MODBUS
Communication Protocol

for energy meters with integrated RS485 interface
C18-45M, C70-100M, C70-5M series

PROTOCOL MANUAL

Ed2601

Limitation of Liability

The Manufacturer reserves the right to modify the specifications in this manual without previous warning. Any copy of this manual, in part or in full, whether by photocopy or by other means, even of electronic nature, without the manufacture giving written authorization, breaches the terms of copyright and is liable to prosecution.

It is absolutely forbidden to use the device for different uses other than those for which it has been devised for, as inferred to in this manual. When using the features in this device, obey all laws and respect privacy and legitimate rights of others.

EXCEPT TO THE EXTENT PROHIBITED BY APPLICABLE LAW, UNDER NO CIRCUMSTANCES SHALL THE MANUFACTURER BE LIABLE FOR CONSEQUENTIAL DAMAGES SUSTAINED IN CONNECTION WITH SAID PRODUCT AND THE MANUFACTURER NEITHER ASSUMES NOR AUTHORIZES ANY REPRESENTATIVE OR OTHER PERSON TO ASSUME FOR IT ANY OBBLIGATION OR LIABILTY OTHER THAN SUCH AS IS EXPRESSLY SET FORTH HEREIN.

The information contained in this manual is for information purposes only, is subject to changes without previous warning and cannot be considered binding for the Manufacturer. The Manufacturer assumes no responsibility for any errors or incoherence possibly contained in this manual.

Index

1. DESCRIPTION

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

Master-slave messages can be:

- Reading (Function codes $03): the communication is between the master and a single slave. It allows to read information about the queried instrument
- Writing (Function code $06, $10): the communication is between the master and a single slave. It allows to change the instrument settings

In a multi-point type connection (MODBUS RTU), slave address (called also logical number) allows to identify each instrument during the communication. Each meter is preset with a default slave address (01) and the user can change it.

Communication frame structure - RTU mode

Bit per byte: 1 Start, 8 Bit, None, 1 Stop (8N1)

Name	Length	Function
START FRAME	4 chars idle	At least 4 character time of silence (MARK condition)
ADDRESS FIELD	8 bits	Instrument logical number
FUNCTION CODE	8 bits	Function code ($03 / $06 / $10)
DATA FIELD	n x 8 bits	Data + length will be filled depending on the message type
ERROR CHECK	16 bits	Error check (CRC)
END FRAME	4 chars idle	At least 4 character time of silence between frames

1.1 CRC Generation

The Cyclical Redundancy Check (CRC) field is two bytes, containing a 16–bit value. The CRC value is calculated by the transmitting device, which appends the CRC to the message. The receiving device recalculates a CRC during receipt of the message, and compares the calculated value to the actual value it received in the CRC field. If the two values are not equal, an error results.

The CRC is started by first preloading a 16–bit register to all 1's. Then a process begins of applying successive 8–bit bytes of the message to the current contents of the register. Only the eight bits of data in each character are used for generating the CRC. Start and stop bits, and the parity bit, do not apply to the CRC.

During generation of the CRC, each 8–bit character is exclusive ORed with the register contents. Then the result is shifted in the direction of the least significant bit (LSB), with a zero filled into the most significant bit (MSB) position. The LSB is extracted and examined. If the LSB was a 1, the register is then exclusive ORed with a preset, fixed value. If the LSB was a 0, no exclusive OR takes place.

This process is repeated until eight shifts have been performed. After the last (eighth) shift, the next 8–bit character is exclusive ORed with the register's current value, and the process repeats for eight more shifts as described above. The final contents of the register, after all the characters of the message have been applied, is the CRC value.

A calculated procedure for generating a CRC is:

1. Load a 16–bit register with $FFFF. Call this the CRC register.
2. Exclusive OR the first 8–bit byte of the message with the low–order byte of the 16–bit CRC register, putting the result in the CRC register.
3. Shift the CRC register one bit to the right (toward the LSB), zero–filling the MSB. Extract and examine the LSB.
4. (If the LSB was 0): Repeat Step 3 (another shift). (If the LSB was 1): Exclusive OR the CRC register with the polynomial value $A001 (1010 0000 0000 0001).
5. Repeat Steps 3 and 4 until 8 shifts have been performed. When this is done, a complete 8–bit byte will have been processed.
6. Repeat Steps 2 through 5 for the next 8–bit byte of the message. Continue doing this until all bytes have been processed.
7. The final contents of the CRC register is the CRC value.
8. When the CRC is placed into the message, its upper and lower bytes must be swapped as described below.

Placing the CRC into the Message

When the 16–bit CRC (two 8–bit bytes) is transmitted in the message, the low-order byte will be transmitted first, followed by the high-order byte.

