
MODBUS communication protocol

Protocollo di comunicazione MODBUS

for MCIMIDMDB - MCIMIDETH modules
per moduli MCIMIDMDB - MCIMIDETH

User manual
Manuale d'uso

MODBUS communication protocol

MCIMIDMDB - MCIMIDETH

MULTILINGUAL MANUAL

March Edition 2012

Limitation of liability

The Manufacturer reserves the right to modify the software or the software 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 authorisation, breaches the terms of copyright and is liable to prosecution.

It is absolutely forbidden to use the software 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 software, 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 OBLIGATION OR LIABILITY OTHER THAN SUCH AS IS EXPRESSLY SET FORTH HEREIN.

All trademarks in this manual are property of their respective owners.

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.

Printed in Italy.

Limitazione di responsabilità

Il Produttore si riserva il diritto di modificare, senza preavviso, il software o le specifiche del software illustrate in questo manuale. Qualunque duplicazione del manuale, parziale o totale, non autorizzata per iscritto dal Produttore, ottenuta per fotocopiatura o con altri sistemi, anche di natura elettronica, viola le condizioni di copyright ed è giuridicamente perseguibile.

E' assolutamente proibito utilizzare il software per usi differenti da quelli per cui è stato costruito, desumibili dal contenuto del presente manuale. Durante l'uso delle funzioni del presente software, assicurarsi di rispettare tutte le leggi, nonché la privacy ed i diritti altrui.

ECCEP TO PER I LIMITI IMPOSTI DALLA LEGGE, IN NESSUN CASO IL PRODUTTORE SARA' RESPONSABILE PER DANNI DERIVANTI DAL PRODOTTO, NE' SI ASSUME O AUTORIZZA ALCUN RAPPRESENTANTE O ALTRA PERSONA AD ASSUMERSI QUALUNQUE OBBLIGO O RESPONSABILITA' DIVERSE DA QUELLE DICHIARATE ESPRESSAMENTE SOPRA.

Tutti i marchi, citati in questo manuale, sono proprietà dei rispettivi possessori.

Le informazioni contenute in questo manuale hanno unicamente scopo informativo, sono soggette a variazioni senza preavviso e non potranno venire considerate impegnative per il Produttore. Il Produttore non assume alcuna responsabilità per eventuali errori o incoerenze che possano essere contenuti nel manuale.

Stampato in Italia.

USER MANUAL

English

Index

1. Description.....	7
1.1 LRC generation.....	8
1.2 CRC generation.....	9
2. Read commands structure	12
2.1 MODBUS ASCII/RTU	12
2.2 MODBUS TCP	13
2.3 Floating point as per IEEE Standard.....	14
3. Write commands structure	15
3.1 MODBUS ASCII/RTU	15
3.2 MODBUS TCP	16
4. Exception codes.....	17
4.1 MODBUS ASCII/RTU	17
4.2 MODBUS TCP	17
5. Register tables	18
5.1 Reading registers (Function code \$01 / \$03 / \$04).....	19
5.2 Writing registers (Function code \$10)	26

1. Description

MODBUS ASCII/RTU is a master-slave communication protocol, able to support up to 247 slaves connected in 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.

MODBUS TCP is a variant of the MODBUS family. Specifically, it covers the use of MODBUS messaging in an "Intranet" or "Internet" environment using the TCP/IP protocol on a fixed port **502**.

Master-slave messages can be:

- **Reading (Function code \$01 / \$03 / \$04)**: the communication is between the master and a single slave. It allows to read information about the queried counter
- **Writing (Function code \$10)**: the communication is between the master and a single slave. It allows to change the counter settings
- **Broadcast (not available for MODBUS TCP)**: the communication is between the master and all the connected slaves. It is always a write command (Function code \$10) and required logical number \$00

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

In case of MODBUS TCP, slave address is replaced by a single byte, the **Unit identifier**.

COMMUNICATION FRAME STRUCTURE

ASCII mode

Bit per byte: 1 Start, 7 Bit, Even, 1 Stop (7E1)

Name	Length	Function
START FRAME	1 char	Message start marker. Starts with colon ":" (\$3A)
ADDRESS FIELD	2 chars	Counter logical number
FUNCTION CODE	2 chars	Function code (\$01 / \$03 / \$04 / \$10)
DATA FIELD	n chars	Data + length will be filled depending on the message type
ERROR CHECK	2 chars	Error check (LRC)
END FRAME	2 chars	Carriage return - line feed (CRLF) pair (\$0D & \$0A)

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	Counter logical number
FUNCTION CODE	8 bits	Function code (\$01 / \$03 / \$04 / \$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

TCP mode

Bit per byte: 1 Start, 7 Bit, Even, 2 Stop (7E2)

Name	Length	Function
TRANSACTION ID	2 bytes	For synchronization between messages of server & client
PROTOCOL ID	2 bytes	Zero for MODBUS TCP
BYTE COUNT	2 bytes	Number of remaining bytes in this frame
UNIT ID	1 byte	Slave address (255 if not used)
FUNCTION CODE	1 byte	Function code (\$01 / \$04 / \$10)
DATA BYTES	n bytes	Data as response or command

1.1 LRC generation

The Longitudinal Redundancy Check (LRC) field is one byte, containing an 8-bit binary value. The LRC value is calculated by the transmitting device, which appends the LRC to the message. The receiving device recalculates an LRC during receipt of the message, and compares the calculated value to the actual value it received in the LRC field. If the two values are not equal, an error results. The LRC is calculated by adding together successive 8-bit bytes in the message, discarding any carries, and then two's complementing the result. The LRC is an 8-bit field, therefore each new addition of a character that would result in a value higher than 255 decimal simply 'rolls over' the field's value through zero. Because there is no ninth bit, the carry is discarded automatically.

A procedure for generating an LRC is:

1. Add all bytes in the message, excluding the starting 'colon' and ending CR LF. Add them into an 8-bit field, so that carries will be discarded.
2. Subtract the final field value from \$FF, to produce the ones-complement.
3. Add 1 to produce the twos-complement.

PLACING THE LRC INTO THE MESSAGE

When the the 8-bit LRC (2 ASCII characters) is transmitted in the message, the high-order character will be transmitted first, followed by the low-order character. For example, if the LRC value is \$52 (0101 0010):

Colon '.'	Addr	Func	Data Count	Data	Data	Data	LRC Hi '5'	LRC Lo '2'	CR	LF
--------------	------	------	---------------	------	------	------	------	---------------	---------------	----	----

C-function calculate LRC

*pucFrame – pointer on "Addr" of message

usLen – length message from "Addr" to end "Data"

```

UCHAR prvucMBLRC( UCHAR * pucFrame, USHORT usLen )
{
    UCHAR          ucLRC = 0; /* LRC char initialized */

    while( usLen-- )
    {
        ucLRC += *pucFrame++; /* Add buffer byte without carry */
    }

    /* Return twos complement */
    ucLRC = ( UCHAR ) ( -( ( CHAR ) ucLRC ) );
    return ucLRC;
}

```


1.2 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
------	------	---------------	------	------	------	------	--------------	--------------


```

        ret=((unsigned short)CRCHi << 8);
    ret|= (unsigned short)CRCLo;
    return ret;
}

```

OR CAN BE USED CALCULATED METHOD WITHOUT TABLE

```

/*      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
*/

/*
    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 (--Lenght);
    return cur_crc;
*/

```

2. Read commands structure

The master communication device can send commands to the module to read its status and setup or the measured values, status and setup relevant to the combined counter. More registers can be read, at the same time, sending a single command, only if the registers are consecutive (see chapter 5).

According to the used MODBUS protocol mode, the read command is structured as follows.

2.1 MODBUS ASCII/RTU

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

Query example in case of MODBUS ASCII/RTU: 010300020002D00B

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		
D0	High	Error check (CRC)	2
0B	Low		

Response example in case of MODBUS ASCII/RTU: 01030400035571F4F0

Example	Byte	Description	No. of bytes
01	-	Slave address	1
03	-	Function code	1
04	-	Byte count	1
00	High	Requested data	4
03	Low		
55	High		
71	Low		
F4	High	Error check (CRC)	2
F0	Low		

2.2 MODBUS TCP

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

Query example in case of MODBUS TCP: 010000000006010400020002

Example	Byte	Description	No. of bytes
01	-	Transaction identifier	1
00	High	Protocol identifier	4
00	Low		
00	High		
00	Low		
06	-	Byte count	1
01	-	Unit identifier	1
04	-	Function code	1
00	High	Starting register	2
02	Low		
00	High	No. of words to be read	2
02	Low		

Response example in case of MODBUS TCP: 01000000000701040400035571

Example	Byte	Description	No. of bytes
01	-	Transaction identifier	1
00	High	Protocol identifier	4
00	Low		
00	High		
00	Low		
07	-	Byte count	1
01	-	Unit identifier	1
04	-	Function code	1
04	-	No. of byte of requested data	2
00	High	Requested data	4
03	Low		
55	High		
71	Low		

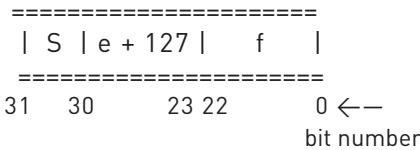
2.3 Floating point as per IEEE Standard (not available for MODBUS TCP)

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:



where:

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

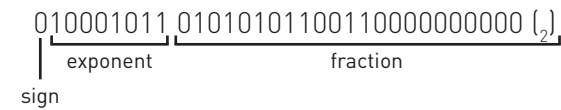


NOTE
Fractions (decimals) are always shown while the leading 1 (hidden bit) is not stored.

EXAMPLE OF CONVERSION OF VALUE SHOWN WITH FLOATING POINT

Value read with floating point:
45AACC00₍₁₆₎

Value converted in binary format:



sign = 0
exponent = 10001011₍₂₎ = 139₍₁₀₎
fraction = 010101011001100000000000₍₂₎ / 8388608₍₁₀₎ =
= 2804736₍₁₀₎ / 8388608₍₁₀₎ = 0.334350585₍₁₀₎

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

3. Write commands structure

The master communication device can send commands to the module to program itself or the combined counter. More settings can be carried out, at the same time, sending a single command, only if the relevant registers are consecutive (see chapter 5).

