 (0)89 379160 - 0 / Telefax +49 (0)89 379160 - 199

http: // www.berg-energie.de - E mail: info@berg-energie.de