For example, if the CRC value is $35F7 (0011 0101 1111 0111):

Addr	Func	Data Count	Data	Data	....	Data	CRC lo F7	CRC Hi 35

CRC generation functions - With Table

All of the possible CRC values are preloaded into two arrays, which are simply indexed as the function increments through the message buffer. One array contains all of the 256 possible CRC values for the high byte of the 16–bit CRC field, and the other array contains all of the values for the low byte. Indexing the CRC in this way provides faster execution than would be achieved by calculating a new CRC value with each new character from the message buffer.

```
/*CRC table for calculate with polynom 0xA001 with init value 0xFFFF, High half word*/
rom unsigned char CRC_Table_Hi[] = {
    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
};
/*CRC table for calculate with polynom 0xA001 with init value 0xFFFF, Low half word*/
rom unsigned char  CRC_Table_Lo[] = {
    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
};

unsigned short ModBus_CRC16( unsigned char * Buffer, unsigned short Length )
{
    unsigned char CRCHi = 0xFF;
    unsigned char CRCLo = 0xFF;
    int           Index;
        unsigned short ret;

    while( Length-- )
    {
        Index = CRCLo ^ *Buffer++ ;
        CRCLo = CRCHi ^ CRC_Table_Hi[Index];
        CRCHi = CRC_Table_Lo[Index];
    }
        ret=((unsigned short)CRCHi << 8);
    ret|= (unsigned short)CRCLo;
    return ret;
}
```

CRC generation functions - Without Table

```
unsigned short ModBus_CRC16( unsigned char * Buffer, unsigned short Length )
{
/* ModBus_CRC16 Calculatd CRC16 with polynome 0xA001 and init value 0xFFFF
Input *Buffer - pointer on data
Input Lenght - number byte in buffer
Output - calculated CRC16
*/
        unsigned int cur_crc;

        cur_crc=0xFFFF;
        do
        {
                unsigned int i = 8;
                cur_crc = cur_crc ^ *Buffer++;
                        do
                {
                        if (0x0001 & cur_crc)
                        {
                                cur_crc >>= 1;
                                cur_crc ^= 0xA001;
                        }
                        else
                        {
                                cur_crc >>= 1;
                        }
                }
                while (--i);
        }
        while (--Length);

        return cur_crc;
}
```

2. READING COMMAND STRUCTURE

The master communication device can send commands to the instrument to read its status, setup and the measured values.
More registers can be read, at the same time, sending a single command, only if the registers are consecutive (see chapter 5). The reading command is structured as follows.

2.1 Modbus RTU

Values contained both in Query or Response messages are in hex format.

Query example in case of MODBUS RTU: 01030002000265CB

Example	Byte	Description	No. of bytes
01	-	Slave address	1
03	-	Function code	1
00	High	Starting register	2
02	Low		
00	High	No. of words to be read	2
02	Low		
65	High	Error check (CRC)	2
CB	Low		

Response example in case of MODBUS RTU: 01030400035571F547

Example	Byte	Description	No. of bytes
01	-	Slave address	1
03	-	Function code	1
04	-	Byte count	1
00	High	Requested data: PHASE 2 TO NEUTRAL VOLTAGE VALUE	4
03	Low		
55	High		
71	Low		
F5	High	Error check (CRC)	2
47	Low		

2.2 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.
A round to nearest method is applied to the calculated value of floating point.
The floating-point format is shown as follows:

```
==================================
|  S  |   e + 127   |        f        |
==================================
31      30               23 22                         0 ←
                                                       bit number
```

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.

Example of conversion of value shown with floating point
Value read with floating point:
$45AACC00_{(16)}$

Value converted in binary format:

0 sign	10001011 exponent	$01010101100110000000000_{(2)}$ fraction

sign = 0
exponent = $10001011_{(2)} = 139_{(10)}$
fraction = $01010101100110000000000_{(2)} / 8388608_{(10)} =$
$= 2804736_{(10)} / 8388608_{(10)} = 0.334350585_{(10)}$

$N.n = (-1)^{S}\ 2^{e'-127}\ (1+f) =$
$= (-1)^{0}\ 2^{139-127}\ (1.334350585) =$
$= (+1)\ (4096)\ (1.334350585) =$
$= 5465.5$

3. WRITING COMMAND STRUCTURE

The master communication device can send commands to the instrument to program it.

More settings can be carried out, at the same time, sending a single command, only if the relevant registers are consecutive (see chapter 5). The writing command is structured as follows.

3.1 Modbus RTU

Values contained both in Request or Response messages are in hex format.