According to the used MODBUS protocol type, the write command is structured as follows.

3.1 MODBUS ASCII/RTU

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

Query example in case of MODBUS ASCII/RTU: 011005150001020008F053

Example	Byte	Description	No. of bytes
01	-	Slave address	1
10	-	Function code	1
05	High	Starting register	2
15	Low		
00	High	No. of words to be written	2
01	Low		
02	-	Data byte counter	1
00	High	Data for programming	2
08	Low		
F0	High	Error check (CRC)	2
53	Low		

Response example in case of MODBUS ASCII/RTU: 01100515000110C1

Example	Byte	Description	No. of bytes
01	-	Slave address	1
10	-	Function code	1
05	High	Starting register	2
15	Low		
00	High	No. of written words	2
01	Low		
10	High	Error check (CRC)	2
C1	Low		

3.2 MODBUS TCP

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

Query example in case of MODBUS TCP: 010000000009011005150001020008

Example	Byte	Description	No. of bytes
01	-	Transaction identifier	1
00	High	Protocol identifier	4
00	Low		
00	High		
00	Low		
09	-	Byte count	1
01	-	Unit identifier	1
10	-	Function code	1
05	High	Starting register	2
15	Low		
00	High	No. of words to be written	2
01	Low		
02	-	Data byte counter	1
00	High	Data for programming	2
08	Low		

Response example in case of MODBUS TCP: 010000000006011005150001

Example	Byte	Description	No. of bytes
01	-	Transaction identifier	1
00	High	Protocol identifier	4
00	Low		
00	High		
00	Low		
06	-	Byte count	1
01	-	Unit identifier	1
10	-	Function code	1
05	High	Starting register	2
15	Low		
00	High	Command successfully sent	2
01	Low		

4. Exception codes

When the module receives a not-valid query, an error message (exception code) is sent. According to the used MODBUS protocol mode, possible exception codes are as follows.

4.1 MODBUS ASCII/RTU

Values contained in Response messages are in hex format.

Response example in case of MODBUS ASCII/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 ASCII/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.

4.2 MODBUS TCP

Values contained in Response messages are in hex format.

Response example in case of MODBUS TCP: 010000000003018302

Example	Byte	Description	No. of bytes
01	-	Transaction identifier	1
00	High	Protocol identifier	4
00	Low		
00	High		
00	Low		
03	-	No. of byte of next data in this string	1
01	-	Unit identifier	1
83	-	Function code [80+03]	1
02	-	Exception code	1

Exception codes for MODBUS TCP are following described:

- \$01 ILLEGAL FUNCTION:** the function code is unknown by the server.
- \$02 ILLEGAL DATA ADDRESS:** the data address received in the query is not an allowable address for the combined counter (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 for the combined counter.
- \$04 SERVER FAILURE:** the server failed during the execution.
- \$05 ACKNOWLEDGE:** the server accepted the server invocation but the service requires a relatively long time to execute. The server therefore returns only an acknowledgement of the service invocation receipt.
- \$06 SERVER BUSY:** the server was unable to accept the MB request PDU. The client application has the responsibility of deciding if and when re-sending the request.
- \$0A GATEWAY PATH UNAVAILABLE:** the communication module is not configured or cannot communicate.
- \$0B GATEWAY TARGET DEVICE FAILED TO RESPOND:** the counter is not available in the network.

5. Register tables



NOTE

Highest number of registers (or bytes) which can be read with a single command:

- in ASCII mode: 63 registers
- in RTU mode: 127 registers
- in TCP mode: 256 bytes



NOTE

Highest number of registers which can be programmed with a single command:

- in ASCII mode: 13 registers
- in RTU mode: 29 registers
- in TCP mode: 1 register



NOTE

The register values are in hex format (\$).



NOTE

IEEE registers cannot be used in case of MODBUS TCP protocol.

TABLE HEADER	MEANING
Parameter	Measuring parameter to be read
Register description	Description of the register to be read / written
F. code (Hex)	Function code in hex format. It identifies the command type (reading / writing)
Sign	<p>If this column is checked, the read register value can have positive or negative sign. Convert a signed register value as shown in the following instructions:</p> <p>The Most Significant Bit (MSB) indicates the sign as follows: 0=positive (+), 1=negative (-). <u>NEGATIVE VALUE EXAMPLE:</u></p> <div style="text-align: center;"> MSB $\\$8020 = 1000000000100000 = -32$ <div style="display: flex; justify-content: space-around; width: 100%;"> HEX BIN DEC </div> </div>
INTEGER	Details for INTEGER type registers
IEEE	Details for IEEE standard type registers
Register (Hex)	Register address in hex format
Words	Number of word to be read / written for the register (length)
M.U.	Measuring unit of parameter
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

5.1 Reading registers (Function code \$01 / \$03 / \$04)

Parameter	F. code (Hex)	Sign	INTEGER			IEEE not available for MODBUS TCP		
			Register (Hex)	Words	M.U.	Register (Hex)	Words	M.U.

REAL TIME VALUES

V1 • L-N voltage phase 1	03/04		0000	2	mV	1000	2	V
V2 • L-N voltage phase 2	03/04		0002	2	mV	1002	2	V
V3 • L-N voltage phase 3	03/04		0004	2	mV	1004	2	V
V12 • L-L voltage line 12	03/04		0006	2	mV	1006	2	V
V23 • L-L voltage line 23	03/04		0008	2	mV	1008	2	V
V31 • L-L voltage line 31	03/04		000A	2	mV	100A	2	V
V Σ • System voltage	03/04		000C	2	mV	100C	2	V
A1 • Phase 1 current	03/04	X	000E	2	mA	100E	2	A
A2 • Phase 2 current	03/04	X	0010	2	mA	1010	2	A
A3 • Phase 3 current	03/04	X	0012	2	mA	1012	2	A
AN • Neutral current	03/04	X	0014	2	mA	1014	2	A
A Σ • System current	03/04	X	0016	2	mA	1016	2	A
PF1 • Phase 1 power factor	03/04	X	0018	1	-	1018	2	-
PF2 • Phase 2 power factor	03/04	X	0019	1	-	101A	2	-
PF3 • Phase 3 power factor	03/04	X	001A	1	-	101C	2	-
PF Σ • System power factor	03/04	X	001B	1	-	101E	2	-
P1 • Phase 1 active power	03/04	X	001C	3	mW	1020	2	W
P2 • Phase 2 active power	03/04	X	001F	3	mW	1022	2	W
P3 • Phase 3 active power	03/04	X	0022	3	mW	1024	2	W
P Σ • System active power	03/04	X	0025	3	mW	1026	2	W
S1 • Phase 1 apparent power	03/04	X	0028	3	mVA	1028	2	VA
S2 • Phase 2 apparent power	03/04	X	002B	3	mVA	102A	2	VA
S3 • Phase 3 apparent power	03/04	X	002E	3	mVA	102C	2	VA
S Σ • System apparent power	03/04	X	0031	3	mVA	102E	2	VA
Q1 • Phase 1 reactive power	03/04	X	0034	3	mvar	1030	2	var
Q2 • Phase 2 reactive power	03/04	X	0037	3	mvar	1032	2	var
Q3 • Phase 3 reactive power	03/04	X	003A	3	mvar	1034	2	var
Q Σ • System reactive power	03/04	X	003D	3	mvar	1036	2	var
F • Frequency	03/04		0040	1	mHz	1038	2	Hz
Phase sequence [\$00=123-CCW, \$01=321-CW]	03/04		0041	1	-	103A	2	-

TOTAL COUNTER VALUES

+kWh1 • Phase 1 imported active energy	03/04		0100	3	0.1 Wh	1100	2	Wh
+kWh2 • Phase 2 imported active energy	03/04		0103	3	0.1 Wh	1102	2	Wh
+kWh3 • Phase 3 imported active energy	03/04		0106	3	0.1 Wh	1104	2	Wh
+kWh Σ • System imported active energy	03/04		0109	3	0.1 Wh	1106	2	Wh
-kWh1 • Phase 1 exported active energy	03/04		010C	3	0.1 Wh	1108	2	Wh
-kWh2 • Phase 2 exported active energy	03/04		010F	3	0.1 Wh	110A	2	Wh
-kWh3 • Phase 3 exported active energy	03/04		0112	3	0.1 Wh	110C	2	Wh
-kWh Σ • System exported active energy	03/04		0115	3	0.1 Wh	110E	2	Wh
+kVAh1-L • Phase 1 imported lagging apparent energy	03/04		0118	3	0.1 VAh	1110	2	VAh
+kVAh2-L • Phase 2 imported lagging apparent energy	03/04		011B	3	0.1 VAh	1112	2	VAh
+kVAh3-L • Phase 3 imported lagging apparent energy	03/04		011E	3	0.1 VAh	1114	2	VAh
+kVAh Σ -L • System imported lagging apparent energy	03/04		0121	3	0.1 VAh	1116	2	VAh

Parameter	F. code (Hex)	Sign	INTEGER			IEEE not available for MODBUS TCP		
			Register (Hex)	Words	M.U.	Register (Hex)	Words	M.U.