Writing commands can be sent with different function codes, according to the register number:
  Function code $06 = write a single register
  Function code $10 = write multiple registers

Query example in case of MODBUS RTU with Function code $06: 0106060000008942

Example	Byte	Description	No. of bytes
01	-	Slave address	1
06	-	Function code	1
06	High	Starting register	2
00	Low		
00	High	Data for programming: Enable setup	2
00	Low		
89	High	Error check (CRC)	2
42	Low		

Response example in case of MODBUS RTU with Function code $06: 0106060000008942

Example	Byte	Description	No. of bytes
01	-	Slave address	1
06	-	Function code	1
06	High	Starting register	2
00	Low		
00	High	Programmed data	2
00	Low		
89	High	Error check (CRC)	2
42	Low		

Query example in case of MODBUS RTU with Function code $10: 0110060000060C000100010001000700000000184DA

Example	Byte	Description	No. of bytes
01	-	Slave address	1
10	-	Function code	1
06	High	Starting register	2
00	Low		
00	High	No. of words to be written	2
06	Low		
0C	-	Data byte counter	1
00	High	Data for programming: Enable setup	2
01	Low		
00	High	Data for programming: Set password to 0001	2
01	Low		
00	High	Data for programming: Set Modbus address to 0001	2
01	Low		
00	High	Data for programming: Set Modbus baudrate to 19200bps	2
07	Low		
00	High	Data for programming: Set Modbus parity to None	2
00	Low		
00	High	Data for programming: Set Modbus stop bit to 1	2
01	Low		
84	High	Error check (CRC)	2
DA	Low		

Response example in case of MODBUS RTU with Function code $10: 0110060000064083

Example	Byte	Description	No. of bytes
01	-	Slave address	1
10	-	Function code	1
06	High	Starting register	2
00	Low		
00	High	No. of written words	2
06	Low		
40	High	Error check (CRC)	2
83	Low		

4. EXCEPTION CODES

When the instrument receives a not-valid query, an error message (exception code) is sent.

The possible exception codes are as follows.

4.1 Modbus RTU

Values contained in Response messages are in hex format.

Response example in case of MODBUS RTU: 01830131F0

Example	Byte	Description	No. of bytes
01	-	Slave address	1
83	-	Function code (80+03)	1
01	-	Exception code	1
31	High	Error check (CRC)	2
F0	Low		

Exception codes for MODBUS RTU are following described:

$01 **ILLEGAL FUNCTION:** the function code received in the query is not an allowable action.
$02 **ILLEGAL DATA ADDRESS:** the data address received in the query is not an allowable address (i.e. the combination of register and transfer length is invalid).
$03 **ILLEGAL DATA VALUE:** a value contained in the query data field is not an allowable value.
$04 **ILLEGAL RESPONSE LENGTH:** the request would generate a response with size bigger than that available for MODBUS protocol.

5. GENERAL INFORMATION ON REGISTER TABLES

NOTE: The register values are in hex format ($).

NOTE: Possibility to read/write a big number of registers with a single command (max 125 registers).

Table HEADER	Meaning
PARAMETER	Symbol and description of the parameter to be read/written.
+/-	The parameter can take negative value. **2's Complement Mode**: If this column is checked, the read register value can have positive or negative sign. The negative values are represented with 2's complement.
INTEGER	INTEGER register data. It shows the Address in hex format, the corresponding Word number and the Unit of measure (if any).
IEEE	IEEE standard register data. It shows the Address in hex format, the corresponding Word number and the Unit of measure (if any).
MODEL	Availability of the parameter for the corresponding model: **1ph 45A:** 45A 1phase meters **3ph 100A:** 100A 3phase meters **3ph CT:** CT 3phase meters If checked with ●, the parameter is available. If not checked, the parameter is not available but the register is readable and it replies $FFFF.
DATA MEANING	Description of data received by a response of a reading command.
PROGRAMMABLE DATA	Description of data which can be sent for a writing command.

6. READING REGISTERS (FUNCTION CODE $03)

REALTIME VALUES

PARAMETER		+/-	INTEGER			IEEE			MODEL		
Symbol	Description	Signed	Words	Address	Unit of measure	Words	Address	Unit of measure	1ph 45A	3ph 100A	3ph CT
U1N	Phase 1 to Neutral Voltage		2	0000	0.001V	2	1000	V		●	●
U2N	Phase 2 to Neutral Voltage		2	0002	0.001V	2	1002	V		●	●
U3N	Phase 3 to Neutral Voltage		2	0004	0.001V	2	1004	V		●	●
V12	Phase 1 to Phase 2 Voltage		2	0006	0.001V	2	1006	V		●	●
V23	Phase 2 to Phase 3 Voltage		2	0008	0.001V	2	1008	V		●	●
V31	Phase 3 to Phase 1 Voltage		2	000A	0.001V	2	100A	V		●	●
V∑	Phase to Phase Voltage Average		2	000C	0.001V	2	100C	V	●	●	●
I1	Phase 1 Current		2	000E	0.001A	2	100E	A		●	●
I2	Phase 2 Current		2	0010	0.001A	2	1010	A		●	●
I3	Phase 3 Current		2	0012	0.001A	2	1012	A		●	●
IN	Neutral Current (calculated)		2	0014	0.001A	2	1014	A		●	●
I∑	Phase Currents Average (system current)		2	0016	0.001A	2	1016	A	●	●	●
±PF1	Phase 1 Power Factor	●	1	0018	x1000	2	1018	-		●	●
±PF2	Phase 2 Power Factor	●	1	0019	x1000	2	101A	-		●	●
±PF3	Phase 3 Power Factor	●	1	001A	x1000	2	101C	-		●	●
±PF∑	System Power Factor	●	1	001B	x1000	2	101E	-	●	●	●
±P1	Phase 1 Active Power	●	3	001C	0.001W	2	1020	W		●	●
±P2	Phase 2 Active Power	●	3	001F	0.001W	2	1022	W		●	●
±P3	Phase 3 Active Power	●	3	0022	0.001W	2	1024	W		●	●
±P∑	System Active Power (sum of phases)	●	3	0025	0.001W	2	1026	W	●	●	●
S1	Phase 1 Apparent Power		3	0028	0.001VA	2	1028	VA		●	●
S2	Phase 2 Apparent Power		3	002B	0.001VA	2	102A	VA		●	●
S3	Phase 3 Apparent Power		3	002E	0.001VA	2	102C	VA		●	●
S∑	System Apparent Power (sum of phases)		3	0031	0.001VA	2	102E	VA	●	●	●
±Q1	Phase 1 Reactive Power	●	3	0034	0.001var	2	1030	var		●	●
±Q2	Phase 2 Reactive Power	●	3	0037	0.001var	2	1032	var		●	●
±Q3	Phase 3 Reactive Power	●	3	003A	0.001var	2	1034	var		●	●
±Q∑	System Reactive Power (sum of phases)	●	3	003D	0.001var	2	1036	var	●	●	●
F	System Frequency		1	0040	0.001Hz	2	1038	Hz	●	●	●
PhSequence	Phase rotation sequence (on voltages) $00=123-CCW, $01=132-CW, $02=not defined		1	0041	-	2	103A	-		●	●
U∑	Phase to Neutral Voltage Average		2	0042	0.001V	2	103C	V		●	●
∑ I1+I2+I3	Phase Currents sum		2	0044	0.001A	2	103E	A		●	●
U1N THD	Phase 1 to Neutral Voltage THD		2	0046	0.001%	2	1040	%			
U2N THD	Phase 2 to Neutral Voltage THD		2	0048	0.001%	2	1042	%			
U3N THD	Phase 3 to Neutral Voltage THD		2	004A	0.001%	2	1044	%			
-	Reserved		2	004C	-	2	1046	-			
V12 THD	Phase 1 to Phase 2 Voltage THD		2	004E	0.001%	2	1048	%			
V23 THD	Phase 2 to Phase 3 Voltage THD		2	0050	0.001%	2	104A	%			
-	Reserved		4	0052	-	4	104C	-			
I1 THD	Phase 1 Current THD		2	0056	0.001%	2	1050	%			
I2 THD	Phase 2 Current THD (N/A in 3.3.2 wiring)		2	0058	0.001%	2	1052	%			
I3 THD	Phase 3 Current THD		2	005A	0.001%	2	1054	%			
-	Reserved		2	005C	-	2	1056	-			
ANG U1N-I1	Phase 1 Voltage-Current Angle		2	005E	0.001°	2	1058	°		●	●
ANG U2N-I2	Phase 2 Voltage-Current Angle		2	0060	0.001°	2	105A	°		●	●
ANG U3N-I3	Phase 3 Voltage-Current Angle		2	0062	0.001°	2	105C	°		●	●
AVG ANG ULN-IL	Phase Voltage-Current Angles Average		2	0064	0.001°	2	105E	°		●	●

REALTIME DMD AND DMD MAX VALUES

PARAMETER		+/-	INTEGER			IEEE			MODEL		
Symbol	**Description**	**Signed**	**Words**	**Address**	**Unit of measure**	**Words**	**Address**	**Unit of measure**	**1ph 45A**	**3ph 100A**	**3ph CT**
I1 DMD	Phase 1 Current DMD		2	00A2	0.001A	2	1094	A	●	●	●
I2 DMD	Phase 2 Current DMD		2	00A4	0.001A	2	1096	A		●	●
I3 DMD	Phase 3 Current DMD		2	00A6	0.001A	2	1098	A		●	●
IN DMD	Neutral Current DMD		2	00A8	0.001A	2	109A	A		●	●
+P∑ DMD	Positive System Active Power DMD		3	00AA	0.001W	2	109C	W			
-P∑ DMD	Negative System Active Power DMD	●	3	00AD	0.001W	2	109E	W			
±P∑ DMD	System Active Power DMD (calculated on +P and -P)	●	3	00B0	0.001W	2	10A0	W	●	●	●
S∑ DMD	System Apparent Power DMD		3	00B3	0.001VA	2	10A2	VA		●	●
±Q∑ DMD	System Reactive Power DMD (calculated on +Q and -Q)	●	3	00B6	0.001var	2	10A4	var		●	●
I1 DMD MAX	Phase 1 Current DMD MAX		2	00B9	0.001A	2	10A6	A		●	●
I2 DMD MAX	Phase 2 Current DMD MAX		2	00BB	0.001A	2	10A8	A		●	●
I3 DMD MAX	Phase 3 Current DMD MAX		2	00BD	0.001A	2	10AA	A		●	●
IN DMD MAX	Neutral Current DMD MAX		2	00BF	0.001A	2	10AC	A			
+P∑ DMD MAX	Positive System Active Power DMD MAX		3	00C1	0.001W	2	10AE	W			
-P∑ DMD MAX	Negative System Active Power DMD MAX	●	3	00C4	0.001W	2	10B0	W			
±P∑ DMD MAX	System Active Power DMD MAX	●	3	00C7	0.001W	2	10B2	W	●	●	●
S∑ DMD MAX	System Apparent Power DMD MAX		3	00CA	0.001VA	2	10B4	VA		●	●
±Q∑ DMD MAX	System Reactive Power DMD MAX	●	3	00CD	0.001var	2	10B6	var		●	●