TOTAL COUNTER VALUES

-kVAh1-L • Phase 1 exported lagging apparent energy	03/04		0124	3	0.1 VAh	1118	2	VAh
-kVAh2-L • Phase 2 exported lagging apparent energy	03/04		0127	3	0.1 VAh	111A	2	VAh
-kVAh3-L • Phase 3 exported lagging apparent energy	03/04		012A	3	0.1 VAh	111C	2	VAh
-kVAh Σ -L • System exported lagging apparent energy	03/04		012D	3	0.1 VAh	111E	2	VAh
+kVAh1-C • Phase 1 imported leading apparent energy	03/04		0130	3	0.1 VAh	1120	2	VAh
+kVAh2-C • Phase 2 imported leading apparent energy	03/04		0133	3	0.1 VAh	1122	2	VAh
+kVAh3-C • Phase 3 imported leading apparent energy	03/04		0136	3	0.1 VAh	1124	2	VAh
+kVAh Σ -C • System imported leading apparent energy	03/04		0139	3	0.1 VAh	1126	2	VAh
-kVAh1-C • Phase 1 exported leading apparent energy	03/04		013C	3	0.1 VAh	1128	2	VAh
-kVAh2-C • Phase 2 exported leading apparent energy	03/04		013F	3	0.1 VAh	112A	2	VAh
-kVAh3-C • Phase 3 exported leading apparent energy	03/04		0142	3	0.1 VAh	112C	2	VAh
-kVAh Σ -C • System exported leading apparent energy	03/04		0145	3	0.1 VAh	112E	2	VAh
+kvarh1-L • Phase 1 imported lagging reactive energy	03/04		0148	3	0.1 varh	1130	2	varh
+kvarh2-L • Phase 2 imported lagging reactive energy	03/04		014B	3	0.1 varh	1132	2	varh
+kvarh3-L • Phase 3 imported lagging reactive energy	03/04		014E	3	0.1 varh	1134	2	varh
+kvarh Σ -L • System imported lagging reactive energy	03/04		0151	3	0.1 varh	1136	2	varh
-kvarh1-L • Phase 1 exported lagging reactive energy	03/04		0154	3	0.1 varh	1138	2	varh
-kvarh2-L • Phase 2 exported lagging reactive energy	03/04		0157	3	0.1 varh	113A	2	varh
-kvarh3-L • Phase 3 exported lagging reactive energy	03/04		015A	3	0.1 varh	113C	2	varh
-kvarh Σ -L • System exported lagging reactive energy	03/04		015D	3	0.1 varh	113E	2	varh
+kvarh1-C • Phase 1 imported leading reactive energy	03/04		0160	3	0.1 varh	1140	2	varh
+kvarh2-C • Phase 2 imported leading reactive energy	03/04		0163	3	0.1 varh	1142	2	varh
+kvarh3-C • Phase 3 imported leading reactive energy	03/04		0166	3	0.1 varh	1144	2	varh
+kvarh Σ -C • System imported leading reactive energy	03/04		0169	3	0.1 varh	1146	2	varh
-kvarh1-C • Phase 1 exported leading reactive energy	03/04		016C	3	0.1 varh	1148	2	varh
-kvarh2-C • Phase 2 exported leading reactive energy	03/04		016F	3	0.1 varh	114A	2	varh
-kvarh3-C • Phase 3 exported leading reactive energy	03/04		0172	3	0.1 varh	114C	2	varh
-kvarh Σ -C • System exported leading reactive energy	03/04		0175	3	0.1 varh	114E	2	varh

TARIFF 1 COUNTER VALUES

+kWh1 • Phase 1 imported active energy	03/04		0200	3	0.1 Wh	1200	2	Wh
+kWh2 • Phase 2 imported active energy	03/04		0203	3	0.1 Wh	1202	2	Wh
+kWh3 • Phase 3 imported active energy	03/04		0206	3	0.1 Wh	1204	2	Wh
+kWh Σ • System imported active energy	03/04		0209	3	0.1 Wh	1206	2	Wh
-kWh1 • Phase 1 exported active energy	03/04		020C	3	0.1 Wh	1208	2	Wh
-kWh2 • Phase 2 exported active energy	03/04		020F	3	0.1 Wh	120A	2	Wh
-kWh3 • Phase 3 exported active energy	03/04		0212	3	0.1 Wh	120C	2	Wh
-kWh Σ • System exported active energy	03/04		0215	3	0.1 Wh	120E	2	Wh
+kVAh1-L • Phase 1 imported lagging apparent energy	03/04		0218	3	0.1 VAh	1210	2	VAh
+kVAh2-L • Phase 2 imported lagging apparent energy	03/04		021B	3	0.1 VAh	1212	2	VAh
+kVAh3-L • Phase 3 imported lagging apparent energy	03/04		021E	3	0.1 VAh	1214	2	VAh
+kVAh Σ -L • System imported lagging apparent energy	03/04		0221	3	0.1 VAh	1216	2	VAh
-kVAh1-L • Phase 1 exported lagging apparent energy	03/04		0224	3	0.1 VAh	1218	2	VAh
-kVAh2-L • Phase 2 exported lagging apparent energy	03/04		0227	3	0.1 VAh	121A	2	VAh
-kVAh3-L • Phase 3 exported lagging apparent energy	03/04		022A	3	0.1 VAh	121C	2	VAh
-kVAh Σ -L • System exported lagging apparent energy	03/04		022D	3	0.1 VAh	121E	2	VAh
+kVAh1-C • Phase 1 imported leading apparent energy	03/04		0230	3	0.1 VAh	1220	2	VAh

Parameter	F. code (Hex)	Sign	INTEGER			IEEE not available for MODBUS TCP		
			Register (Hex)	Words	M.U.	Register (Hex)	Words	M.U.

TARIFF 1 COUNTER VALUES

+kVAh2-C • Phase 2 imported leading apparent energy	03/04		0233	3	0.1 VAh	1222	2	VAh
+kVAh3-C • Phase 3 imported leading apparent energy	03/04		0236	3	0.1 VAh	1224	2	VAh
+kVAh Σ -C • System imported leading apparent energy	03/04		0239	3	0.1 VAh	1226	2	VAh
-kVAh1-C • Phase 1 exported leading apparent energy	03/04		023C	3	0.1 VAh	1228	2	VAh
-kVAh2-C • Phase 2 exported leading apparent energy	03/04		023F	3	0.1 VAh	122A	2	VAh
-kVAh3-C • Phase 3 exported leading apparent energy	03/04		0242	3	0.1 VAh	122C	2	VAh
-kVAh Σ -C • System exported leading apparent energy	03/04		0245	3	0.1 VAh	122E	2	VAh
+kvarh1-L • Phase 1 imported lagging reactive energy	03/04		0248	3	0.1 varh	1230	2	varh
+kvarh2-L • Phase 2 imported lagging reactive energy	03/04		024B	3	0.1 varh	1232	2	varh
+kvarh3-L • Phase 3 imported lagging reactive energy	03/04		024E	3	0.1 varh	1234	2	varh
+kvarh Σ -L • System imported lagging reactive energy	03/04		0251	3	0.1 varh	1236	2	varh
-kvarh1-L • Phase 1 exported lagging reactive energy	03/04		0254	3	0.1 varh	1238	2	varh
-kvarh2-L • Phase 2 exported lagging reactive energy	03/04		0257	3	0.1 varh	123A	2	varh
-kvarh3-L • Phase 3 exported lagging reactive energy	03/04		025A	3	0.1 varh	123C	2	varh
-kvarh Σ -L • System exported lagging reactive energy	03/04		025D	3	0.1 varh	123E	2	varh
+kvarh1-C • Phase 1 imported leading reactive energy	03/04		0260	3	0.1 varh	1240	2	varh
+kvarh2-C • Phase 2 imported leading reactive energy	03/04		0263	3	0.1 varh	1242	2	varh
+kvarh3-C • Phase 3 imported leading reactive energy	03/04		0266	3	0.1 varh	1244	2	varh
+kvarh Σ -C • System imported leading reactive energy	03/04		0269	3	0.1 varh	1246	2	varh
-kvarh1-C • Phase 1 exported leading reactive energy	03/04		026C	3	0.1 varh	1248	2	varh
-kvarh2-C • Phase 2 exported leading reactive energy	03/04		026F	3	0.1 varh	124A	2	varh
-kvarh3-C • Phase 3 exported leading reactive energy	03/04		0272	3	0.1 varh	124C	2	varh
-kvarh Σ -C • System exported leading reactive energy	03/04		0275	3	0.1 varh	124E	2	varh

TARIFF 2 COUNTER VALUES

+kWh1 • Phase 1 imported active energy	03/04		0300	3	0.1 Wh	1300	2	Wh
+kWh2 • Phase 2 imported active energy	03/04		0303	3	0.1 Wh	1302	2	Wh
+kWh3 • Phase 3 imported active energy	03/04		0306	3	0.1 Wh	1304	2	Wh
+kWh Σ • System imported active energy	03/04		0309	3	0.1 Wh	1306	2	Wh
-kWh1 • Phase 1 exported active energy	03/04		030C	3	0.1 Wh	1308	2	Wh
-kWh2 • Phase 2 exported active energy	03/04		030F	3	0.1 Wh	130A	2	Wh
-kWh3 • Phase 3 exported active energy	03/04		0312	3	0.1 Wh	130C	2	Wh
-kWh Σ • System exported active energy	03/04		0315	3	0.1 Wh	130E	2	Wh
+kVAh1-L • Phase 1 imported lagging apparent energy	03/04		0318	3	0.1 VAh	1310	2	VAh
+kVAh2-L • Phase 2 imported lagging apparent energy	03/04		031B	3	0.1 VAh	1312	2	VAh
+kVAh3-L • Phase 3 imported lagging apparent energy	03/04		031E	3	0.1 VAh	1314	2	VAh
+kVAh Σ -L • System imported lagging apparent energy	03/04		0321	3	0.1 VAh	1316	2	VAh
-kVAh1-L • Phase 1 exported lagging apparent energy	03/04		0324	3	0.1 VAh	1318	2	VAh
-kVAh2-L • Phase 2 exported lagging apparent energy	03/04		0327	3	0.1 VAh	131A	2	VAh
-kVAh3-L • Phase 3 exported lagging apparent energy	03/04		032A	3	0.1 VAh	131C	2	VAh
-kVAh Σ -L • System exported lagging apparent energy	03/04		032D	3	0.1 VAh	131E	2	VAh
+kVAh1-C • Phase 1 imported leading apparent energy	03/04		0330	3	0.1 VAh	1320	2	VAh
+kVAh2-C • Phase 2 imported leading apparent energy	03/04		0333	3	0.1 VAh	1322	2	VAh
+kVAh3-C • Phase 3 imported leading apparent energy	03/04		0336	3	0.1 VAh	1324	2	VAh
+kVAh Σ -C • System imported leading apparent energy	03/04		0339	3	0.1 VAh	1326	2	VAh
-kVAh1-C • Phase 1 exported leading apparent energy	03/04		033C	3	0.1 VAh	1328	2	VAh
-kVAh2-C • Phase 2 exported leading apparent energy	03/04		033F	3	0.1 VAh	132A	2	VAh

Parameter	F. code (Hex)	Sign	INTEGER			IEEE not available for MODBUS TCP		
			Register (Hex)	Words	M.U.	Register (Hex)	Words	M.U.