TOTAL COUNTERS

PARAMETER		+/-	INTEGER			IEEE			MODEL		
Symbol	**Description**	**Signed**	**Words**	**Address**	**Unit of measure**	**Words**	**Address**	**Unit of measure**	**1ph 45A**	**3ph 100A**	**3ph CT**
+Ea1	Phase 1 IMP Active Energy		3	0100	Wh	2	1100	Wh		●	●
+Ea2	Phase 2 IMP Active Energy		3	0103	Wh	2	1102	Wh		●	●
+Ea3	Phase 3 IMP Active Energy		3	0106	Wh	2	1104	Wh		●	●
+Ea∑	System IMP Active Energy		3	0109	Wh	2	1106	Wh	●	●	●
-Ea1	Phase 1 EXP Active Energy		3	010C	Wh	2	1108	Wh		●	●
-Ea2	Phase 2 EXP Active Energy		3	010F	Wh	2	110A	Wh		●	●
-Ea3	Phase 3 EXP Active Energy		3	0112	Wh	2	110C	Wh		●	●
-Ea∑	System EXP Active Energy		3	0115	Wh	2	110E	Wh	●	●	●
-	Reserved		96	0118	-	64	1110	-			
+Er1 (IND+CAP)	Phase 1 IMP Reactive Energy		3	0178	varh	2	1150	varh		●	●
+Er2 (IND+CAP)	Phase 2 IMP Reactive Energy		3	017B	varh	2	1152	varh		●	●
+Er3 (IND+CAP)	Phase 3 IMP Reactive Energy		3	017E	varh	2	1154	varh		●	●
-Er1 (IND+CAP)	Phase 1 EXP Reactive Energy		3	0181	varh	2	1156	varh		●	●
-Er2 (IND+CAP)	Phase 2 EXP Reactive Energy		3	0184	varh	2	1158	varh		●	●
-Er3 (IND+CAP)	Phase 3 EXP Reactive Energy		3	0187	varh	2	115A	varh		●	●
+Er∑ (IND+CAP)	System IMP Reactive Energy		3	018A	varh	2	115C	varh	●	●	●
-Er∑ (IND+CAP)	System EXP Reactive Energy		3	018D	varh	2	115E	varh	●	●	●
Es∑	System Apparent Energy		3	0190	VAh	2	1160	VAh	●	●	●
MHrCnt	Measure Hour Counter		3	0193	0.1h	2	1162	0.1h		●	●

PARTIAL AND BALANCE COUNTERS

PARAMETER		+/-	INTEGER			IEEE			MODEL		
Symbol	**Description**	**Signed**	**Words**	**Address**	**Unit of measure**	**Words**	**Address**	**Unit of measure**	**1ph 45A**	**3ph 100A**	**3ph CT**
+Ea∑ PAR	System IMP Active Energy PARTIAL		3	0400	Wh	2	1400	Wh	●	●	●
-Ea∑ PAR	System EXP Active Energy PARTIAL		3	0403	Wh	2	1402	Wh	●	●	●
-	Reserved		9	0406	-	6	1404	-			
Es∑ PAR	System Apparent Energy PARTIAL		3	040F	VAh	2	140A	VAh	●	●	●
-	Reserved		6	0412	-	4	140C	-			
+Er∑ PAR IMP	System IMP Reactive Energy PARTIAL		3	0418	varh	2	1410	varh	●	●	●
-Er∑ PAR EXP	System EXP Reactive Energy PARTIAL		3	041B	varh	2	1412	varh	●	●	●

PARTIAL AND BALANCE COUNTERS

PARAMETER		+/-	INTEGER			IEEE			MODEL		
Symbol	Description	Signed	Words	Address	Unit of measure	Words	Address	Unit of measure	1ph 45A	3ph 100A	3ph CT
±Ea∑ BAL	System Active Energy BALANCE (IMP-EXP)	●	3	041E	Wh	2	1414	Wh		●	●
-	Reserved		12	0421	-	8	1416	-			
±Er∑ BAL	System Reactive Energy BALANCE (IMP-EXP)	●	3	042D	varh	2	141E	varh		●	●
±Ea1 BAL	Phase 1 Active Energy BALANCE (IMP-EXP)	●	3	0430	Wh	2	1420	Wh			
±Ea2 BAL	Phase 2 Active Energy BALANCE (IMP-EXP)	●	3	0433	Wh	2	1422	Wh			
±Ea3 BAL	Phase 3 Active Energy BALANCE (IMP-EXP)	●	3	0436	Wh	2	1424	Wh			
±Er1 BAL	Phase 1 Reactive Energy BALANCE (IMP-EXP)	●	3	0439	varh	2	1426	varh			
±Er2 BAL	Phase 2 Reactive Energy BALANCE (IMP-EXP)	●	3	043C	varh	2	1428	varh			
±Er3 BAL	Phase 3 Reactive Energy BALANCE (IMP-EXP)	●	3	043F	varh	2	142A	varh			

TARIFF 1 COUNTERS

PARAMETER		+/-	INTEGER			IEEE			MODEL		
Symbol	Description	Signed	Words	Address	Unit of measure	Words	Address	Unit of measure	1ph 45A	3ph 100A	3ph CT
-	Reserved		9	0200	-	2	1200	-			
+Ea∑ T1	System IMP Active Energy TARIFF1		3	0209	Wh	2	1206	Wh	●	●	●
-	Reserved		12	020C	-	8	1208	-			
-Ea∑ T1	System EXP Active Energy TARIFF1		3	0215	Wh	2	120E	Wh	●	●	●
-	Reserved		96	0218	-	64	1210	-			
+Er∑ T1	System IMP Reactive Energy TARIFF1		3	0278	varh	2	1250	varh	●	●	●
-Er∑ T1	System EXP Reactive Energy TARIFF1		3	027B	varh	2	1252	varh	●	●	●
-	Reserved		3	027E	-	2	1254	-			
Es∑ T1	System Apparent Energy TARIFF1		3	0281	VAh	2	1256	VAh	●	●	●

TARIFF 2 COUNTERS

PARAMETER		+/-	INTEGER			IEEE			MODEL		
Symbol	Description	Signed	Words	Address	Unit of measure	Words	Address	Unit of measure	1ph 45A	3ph 100A	3ph CT
-	Reserved		9	0300	-	2	1300	-			
+Ea∑ T2	System IMP Active Energy TARIFF2		3	0309	Wh	2	1306	Wh	●	●	●
-	Reserved		12	030C	-	8	1308	-			
-Ea∑ T2	System EXP Active Energy TARIFF2		3	0315	Wh	2	130E	Wh	●	●	●
-	Reserved		96	0318	-	64	1310	-			
+Er∑ T2	System IMP Reactive Energy TARIFF2		3	0378	varh	2	1350	varh	●	●	●
-Er∑ T2	System EXP Reactive Energy TARIFF2		3	037B	varh	2	1352	varh	●	●	●
-	Reserved		3	037E	-	2	1354	-			
Es∑ T2	System Apparent Energy TARIFF2		3	0381	VAh	2	1356	VAh	●	●	●

TARIFF 3 COUNTERS

PARAMETER		+/-	INTEGER			IEEE			MODEL		
Symbol	Description	Signed	Words	Address	Unit of measure	Words	Address	Unit of measure	1ph 45A	3ph 100A	3ph CT
-	Reserved		9	2100	-	2	1500	-			
+Ea∑ T3	System IMP Active Energy TARIFF3		3	2109	Wh	2	1506	Wh	●	●	●
-	Reserved		12	210C	-	8	1508	-			

TARIFF 3 COUNTERS

PARAMETER		+/-	INTEGER			IEEE			MODEL		
Symbol	Description	Signed	Words	Address	Unit of measure	Words	Address	Unit of measure	1ph 45A	3ph 100A	3ph CT
-Ea∑ T3	System EXP Active Energy TARIFF3		3	2115	Wh	2	150E	Wh	●	●	●
-	Reserved		96	2118	-	64	1510	-			
+Er∑ T3	System IMP Reactive Energy TARIFF3		3	2178	varh	2	1550	varh	●	●	●
-Er∑ T3	System EXP Reactive Energy TARIFF3		3	217B	varh	2	1552	varh	●	●	●
-	Reserved		3	217E	-	2	1554	-			
Es∑ T3	System Apparent Energy TARIFF3		3	2181	VAh	2	1556	VAh	●	●	●