TARIFF 2 COUNTER VALUES

-kVAh3-C • Phase 3 exported leading apparent energy	03/04		0342	3	0.1 VAh	132C	2	VAh
-kVAh Σ -C • System exported leading apparent energy	03/04		0345	3	0.1 VAh	132E	2	VAh
+kvarh1-L • Phase 1 imported lagging reactive energy	03/04		0348	3	0.1 varh	1330	2	varh
+kvarh2-L • Phase 2 imported lagging reactive energy	03/04		034B	3	0.1 varh	1332	2	varh
+kvarh3-L • Phase 3 imported lagging reactive energy	03/04		034E	3	0.1 varh	1334	2	varh
+kvarh Σ -L • System imported lagging reactive energy	03/04		0351	3	0.1 varh	1336	2	varh
-kvarh1-L • Phase 1 exported lagging reactive energy	03/04		0354	3	0.1 varh	1338	2	varh
-kvarh2-L • Phase 2 exported lagging reactive energy	03/04		0357	3	0.1 varh	133A	2	varh
-kvarh3-L • Phase 3 exported lagging reactive energy	03/04		035A	3	0.1 varh	133C	2	varh
-kvarh Σ -L • System exported lagging reactive energy	03/04		035D	3	0.1 varh	133E	2	varh
+kvarh1-C • Phase 1 imported leading reactive energy	03/04		0360	3	0.1 varh	1340	2	varh
+kvarh2-C • Phase 2 imported leading reactive energy	03/04		0363	3	0.1 varh	1342	2	varh
+kvarh3-C • Phase 3 imported leading reactive energy	03/04		0366	3	0.1 varh	1344	2	varh
+kvarh Σ -C • System imported leading reactive energy	03/04		0369	3	0.1 varh	1346	2	varh
-kvarh1-C • Phase 1 exported leading reactive energy	03/04		036C	3	0.1 varh	1348	2	varh
-kvarh2-C • Phase 2 exported leading reactive energy	03/04		036F	3	0.1 varh	134A	2	varh
-kvarh3-C • Phase 3 exported leading reactive energy	03/04		0372	3	0.1 varh	134C	2	varh
-kvarh Σ -C • System exported leading reactive energy	03/04		0375	3	0.1 varh	134E	2	varh

PARTIAL COUNTER VALUES

+kWh Σ • System imported active energy	03/04		0400	3	0.1 Wh	1400	2	Wh
-kWh Σ • System exported active energy	03/04		0403	3	0.1 Wh	1402	2	Wh
+kVAh Σ -L • System imported lagging apparent energy	03/04		0406	3	0.1 VAh	1404	2	VAh
-kVAh Σ -L • System exported lagging apparent energy	03/04		0409	3	0.1 VAh	1406	2	VAh
+kVAh Σ -C • System imported leading apparent energy	03/04		040C	3	0.1 VAh	1408	2	VAh
-kVAh Σ -C • System exported leading apparent energy	03/04		040F	3	0.1 VAh	140A	2	VAh
+kvarh Σ -L • System imported lagging reactive energy	03/04		0412	3	0.1 varh	140C	2	varh
-kvarh Σ -L • System exported lagging reactive energy	03/04		0415	3	0.1 varh	140E	2	varh
+kvarh Σ -C • System imported leading reactive energy	03/04		0418	3	0.1 varh	1410	2	varh
-kvarh Σ -C • System exported leading reactive energy	03/04		041B	3	0.1 varh	1412	2	varh

BALANCE VALUES

kWh Σ • System active energy	03/04	X	041E	3	0.1 Wh	1414	2	Wh
kVAh Σ -L • System lagging apparent energy	03/04	X	0421	3	0.1 VAh	1416	2	VAh
kVAh Σ -C • System leading apparent energy	03/04	X	0424	3	0.1 VAh	1418	2	VAh
kvarh Σ -L • System lagging reactive energy	03/04	X	0427	3	0.1 varh	141A	2	varh
kvarh Σ -C • System leading reactive energy	03/04	X	042A	3	0.1 varh	141C	2	varh

Register description	F. code (Hex)	INTEGER		Data meaning
		Register (Hex)	Words	
COUNTER & COMMUNICATION MODULE DATA				
Serial number	03/04	0500	5	10 ASCII chars. (\$00÷\$FF)
Model	03/04	0505	1	\$03=6A 3phases/4wires \$06=6A 3phases/3wires \$08=80A 3phases/4wires \$0A=80A 3phases/3wires \$0C=80A 1phase/2wires
Type	03/04	0506	1	\$02=MID
Firmware release	03/04	0507	1	Convert the read Hex value in Decimal value. e.g. \$66=102 =rel. 1.02
Hardware version	03/04	0508	1	Convert the read Hex value in Decimal value. e.g. \$64=100 =rev. 1.00
Reserved	03/04	0509	2	
Tariff in use	03/04	050B	1	\$01=tariff 1 \$02=tariff 2
Primary/secondary value	03/04	050C	1	\$00=primary \$01=secondary
Error code	03/04	050D	1	\$00=none \$01=phase sequence error
CT value (only for combined counter 6A 3phase model)	03/04	050E	1	\$0001÷\$2710
Reserved	03/04	050F	2	
FSA value	03/04	0511	1	\$00=1A \$01=5A \$02=80A
Wiring mode	03	0512	1	\$01=3-phases/4 wires/3 currents \$02=3-phases/3 wires/2 currents \$03=1-phase
MODBUS address (not available for LAN GATEWAY)	03	0513	1	\$01÷\$F7
MODBUS mode (not available for LAN GATEWAY)	03	0514	1	\$00=7E2 (ASCII) \$01=8N1 (RTU)
Communication speed (not available for LAN GATEWAY)	03	0515	1	\$01=300 bps \$02=600 bps \$03=1200 bps \$04=2400 bps \$05=4800 bps \$06=9600 bps \$07=19200 bps \$08=38400 bps \$09=57600 bps \$0A=115200 bps

Register description	F. code (Hex)	INTEGER		Data meaning
		Register (Hex)	Words	

COUNTER & COMMUNICATION MODULE DATA

Partial counters status	03	0517	1	<p>Convert the read Hex value in Binary. e.g. \$0003= 0000000000000011</p> <p>Each bit corresponds to the status of a partial counter. 0=inactive 1=active</p> <p>0000000000000011 ←</p> <p>Start to read bit string following the arrow. The first bit corresponds to the status of the first counter in the list:</p> <ul style="list-style-type: none"> 1) +kWhΣ PAR 2) -kWhΣ PAR 3) +kVAhΣ-L PAR 4) -kVAhΣ-L PAR 5) +kVAhΣ-C PAR 6) -kVAhΣ-C PAR 7) +kvarhΣ-L PAR 8) -kvarhΣ-L PAR 9) +kvarhΣ-C PAR 10) -kvarhΣ-C PAR <p>The last six bits of the string are reserved.</p> <p>In the example, only +kWhΣ PAR and -kWhΣ PAR counters are active.</p>
-------------------------	----	------	---	---

COMMUNICATION MODULE DATA

Serial number	03/04	0518	5	10 ASCII chars. {\$00÷\$FF}
Reserved	03/04	051D	2	
Firmware release	03/04	051F	1	<p>Convert the read Hex value in Decimal value. e.g. \$66=102 =rel. 1.02</p>
Hardware version	03/04	0520	1	<p>Convert the read Hex value in Decimal value. e.g. \$64=100 =rev. 1.00</p>

Register description	F. code (Hex)	Register (Hex)	Data meaning
COILS			
Alarm events	01	0000	<p>40 coils</p> <p>Byte 1 - voltage out of range UV3 UV2 UV1 UVΣ OV3 OV2 OV1 OVΣ </p> <p>Byte 2 - line voltage out of range COM RES UV23 UV31 UV12 OV23 OV31 OV12 </p> <p>Byte 3/4 - current out of range RES RES RES RES RES RES UIN UI3 UI2 UI1 UIΣ OIN OI3 OI2 OI1 OIΣ </p> <p>Byte 5 - frequency out of range RES RES RES RES RES RES RES F </p> <p>LEGEND UV=undervoltage OV=overvoltage UI=undercurrent OI=overcurrent F=frequency out of range COM=communication in progress RES=reserved bit to 0</p> <p>NOTE: the voltage, current and frequency threshold values can change according to the combined counter model. Please refer to the table shown below.</p>

COUNTER NOMINAL VOLTAGE	PARAMETER THRESHOLDS			
	PHASE VOLTAGE available only for 2-4 wire model counters	LINE VOLTAGE not available for 2 wire model counter	CURRENT	FREQUENCY
A	UV _{L-N} : Vnom -20% OV _{L-N} : Vnom +20%	UV _{L-L} : Vnom * $\sqrt{3}$ -20% OV _{L-L} : Vnom * $\sqrt{3}$ +20%	UI: Start current value (Ist) OI: Full scale value (FS)	F low: 45Hz F high: 65Hz
B				
C	UV _{L-N} : 230V -20% OV _{L-N} : 240V +20%	UV _{L-L} : 400V -20% OV _{L-L} : 415V +20%		
D				

5.2 Writing registers (Function code \$10)

Register description	F. code (Hex)	INTEGER		Programmable data
		Register (Hex)	Words	
COUNTER & COMMUNICATION MODULE DATA				
MODBUS address (not available for LAN GATEWAY)	10	0513	1	\$01÷\$F7
MODBUS mode (not available for LAN GATEWAY)	10	0514	1	\$00=7E2 (ASCII) \$01=8N1 (RTU)
Communication speed (not available for LAN GATEWAY)	10	0515	1	\$01=300 bps \$02=600 bps \$03=1200 bps \$04=2400 bps \$05=4800 bps \$06=9600 bps \$07=19200 bps \$08=38400 bps \$09=57600 bps \$0A=115200 bps
Reserved	10	0516	1	
Partial counters status	10	0517	1	Byte 1 - partial counter selection: \$00=+kWh∑ PAR \$01=-kWh∑ PAR \$02=+kVAh∑-L PAR \$03=-kVAh∑-L PAR \$04=+kVAh∑-C PAR \$05=-kVAh∑-C PAR \$06=+kvarh∑-L PAR \$07=-kvarh∑-L PAR \$08=+kvarh∑-C PAR \$09=-kvarh∑-C PAR \$0A=all partial counters Byte 2 - partial counter/s operation: \$01=start \$02=stop \$03=reset e.g. start +kWh∑ PAR counter 00=+kWh∑ PAR 01=start final value to be set: 0001



NOTE

\$0513, \$0514, \$0515 writing registers allow to program the module communication parameters.