TARIFF 4 COUNTERS

PARAMETER		+/-	INTEGER			IEEE			MODEL		
Symbol	Description	Signed	Words	Address	Unit of measure	Words	Address	Unit of measure	1ph 45A	3ph 100A	3ph CT
-	Reserved		9	2200	-	2	1600	-			
+Ea∑ T4	System IMP Active Energy TARIFF4		3	2209	Wh	2	1606	Wh	●	●	●
-	Reserved		12	220C	-	8	1608	-			
-Ea∑ T4	System EXP Active Energy TARIFF4		3	2215	Wh	2	160E	Wh	●	●	●
-	Reserved		96	2218	-	64	1610	-			
+Er∑ T4	System IMP Reactive Energy TARIFF4		3	2278	varh	2	1650	varh	●	●	●
-Er∑ T4	System EXP Reactive Energy TARIFF4		3	227B	varh	2	1652	varh	●	●	●
-	Reserved		3	227E	-	2	1654	-			
Es∑ T4	System Apparent Energy TARIFF4		3	2281	VAh	2	1656	VAh	●	●	●

INSTRUMENT INFO

PARAMETER		INTEGER		DATA MEANING	MODEL		
Symbol	Description	Words	Address	Values	1ph 45A	3ph 100A	3ph CT
SN-LOT	Serial number and Lot number	5	0500	The value is structured as follows: word1(MSw)-2=Serial number (SN) word3-4=Lot number (LOT, fixed value) word5(LSw)=not used Convert the read Hex value in Dec value. e.g. $0E4E1BFF0007A1200000: SN=$0E4E1BFF => 239999999 LOT=$0007A120 => 500000 (fixed value)	●	●	●
MODEL	Model	1	0505	$20=1ph 45A $21=1ph 100A $22=3ph 100A $23=3ph 4ML $25=3ph RGW $26=3ph CT $27=3ph 2ML	●	●	●
TYPE	Type	1	0506	$02=1phase meter $09=3phase meter	●	●	●
FW REL	Firmware release	1	0507	Major & minor version of Firmware release. Convert the read Hex value in Dec value. e.g. $0D80=3456 major FW=34, minor FW=56 => 12.34.56 FW on display	●	●	●
HW VER	Hardware version	1	0508	e.g. $0101 => ver. 1.01	●	●	●
OEM	OEM code	2	0509	Fixed to $414C474F	●	●	●
TAR	Tariff in use	1	050B	$01=Tariff 1 $02=Tariff 2 $03=Tariff 3 $04=Tariff 4	●	●	●
PRI/SEC	Primary/Secondary value in use	1	050C	$00=Primary (fixed for direct meters) $01=Secondary (for 3phCT only)	●	●	●
ERR	Error code	1	050D	$00=No error $01=Memory error $02=Calibration parameter error $03=Metrology parameter error	●	●	●
-	Reserved	15	050E	-			
SIGN REPR	Sign representation method	1	051D	$00=Sign bit	●	●	●
-	Reserved	6	051E	-			
CHK	Checksum	2	0524	8 Hex characters	●	●	●
ADDR	Modbus Address	1	0602	$01…$F7	●	●	●
BAUD	Modbus Baudrate	1	0603	$04=2400 bps $05=4800 bps $06=9600 bps $07=19200 bps $08=38400 bps	●	●	●
PARITY	Modbus Parity	1	0604	$00=None $01=Odd $02=Even	●	●	●
STOP BIT	Modbus Stop bit	1	0605	$01=1 $02=2	●	●	●
CT PRI (CH1)	CT or Rogowski coil primary	1	0606	Convert the read Hex value in Dec value. e.g. $270F=9999A			●
CT SEC	CT or Rogowski coil secondary	1	0607	$00=1A $01=5A $09=100mV (fixed for 3phML & 3phRGW only)	●	●	●
PT PRI	PT primary	1	0608	Convert the read Hex value in Dec value. e.g. $270F=9999V			●
PT SEC	PT secondary	1	0609	Convert the read Hex value in Dec value. e.g. $01F4=500V			●
I DIR	Current direction	1	060A	Bit field coding: bit15(MSb)...bit12 not used, always 0 bit11=I3 CH4 for 3phML only bit10=I2 CH4 for 3phML only bit9=I1 CH4 for 3phML only bit8=I3 CH3 for 3phML only bit7=I2 CH3 for 3phML only bit6=I1 CH3 for 3phML only bit5=I3 CH2 for 3phML only bit4=I2 CH2 for 3phML only bit3=I1 CH2 for 3phML only bit2=I3 bit1=I2 bit0(LSb)=I1 Bit=0 means Forward current (FWD) Bit=1 means Reverse current (REV)			●