MANUALE D'USO

Italiano

Sommario

1. Descrizione	31
1.1 Generazione LRC	32
1.2 Generazione CRC	33
2. Struttura comandi di lettura.....	36
2.1 MODBUS ASCII/RTU	36
2.2 MODBUS TCP	37
2.3 Virgola mobile secondo lo standard IEEE	37
3. Struttura comandi di scrittura.....	39
3.1 MODBUS ASCII/RTU	39
3.2 MODBUS TCP	40
4. Codici di errore.....	41
4.1 MODBUS ASCII/RTU	41
4.2 MODBUS TCP	41
5. Tabelle dei registri	42
5.1 Registri di lettura (Codice di funzione \$01 / \$03 / \$04) ..	43
5.2 Registri di scrittura (Codice di funzione \$10)	50

1. Descrizione

MODBUS ASCII/RTU è un protocollo di comunicazione master-slave in grado di supportare fino a 247 slave organizzati in forma di bus o di rete a stella.

Il protocollo usa una connessione simplex su una singola linea. In questo modo, i messaggi di comunicazione si muovono in due direzioni diverse su una stessa linea.

MODBUS TCP è una variante della famiglia MODBUS. Nello specifico, svolge lo scambio di messaggi MODBUS in ambiente "Intranet" o "Internet" utilizzando il protocollo TCP/IP su porta fissa **502**.

I messaggi master-slave possono essere:

- **Lettura (Codice di funzione \$01 / \$03 / \$04)**: la comunicazione avviene tra il master ed un solo slave. Consente di leggere informazioni sul contatore interrogato
- **Scrittura (Codice di funzione \$10)**: la comunicazione avviene tra il master ed un solo slave. Consente di cambiare le impostazioni del contatore
- **Broadcast (non disponibile per MODBUS TCP)**: la comunicazione avviene tra il master e tutti gli slave connessi. E' sempre un comando di scrittura (Codice di funzione \$10) e richiede come numero logico \$00

In una connessione di tipo multi-point (MODBUS ASCII/RTU), lo **slave address** (detto anche **numero logico**) consente di identificare ogni contatore durante la comunicazione. Ogni contatore è preimpostato con uno slave address di default (01) e l'utente può modificarlo.

In caso di MODBUS TCP, lo slave address è sostituito da un singolo byte, lo **Unit identifier**.

STRUTTURA DI UN FRAME DI COMUNICAZIONE

Modalità ASCII

Bit per byte: 1 Start, 7 Bit, Even, 1 Stop (7E1)

Nome	Lunghezza	Funzione
START FRAME	1 car.	Segnale di inizio messaggio. Inizia con due punti ":" (\$3A)
CAMPO INDIRIZZO	2 car.	Numero logico contatore
CODICE DI FUNZIONE	2 car.	Codice di funzione (\$01 / \$03 / \$04 / \$10)
CAMPO DATI	n car.	Dati + la lunghezza cambia in relazione al tipo di messaggio
ERROR CHECK	2 car.	Controllo errori (LRC)
END FRAME	2 car.	Carriage return - line feed (CRLF) (\$0D & \$0A)

Modalità RTU

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

Nome	Lunghezza	Funzione
START FRAME	4 car. idle	Il tempo di silenzio deve durare almeno 4 caratteri (MARK condition)
CAMPO INDIRIZZO	8 bit	Numero logico contatore
CODICE DI FUNZIONE	8 bit	Codice di funzione (\$01 / \$03 / \$04 / \$10)
CAMPO DATI	n x 8 bit	Dati + la lunghezza cambia in relazione al tipo di messaggio
ERROR CHECK	16 bit	Controllo errori (CRC)
END FRAME	4 car. idle	Il tempo di silenzio tra i frame deve durare almeno 4 caratteri

Modalità TCP

Bit per byte: 1 Start, 7 Bit, Even, 2 Stop (7E2)

Nome	Lunghezza	Funzione
TRANSACTION ID	2 byte	Per la sincronizzazione tra i messaggi server & client
PROTOCOL ID	2 byte	Zero per MODBUS TCP
BYTE COUNT	2 byte	Numero di byte rimanenti in questo frame
UNIT ID	1 byte	Slave address (255 se non utilizzato)
CODICE DI FUNZIONE	1 byte	Codice di funzione (\$01 / \$04 / \$10)
BYTE DI DATI	n byte	Dati come risposta o comando

1.1 Generazione LRC

Il campo "Longitudinal Redundancy Check" (LRC) è composto da un byte, contenente un valore binario codificato su 8 bit. Il valore LRC è calcolato dal dispositivo di trasmissione che lo pone nel messaggio. Il dispositivo ricevente calcola a sua volta il valore LRC durante la ricezione del messaggio, e lo confronta con il valore presente nel campo LRC. Se i due valori non sono uguali, viene segnalato un errore. Per il calcolo LRC occorre sommare tutti i campi che compongono il frame tra di loro con una somma ad 8 bit senza riporto, il risultato ottenuto andrà poi espresso in complemento a 2. LRC è un campo da 8 bit, pertanto ogni carattere aggiunto che potrebbe risultare in un valore decimale maggiore di 255, porterebbe il valore del campo completamente a zero. Dato che non esiste un nono bit, il riporto viene eliminato automaticamente.

La procedura per generare il valore LRC è la seguente:

1. Aggiungere tutti i bytes nel messaggio, ad esclusione dei due punti iniziali e dei CR LF finali. Aggiungerli in un campo da 8 bit, così da eliminare il riporto.
2. Sottrarre il valore del campo finale da \$FF, per produrre il complemento a 2.
3. Aggiungere 1 per produrre il complemento a 2.

INSERIRE IL VALORE LRC NEL MESSAGGIO

Quando il valore LRC da 8 bit (2 caratteri ASCII) viene trasmesso nel messaggio, viene inviato prima il carattere high seguito poi dal carattere low. Per esempio, se il valore LRC è \$52 (0101 0010):

Colon '.'	Addr	Func	Data Count	Data	Data	Data	LRC Hi '5'	LRC Lo '2'	CR	LF
--------------	------	------	---------------	------	------	------	------	---------------	---------------	----	----

C-function calculate LRC

*pucFrame – pointer on "Addr" of message

usLen – length message from "Addr" to end "Data"

```
UCHAR prvucMBLRC( UCHAR * pucFrame, USHORT usLen )
{
    UCHAR          ucLRC = 0; /* LRC char initialized */

    while( usLen-- )
    {
        ucLRC += *pucFrame++; /* Add buffer byte without carry */
    }

    /* Return twos complement */
    ucLRC = ( UCHAR ) ( -( ( CHAR ) ucLRC ) );
    return ucLRC;
}
```


1.2 Generazione CRC

Il campo "Cyclical Redundancy Check" (CRC) è composto da 2 byte, contenente un valore binario codificato su 16 bit. Il valore CRC è calcolato dal dispositivo di trasmissione che lo pone nel messaggio. Il dispositivo ricevente calcola a sua volta il valore CRC durante la ricezione del messaggio, e lo confronta con il valore presente nel campo CRC. Se i due valori non sono uguali, viene segnalato un errore.

Per generare il CRC occorre prima di tutto precaricare un registro da 16 bit tutti a 1. Poi verrà avviata l'elaborazione per l'applicazione dei successivi bytes (da 8 bit) del messaggio al contenuto corrente del registro. In ogni carattere solo 8 bit di dati sono utilizzati per generare il CRC. Non vengono applicati al CRC i bit di start e stop e il bit di parità. Durante la generazione del CRC, per ogni carattere da 8 bit viene effettuato il calcolo XOR con i contenuti del registro. Successivamente il risultato viene spostato nella direzione del bit meno significativo (LSB= least significant bit), con uno zero inserito alla posizione del bit più significativo (MSB= most significant bit). LSB viene estratto ed esaminato. Se LSB era 1, al registro viene effettuato il calcolo XOR con valore fisso preimpostato. Se LSB era 0, non viene effettuato nessun calcolo XOR. Questo processo viene ripetuto fino a quando si raggiungono 8 spostamenti. Dopo l'ultimo spostamento (l'ottavo), al carattere a 8 bit successivo viene effettuato il calcolo XOR con il valore di registro corrente, e il processo ripete nuovamente altri 8 spostamenti come precedentemente descritto. Il valore CRC corrisponderà al contenuto finale del registro, dopo che tutti i caratteri del messaggio sono stati applicati.

La procedura per generare il valore CRC è la seguente:

1. Caricare un registro da 16 bit con \$FFFF. Chiamarlo registro CRC.
2. Effettuare il calcolo XOR sul primo byte (da 8 bit) del messaggio con il byte low del registro CRC da 16 bit, inserendo il risultato nel registro CRC.
3. Spostare il bit 1 del registro CRC a destra (verso LSB), e porre a zero MSB. Estrarre ed esaminare LSB.
4. (Se LSB era 0): Ripetere il punto 3 (un altro spostamento). (Se LSB era 1): Operazione XOR sul registro CRC con il valore polinomio \$A001 (1010 0000 0000 0001).
5. Ripetere i punti 3 e 4 fino a raggiungere 8 spostamenti. Dopo aver effettuato questi 8 spostamenti, verrà elaborato un byte completo da 8 bit.
6. Ripetere i punti dal 2 al 5 per il byte (da 8 bit) del messaggio successivo. Continuare questo procedimento fino a quando tutti i byte saranno elaborati.
7. Il contenuto finale del registro CRC corrisponderà al valore CRC.
8. Quando il valore CRC viene posto nel messaggio, i relativi byte high e low devono essere scambiati come segue.

INSERIRE IL VALORE CRC NEL MESSAGGIO

Quando il valore CRC da 16 bit (due byte da 8 bit) viene trasmesso nel messaggio, viene inviato prima il carattere low seguito poi dal carattere high.

Per esempio, se il valore CRC è \$35F7 (0011 0101 1111 0111):

Addr	Func	Data Count	Data	Data	Data	CRC lo F7	CRC hi 35
------	------	---------------	------	------	------	------	--------------	--------------

TABELLA FUNZIONI PER IL CALCOLO DEL CRC

Tutti i possibili valori CRC sono precaricati in due schieramenti, che sono semplicemente indicizzati mentre la funzione incrementa attraverso il buffer del messaggio. Uno schieramento contiene tutti i 256 valori CRC possibili per il byte high del campo CRC a 16 bit, e l'altro schieramento contiene tutti i valori per il byte low. Indicizzando in questo modo il CRC è possibile avere un'esecuzione più veloce rispetto a quella ottenuta dal calcolo di un nuovo valore CRC con ogni carattere nuovo dal buffer del messaggio.

```
/*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;
}

```

OPPURE PUO' ESSERE UTILIZZATO UN METODO DI CALCOLO SENZA TABELLA

```

/*      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
*/

/*
    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 (--Lenght);
    return cur_crc;
*/

```

2. Struttura comandi di lettura

Il dispositivo di comunicazione master può inviare comandi al modulo per leggerne lo stato e le impostazioni oppure per leggere i valori misurati, lo stato e le impostazioni del contatore abbinato. Possono essere letti più registri contemporaneamente solo se consecutivi (vedere capitolo 5). A seconda della modalità di protocollo MODBUS utilizzata, il comando di lettura sarà strutturato come segue.

2.1 MODBUS ASCII/RTU

I valori contenuti nei messaggi d'interrogazione e di risposta sono in formato hex.

Esempio messaggio d'interrogazione in caso di MODBUS ASCII/RTU: 010300020002D00B

Esempio	Byte	Descrizione	N. di byte
01	-	Slave address	1
03	-	Codice di funzione	1
00	High	Registro di partenza	2
02	Low		
00	High	N. di word da leggere	2
02	Low		
D0	High	Error check (CRC)	2
0B	Low		

Esempio messaggio di risposta in caso di MODBUS ASCII/RTU: 01030400035571F4F0

Esempio	Byte	Descrizione	N. di byte
01	-	Slave address	1
03	-	Codice di funzione	1
04	-	Conteggio byte	1
00	High	Dati richiesti	4
03	Low		
55	High		
71	Low		
F4	High	Error check (CRC)	2
F0	Low		

2.2 MODBUS TCP

I valori contenuti nei messaggi d'interrogazione e di risposta sono in formato hex.

Esempio messaggio d'interrogazione in caso di MODBUS TCP: 0100000000006010400020002

Esempio	Byte	Descrizione	N. di byte
01	-	Transaction identifier	1
00	High	Protocol identifier	4
00	Low		
00	High		
00	Low		
06	-	Conteggio byte	1
01	-	Unit identifier	1
04	-	Codice di funzione	1
00	High	Registro di partenza	2
02	Low		
00	High	N. di word da leggere	2
02	Low		

Esempio messaggio di risposta in caso di MODBUS TCP: 010000000000701040400035571

Esempio	Byte	Descrizione	N. di byte
01	-	Transaction identifier	1
00	High	Protocol identifier	4
00	Low		
00	High		
00	Low		
07	-	Conteggio byte	1
01	-	Unit identifier	1
04	-	Codice di funzione	1
04	-	N. di byte dei dati richiesti	2
00	High	Dati richiesti	4
03	Low		
55	High		
71	Low		

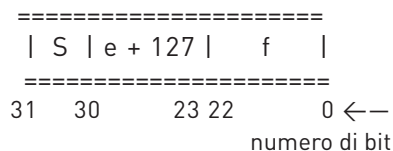
2.3 Virgola mobile secondo lo standard IEEE (non disponibile per MODBUS TCP)

Il formato di base consente la rappresentazione di un numero in virgola mobile secondo lo standard IEEE nel formato singola precisione (32 bit), come di seguito indicato:

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

dove **S** è il bit del segno, **e'** è la prima parte dell'esponente ed **f** è la frazione decimale da accostare ad 1. Internamente l'esponente ha una lunghezza di 8 bit e la frazione memorizzata è lunga 23 bit. Il valore di virgola mobile calcolato viene arrotondato.

Rappresentazione del formato della virgola mobile:



dove:

	lunghezza in bit
Segno	1
Esponente	8
Frazione	23 + {1}
Totale	m = 32 + {1}
Esponente	
Min e'	0
Max e'	255
Bias	127



NOTA

Le frazioni (le parti decimali) sono sempre riportate mentre l'unità (bit nascosto) non è memorizzato.

ESEMPIO DI CONVERSIONE DI UN VALORE RAPPRESENTATO IN VIRGOLA MOBILE

Valore letto in virgola mobile:

45AACC00₍₁₆₎

Valore convertito in formato binario:

```

010001011 010101011001100000000000 (2)
|-----|-----|
| esponente          frazione
segno
  
```

segno = 0

esponente = 10001011₍₂₎ = 139₍₁₀₎

frazione = 010101011001100000000000₍₂₎ / 8388608₍₁₀₎ =
= 2804736₍₁₀₎ / 8388608₍₁₀₎ = 0.334350585₍₁₀₎

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

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

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

$$= 5465.5$$

3. Struttura comandi di scrittura

Il dispositivo di comunicazione master può inviare comandi al modulo per programmarlo oppure per effettuare impostazioni sul contatore abbinato. Possono essere effettuate più impostazioni contemporaneamente con un solo comando solo se i registri relativi sono consecutivi (vedere capitolo 5). A seconda della modalità di protocollo MODBUS utilizzata, il comando di scrittura sarà strutturato come segue.

3.1 MODBUS ASCII/RTU

I valori contenuti nei messaggi d'interrogazione e di risposta sono in formato hex.

Esempio messaggio d'interrogazione in caso di MODBUS ASCII/RTU: 011005150001020008F053

Esempio	Byte	Descrizione	N. di byte
01	-	Slave address	1
10	-	Codice di funzione	1
05	High	Registro di partenza	2
15	Low		
00	High	N. di word da programmare	2
01	Low		
02	-	Conteggio byte dati	1
00	High	Dati per la programmazione	2
08	Low		
F0	High	Error check (CRC)	2
53	Low		

Esempio messaggio di risposta in caso di MODBUS ASCII/RTU: 01100515000110C1

Esempio	Byte	Descrizione	N. di byte
01	-	Slave address	1
10	-	Codice di funzione	1
05	High	Registro di partenza	2
15	Low		
00	High	N. di word programmati	2
01	Low		
10	High	Error check (CRC)	2
C1	Low		

3.2 MODBUS TCP

I valori contenuti nei messaggi d'interrogazione e di risposta sono in formato hex.

Esempio messaggio d'interrogazione in caso di MODBUS TCP: 0100000000009011005150001020008

Esempio	Byte	Descrizione	N. di byte
01	-	Transaction identifier	1
00	High	Protocol identifier	4
00	Low		
00	High		
00	Low		
09	-	Conteggio byte	1
01	-	Unit identifier	1
10	-	Codice di funzione	1
05	High	Registro di partenza	2
15	Low		
00	High	N. di word da leggere	2
01	Low		
02	-	Conteggio byte dati	1
00	High	Dati per la programmazione	2
08	Low		

Esempio messaggio di risposta in caso di MODBUS TCP: 0100000000006011005150001

Esempio	Byte	Descrizione	N. di byte
01	-	Transaction identifier	1
00	High	Protocol identifier	4
00	Low		
00	High		
00	Low		
06	-	Conteggio byte	1
01	-	Unit identifier	1
10	-	Codice di funzione	1
05	High	Registro di partenza	2
15	Low		
00	High	Command successfully sent	2
01	Low		

4. Codici di errore

Quando il modulo riceve un'interrogazione non valida, viene inviato un messaggio di errore (codice di errore). A seconda della modalità di protocollo MODBUS utilizzata, l'eventuale messaggio di errore sarà strutturato come segue.

4.1 MODBUS ASCII/RTU

I valori contenuti nei messaggi di risposta sono in formato hex.

Esempio messaggio di risposta in caso di MODBUS ASCII/RTU: 01830131F0

Esempio	Byte	Descrizione	N. di byte
01	-	Slave address	1
83	-	Codice di funzione (80+03)	1
01	-	Codice di errore	1
31	High	Error check (CRC)	2
F0	Low		

I codici di errore per MODBUS ASCII/RTU sono qui di seguito descritti:

- \$01 ILLEGAL FUNCTION:** il codice di funzione ricevuto nel messaggio d'interrogazione non è valido.
- \$02 ILLEGAL DATA ADDRESS:** l'indirizzo del registro ricevuto nel messaggio d'interrogazione non è valido (es. la combinazione di un registro e la lunghezza di trasferimento dati relativa non è valida).
- \$03 ILLEGAL DATA VALUE:** un valore contenuto nel campo dati del messaggio d'interrogazione ricevuto non è valido.
- \$04 ILLEGAL RESPONSE LENGTH:** la richiesta potrebbe generare una risposta con una dimensione maggiore di quella supportata dal protocollo MODBUS.