INSTRUMENT INFO

PARAMETER		INTEGER		DATA MEANING	MODEL		
Symbol	**Description**	**Words**	**Address**	**Values**	**1ph 45A**	**3ph 100A**	**3ph CT**
WIR	Wiring mode	1	060B	\$01=3phases, 4 wires, 3 currents \$02=3phases, 3 wires, 2 currents \$03=1phase, 2 wires, 1 current \$04=3phases, 3 wires, 3 currents	●	●	●
DMD INT	Demand integration time	1	060C	\$05=5 min \$08=8 min \$0A=10 min \$0F=15 min \$14=20 min \$1E=30 min \$3C=60 min	●	●	●
S01-EN	Pulse output 1 energy type	1	060D	\$01=+Ea∑ \$02=-Ea∑ \$03=+Er∑ \$04=-Er∑	●	●	●
S01-RATE	Pulse output 1 rate	1	060E	\$01=0.001 En/imp \$02=0.01 En/imp \$03=0.1 En/imp \$04=1 En/imp \$05=10 En/imp \$06=100 En/imp \$07=1000 En/imp	●	●	●
S01-DUR	Pulse output 1 duration	1	060F	\$3C=60 s \$64=100 s \$C8=200 s	●	●	●
BCKLIT	Backlight time	1	0610	\$00=always ON \$01=1 min \$05=5 min \$0A=10 min \$1E=30 min \$3C=60 min \$78=120 min		●	●
MHrCntThr	Measure Hour counter threshold	1	0613	Threshold value conversion changes according to the model. **For 100A model:** Convert the read Hex value in Dec value and multiply it by 0.01 e.g.: \$1388 = 5000 => 50 **For CT model:** Convert the read Hex value in Dec value and multiply it by one of the following value according to the CT primary: 0.005 if CT primary is <10 0.01 if CT primary is <100 0.1 if CT primary is <1000 1 if CT primary is <10000 e.g. if CT primary is <10: \$0258 = 600 => 3		●	●

7. WRITING REGISTERS (FUNCTION CODES $06, $10)

To set the instrument, first enable the setup by sending the password (writing register $0600), then change settings by sending the desired writing command within 60 s.
If register $0600 is not sent or is sent uncorrectly, the next writing command has no effect and the instrument will reply with an exception code.

INSTRUMENT SETUP

PARAMETER		INTEGER		PROGRAMMABLE DATA	MODEL		
Symbol	Description	Words	Address	Values	1ph 45A	3ph 100A	3ph CT
SETUP EN	Enable setup by entering password	1	0600	Send setup password (default $03E8) to enable the writing command sending. After that, it is allowed to send writing commands within 60 s.		●	●
SETUP PW CHANGE	Setup password change	1	0601	$0000 … $270F		●	●
ADDR	Modbus Address	1	0602	$01 … $F7	●	●	●
BAUD	Modbus Baudrate	1	0603	$04=2400 bps $05=4800 bps $06=9600 bps $07=19200 bps $08=38400 bps	●	●	●
PARITY	Modbus Parity	1	0604	$00=None $01=Odd $02=Even	●	●	●
STOP BIT	Modbus Stop bit	1	0605	$01=1 $02=2	●	●	●
DMD INT	Demand integration time	1	060C	$05=5 min $08=8 min $0A=10 min $0F=15 min $14=20 min $1E=30 min $3C=60 min	●	●	●
S01-EN	Pulse output 1 energy type	1	060D	$01=+Ea∑ $02=-Ea∑ $03=+Er∑ $04=-Er∑	●	●	●
S01-RATE	Pulse output 1 rate	1	060E	$01= 0.001 En/imp $02=0.01 En/imp $03=0.1 En/imp $04=1 En/imp $05=10 En/imp $06=100 En/imp $07=1000 En/imp	●	●	●
S01-DUR	Pulse output 1 duration	1	060F	$3C=60 s $64=100 s $C8=200 s	●	●	●
BCKLIT	Backlit time	1	0610	$00=Always ON $01=1 min $05=5 min $0A=10 min $1E=30 min $3C=60 min $78=120 min		●	●
RES MD/mM	Reset MAX DMD and MIN/MAX values	1	0611	$01=Reset all MAX DMD values $02=Reset all MIN/MAX values (for 3phML only)	●	●	●
RES PAR	Reset Partial counter values	1	0612	$01=Reset all partial counters	●	●	●
MHrCntThr	Measure Hour counter threshold	1	0613	It must be set considering that threshold range values change according to the model. **For 100A model:** Ist … Imax (0.04 … 99.99) To set threshold, first divide the Dec value to be programmed by 0.01, then convert it in Hex value. e.g.: 50 = 5000 => $1388 **For CT model:** Ist*CT ratio … CT primary To set threshold, first divide the Dec value to be programmed by one of the following value according to the CT primary: 0.005 if CT primary is <10 0.01 if CT primary is <100 0.1 if CT primary is <1000 1 if CT primary is <10000 Then convert it in Hex value. e.g. if CT primary is <10: 3 = 600 => $0258		●	●
TAR SWITCH	Tariff switch (available if Tariff Control is set on COM port)	1	0617	$01=Tariff 1 $02=Tariff 2 $03=Tariff 3 $04=Tariff 4	●	●	●