4.2 MODBUS TCP

I valori contenuti nei messaggi di risposta sono in formato hex.

Esempio messaggio di risposta in caso di MODBUS TCP: 010000000003018302

Esempio	Byte	Descrizione	N. di byte
01	-	Transaction identifier	1
00	High	Protocol identifier	4
00	Low		
00	High		
00	Low		
03	-	No. of byte of next data in this string	1
01	-	Unit identifier	1
83	-	Codice di funzione (80+03)	1
02	-	Codice di errore	1

I codici di errore per MODBUS TCP sono qui di seguito descritti:

- \$01 ILLEGAL FUNCTION:** il codice di funzione non è riconosciuto dal server.
- \$02 ILLEGAL DATA ADDRESS:** l'indirizzo del registro ricevuto nel messaggio d'interrogazione non è valido (es. la combinazione di un registro e la lunghezza di trasferimento dati relativa non è valida).

- \$03 ILLEGAL DATA VALUE:** un valore contenuto nel campo dati del messaggio d'interrogazione ricevuto non è valido.
- \$04 SERVER FAILURE:** errore di esecuzione del server.
- \$05 ACKNOWLEDGE:** il server ha ricevuto e accettato il messaggio d'interrogazione ma il servizio richiede un tempo piuttosto lungo per l'esecuzione. Il server quindi risponde solo con la presa visione del comando ricevuto.
- \$06 SERVER BUSY:** il server non è in grado di accettare la richiesta PDU MB. L'applicazione client ha la responsabilità di decidere se e quando rinviare la richiesta.
- \$0A GATEWAY PATH UNAVAILABLE:** il modulo di comunicazione non è configurato oppure non può comunicare.
- \$0B GATEWAY TARGET DEVICE FAILED TO RESPOND:** il contatore non è disponibile nella rete.

5. Tabelle dei registri



NOTA

Numero massimo di registri (o byte) leggibili con un unico comando:

- in modalità ASCII: 63 registri
- in modalità RTU: 127 registri
- in modalità TCP: 256 byte



NOTA

Numero massimo di registri programmabili con un unico comando:

- in modalità ASCII: 13 registri
- in modalità RTU: 29 registri
- in modalità TCP: 1 registro



NOTA

I valori dei registri sono in formato hex (\$).



NOTA

In protocollo MODBUS TCP non è possibile utilizzare i registri IEEE.

COLONNE TABELLA	SIGNIFICATO
Parametro	Parametro di misura da leggere
Descrizione registro	Descrizione del registro da leggere / programmare
Cod. di funzione (Hex)	Codice di funzione in formato hex. Identifica il tipo di comando (lettura / scrittura)
Segno	Se selezionata, il valore del registro di lettura può avere segno positivo o negativo. Per convertire un valore di registro con segno, seguire le istruzioni: Il bit più significativo (MSB=Most Significant Bit) indica il segno: 0=positivo (+), 1=negativo (-). <u>ESEMPIO DI VALORE NEGATIVO:</u> <div style="text-align: center;"> <div>MSB</div> <div>\$8020 = 10000000000100000 = -32</div> <div>HEX BIN DEC</div> </div>
INTERO	Dettagli per registri di tipo INTERO
IEEE	Dettagli per registri di tipo standard IEEE
Registro (Hex)	Indirizzo del registro in formato hex
Word	Numero di word da leggere / programmare per il registro (lunghezza)
U.M.	Unità di misura del parametro
Significato valori	Descrizione dei valori ricevuti da una risposta di un comando di lettura
Valori programmabili	Descrizione dei valori che possono essere inviati per un comando di scrittura

5.1 Registri di lettura (Codice di funzione \$01 / \$03 / \$04)

Parametro	Cod. di funzione (Hex)	Segno	INTERO			IEEE non disponibile per MODBUS TCP		
			Registro (Hex)	Word	U. M.	Registro (Hex)	Word	U. M.

VALORI IN TEMPO REALE

V1 • Tensione L-N fase 1	03 / 04		0000	2	mV	1000	2	V
V2 • Tensione L-N fase 2	03 / 04		0002	2	mV	1002	2	V
V3 • Tensione L-N fase 3	03 / 04		0004	2	mV	1004	2	V
V12 • Tensione di linea L-L 12	03 / 04		0006	2	mV	1006	2	V
V23 • Tensione di linea L-L 23	03 / 04		0008	2	mV	1008	2	V
V31 • Tensione di linea L-L 31	03 / 04		000A	2	mV	100A	2	V
V Σ • Tensione di sistema	03 / 04		000C	2	mV	100C	2	V
A1 • Corrente fase 1	03 / 04	X	000E	2	mA	100E	2	A
A2 • Corrente fase 2	03 / 04	X	0010	2	mA	1010	2	A
A3 • Corrente fase 3	03 / 04	X	0012	2	mA	1012	2	A
AN • Corrente di neutro	03 / 04	X	0014	2	mA	1014	2	A
A Σ • Corrente di sistema	03 / 04	X	0016	2	mA	1016	2	A
PF1 • Fattore di potenza fase 1	03 / 04	X	0018	1	-	1018	2	-
PF2 • Fattore di potenza fase 2	03 / 04	X	0019	1	-	101A	2	-
PF3 • Fattore di potenza fase 3	03 / 04	X	001A	1	-	101C	2	-
PF Σ • Fattore di potenza di sistema	03 / 04	X	001B	1	-	101E	2	-
P1 • Potenza attiva fase 1	03 / 04	X	001C	3	mW	1020	2	W
P2 • Potenza attiva fase 2	03 / 04	X	001F	3	mW	1022	2	W
P3 • Potenza attiva fase 3	03 / 04	X	0022	3	mW	1024	2	W
P Σ • Potenza attiva di sistema	03 / 04	X	0025	3	mW	1026	2	W
S1 • Potenza apparente fase 1	03 / 04	X	0028	3	mVA	1028	2	VA
S2 • Potenza apparente fase 2	03 / 04	X	002B	3	mVA	102A	2	VA
S3 • Potenza apparente fase 3	03 / 04	X	002E	3	mVA	102C	2	VA
S Σ • Potenza apparente di sistema	03 / 04	X	0031	3	mVA	102E	2	VA
Q1 • Potenza reattiva fase 1	03 / 04	X	0034	3	mvar	1030	2	var
Q2 • Potenza reattiva fase 2	03 / 04	X	0037	3	mvar	1032	2	var
Q3 • Potenza reattiva fase 3	03 / 04	X	003A	3	mvar	1034	2	var
Q Σ • Potenza reattiva di sistema	03 / 04	X	003D	3	mvar	1036	2	var
F • Frequenza	03 / 04		0040	1	mHz	1038	2	Hz
Ordine delle fasi (\$00=123-CCW, \$01=321-CW)	03 / 04		0041	1	-	103A	2	-

VALORI DEI CONTATORI TOTALI

+kWh1 • Energia attiva importata fase 1	03 / 04		0100	3	0.1 Wh	1100	2	Wh
+kWh2 • Energia attiva importata fase 2	03 / 04		0103	3	0.1 Wh	1102	2	Wh
+kWh3 • Energia attiva importata fase 3	03 / 04		0106	3	0.1 Wh	1104	2	Wh
+kWh Σ • Energia attiva importata di sistema	03 / 04		0109	3	0.1 Wh	1106	2	Wh
-kWh1 • Energia attiva esportata fase 1	03 / 04		010C	3	0.1 Wh	1108	2	Wh
-kWh2 • Energia attiva esportata fase 2	03 / 04		010F	3	0.1 Wh	110A	2	Wh
-kWh3 • Energia attiva esportata fase 3	03 / 04		0112	3	0.1 Wh	110C	2	Wh
-kWh Σ • Energia attiva esportata di sistema	03 / 04		0115	3	0.1 Wh	110E	2	Wh
+kVAh1-L • Energia apparente induttiva importata fase 1	03 / 04		0118	3	0.1 VAh	1110	2	VAh
+kVAh2-L • Energia apparente induttiva importata fase 2	03 / 04		011B	3	0.1 VAh	1112	2	VAh
+kVAh3-L • Energia apparente induttiva importata fase 3	03 / 04		011E	3	0.1 VAh	1114	2	VAh
+kVAh Σ -L • Energia apparente induttiva importata di sistema	03 / 04		0121	3	0.1 VAh	1116	2	VAh

Parametro	Cod. di funzione (Hex)	Segno	INTERO			IEEE non disponibile per MODBUS TCP		
			Registro (Hex)	Word	U. M.	Registro (Hex)	Word	U. M.

VALORI DEI CONTATORI TOTALI

-kVAh1-L • Energia apparente induttiva esportata fase 1	03 / 04		0124	3	0.1 VAh	1118	2	VAh
-kVAh2-L • Energia apparente induttiva esportata fase 2	03 / 04		0127	3	0.1 VAh	111A	2	VAh
-kVAh3-L • Energia apparente induttiva esportata fase 3	03 / 04		012A	3	0.1 VAh	111C	2	VAh
-kVAh Σ -L • Energia apparente induttiva esportata di sistema	03 / 04		012D	3	0.1 VAh	111E	2	VAh
+kVAh1-C • Energia apparente capacitiva importata fase 1	03 / 04		0130	3	0.1 VAh	1120	2	VAh
+kVAh2-C • Energia apparente capacitiva importata fase 2	03 / 04		0133	3	0.1 VAh	1122	2	VAh
+kVAh3-C • Energia apparente capacitiva importata fase 3	03 / 04		0136	3	0.1 VAh	1124	2	VAh
+kVAh Σ -C • Energia apparente capacitiva importata di sistema	03 / 04		0139	3	0.1 VAh	1126	2	VAh
-kVAh1-C • Energia apparente capacitiva esportata fase 1	03 / 04		013C	3	0.1 VAh	1128	2	VAh
-kVAh2-C • Energia apparente capacitiva esportata fase 2	03 / 04		013F	3	0.1 VAh	112A	2	VAh
-kVAh3-C • Energia apparente capacitiva esportata fase 3	03 / 04		0142	3	0.1 VAh	112C	2	VAh
-kVAh Σ -C • Energia apparente capacitiva esportata di sistema	03 / 04		0145	3	0.1 VAh	112E	2	VAh
+kvarh1-L • Energia reattiva induttiva importata fase 1	03 / 04		0148	3	0.1 varh	1130	2	varh
+kvarh2-L • Energia reattiva induttiva importata fase 2	03 / 04		014B	3	0.1 varh	1132	2	varh
+kvarh3-L • Energia reattiva induttiva importata fase 3	03 / 04		014E	3	0.1 varh	1134	2	varh
+kvarh Σ -L • Energia reattiva induttiva importata di sistema	03 / 04		0151	3	0.1 varh	1136	2	varh
-kvarh1-L • Energia reattiva induttiva esportata fase 1	03 / 04		0154	3	0.1 varh	1138	2	varh
-kvarh2-L • Energia reattiva induttiva esportata fase 2	03 / 04		0157	3	0.1 varh	113A	2	varh
-kvarh3-L • Energia reattiva induttiva esportata fase 3	03 / 04		015A	3	0.1 varh	113C	2	varh
-kvarh Σ -L • Energia reattiva induttiva esportata di sistema	03 / 04		015D	3	0.1 varh	113E	2	varh
+kvarh1-C • Energia reattiva capacitiva importata fase 1	03 / 04		0160	3	0.1 varh	1140	2	varh
+kvarh2-C • Energia reattiva capacitiva importata fase 2	03 / 04		0163	3	0.1 varh	1142	2	varh
+kvarh3-C • Energia reattiva capacitiva importata fase 3	03 / 04		0166	3	0.1 varh	1144	2	varh
+kvarh Σ -C • Energia reattiva capacitiva importata di sistema	03 / 04		0169	3	0.1 varh	1146	2	varh
-kvarh1-C • Energia reattiva capacitiva esportata fase 1	03 / 04		016C	3	0.1 varh	1148	2	varh
-kvarh2-C • Energia reattiva capacitiva esportata fase 2	03 / 04		016F	3	0.1 varh	114A	2	varh
-kvarh3-C • Energia reattiva capacitiva esportata fase 3	03 / 04		0172	3	0.1 varh	114C	2	varh
-kvarh Σ -C • Energia reattiva capacitiva esportata di sistema	03 / 04		0175	3	0.1 varh	114E	2	varh

VALORI DEI CONTATORI DI TARIFFA 1

+kWh1 • Energia attiva importata fase 1	03 / 04		0200	3	0.1 Wh	1200	2	Wh
+kWh2 • Energia attiva importata fase 2	03 / 04		0203	3	0.1 Wh	1202	2	Wh
+kWh3 • Energia attiva importata fase 3	03 / 04		0206	3	0.1 Wh	1204	2	Wh
+kWh Σ • Energia attiva importata di sistema	03 / 04		0209	3	0.1 Wh	1206	2	Wh
-kWh1 • Energia attiva esportata fase 1	03 / 04		020C	3	0.1 Wh	1208	2	Wh
-kWh2 • Energia attiva esportata fase 2	03 / 04		020F	3	0.1 Wh	120A	2	Wh
-kWh3 • Energia attiva esportata fase 3	03 / 04		0212	3	0.1 Wh	120C	2	Wh
-kWh Σ • Energia attiva esportata di sistema	03 / 04		0215	3	0.1 Wh	120E	2	Wh
+kVAh1-L • Energia apparente induttiva importata fase 1	03 / 04		0218	3	0.1 VAh	1210	2	VAh
+kVAh2-L • Energia apparente induttiva importata fase 2	03 / 04		021B	3	0.1 VAh	1212	2	VAh
+kVAh3-L • Energia apparente induttiva importata fase 3	03 / 04		021E	3	0.1 VAh	1214	2	VAh
+kVAh Σ -L • Energia apparente induttiva importata di sistema	03 / 04		0221	3	0.1 VAh	1216	2	VAh
-kVAh1-L • Energia apparente induttiva esportata fase 1	03 / 04		0224	3	0.1 VAh	1218	2	VAh
-kVAh2-L • Energia apparente induttiva esportata fase 2	03 / 04		0227	3	0.1 VAh	121A	2	VAh
-kVAh3-L • Energia apparente induttiva esportata fase 3	03 / 04		022A	3	0.1 VAh	121C	2	VAh
-kVAh Σ -L • Energia apparente induttiva esportata di sistema	03 / 04		022D	3	0.1 VAh	121E	2	VAh
+kVAh1-C • Energia apparente capacitiva importata fase 1	03 / 04		0230	3	0.1 VAh	1220	2	VAh

Parametro	Cod. di funzione (Hex)	Segno	INTERO			IEEE non disponibile per MODBUS TCP		
			Registro (Hex)	Word	U. M.	Registro (Hex)	Word	U. M.

VALORI DEI CONTATORI DI TARIFFA 1

+kVAh2-C • Energia apparente capacitiva importata fase 2	03 / 04		0233	3	0.1 VAh	1222	2	VAh
+kVAh3-C • Energia apparente capacitiva importata fase 3	03 / 04		0236	3	0.1 VAh	1224	2	VAh
+kVAh Σ -C • Energia apparente capacitiva importata di sistema	03 / 04		0239	3	0.1 VAh	1226	2	VAh
-kVAh1-C • Energia apparente capacitiva esportata fase 1	03 / 04		023C	3	0.1 VAh	1228	2	VAh
-kVAh2-C • Energia apparente capacitiva esportata fase 2	03 / 04		023F	3	0.1 VAh	122A	2	VAh
-kVAh3-C • Energia apparente capacitiva esportata fase 3	03 / 04		0242	3	0.1 VAh	122C	2	VAh
-kVAh Σ -C • Energia apparente capacitiva esportata di sistema	03 / 04		0245	3	0.1 VAh	122E	2	VAh
+kvarh1-L • Energia reattiva induttiva importata fase 1	03 / 04		0248	3	0.1 varh	1230	2	varh
+kvarh2-L • Energia reattiva induttiva importata fase 2	03 / 04		024B	3	0.1 varh	1232	2	varh
+kvarh3-L • Energia reattiva induttiva importata fase 3	03 / 04		024E	3	0.1 varh	1234	2	varh
+kvarh Σ -L • Energia reattiva induttiva importata di sistema	03 / 04		0251	3	0.1 varh	1236	2	varh
-kvarh1-L • Energia reattiva induttiva esportata fase 1	03 / 04		0254	3	0.1 varh	1238	2	varh
-kvarh2-L • Energia reattiva induttiva esportata fase 2	03 / 04		0257	3	0.1 varh	123A	2	varh
-kvarh3-L • Energia reattiva induttiva esportata fase 3	03 / 04		025A	3	0.1 varh	123C	2	varh
-kvarh Σ -L • Energia reattiva induttiva esportata di sistema	03 / 04		025D	3	0.1 varh	123E	2	varh
+kvarh1-C • Energia reattiva capacitiva importata fase 1	03 / 04		0260	3	0.1 varh	1240	2	varh
+kvarh2-C • Energia reattiva capacitiva importata fase 2	03 / 04		0263	3	0.1 varh	1242	2	varh
+kvarh3-C • Energia reattiva capacitiva importata fase 3	03 / 04		0266	3	0.1 varh	1244	2	varh
+kvarh Σ -C • Energia reattiva capacitiva importata di sistema	03 / 04		0269	3	0.1 varh	1246	2	varh
-kvarh1-C • Energia reattiva capacitiva esportata fase 1	03 / 04		026C	3	0.1 varh	1248	2	varh
-kvarh2-C • Energia reattiva capacitiva esportata fase 2	03 / 04		026F	3	0.1 varh	124A	2	varh
-kvarh3-C • Energia reattiva capacitiva esportata fase 3	03 / 04		0272	3	0.1 varh	124C	2	varh
-kvarh Σ -C • Energia reattiva capacitiva esportata di sistema	03 / 04		0275	3	0.1 varh	124E	2	varh

VALORI DEI CONTATORI DI TARIFFA 2

+kWh1 • Energia attiva importata fase 1	03 / 04		0300	3	0.1 Wh	1300	2	Wh
+kWh2 • Energia attiva importata fase 2	03 / 04		0303	3	0.1 Wh	1302	2	Wh
+kWh3 • Energia attiva importata fase 3	03 / 04		0306	3	0.1 Wh	1304	2	Wh
+kWh Σ • Energia attiva importata di sistema	03 / 04		0309	3	0.1 Wh	1306	2	Wh
-kWh1 • Energia attiva esportata fase 1	03 / 04		030C	3	0.1 Wh	1308	2	Wh
-kWh2 • Energia attiva esportata fase 2	03 / 04		030F	3	0.1 Wh	130A	2	Wh
-kWh3 • Energia attiva esportata fase 3	03 / 04		0312	3	0.1 Wh	130C	2	Wh
-kWh Σ • Energia attiva esportata di sistema	03 / 04		0315	3	0.1 Wh	130E	2	Wh
+kVAh1-L • Energia apparente induttiva importata fase 1	03 / 04		0318	3	0.1 VAh	1310	2	VAh
+kVAh2-L • Energia apparente induttiva importata fase 2	03 / 04		031B	3	0.1 VAh	1312	2	VAh
+kVAh3-L • Energia apparente induttiva importata fase 3	03 / 04		031E	3	0.1 VAh	1314	2	VAh
+kVAh Σ -L • Energia apparente induttiva importata di sistema	03 / 04		0321	3	0.1 VAh	1316	2	VAh
-kVAh1-L • Energia apparente induttiva esportata fase 1	03 / 04		0324	3	0.1 VAh	1318	2	VAh
-kVAh2-L • Energia apparente induttiva esportata fase 2	03 / 04		0327	3	0.1 VAh	131A	2	VAh
-kVAh3-L • Energia apparente induttiva esportata fase 3	03 / 04		032A	3	0.1 VAh	131C	2	VAh
-kVAh Σ -L • Energia apparente induttiva esportata di sistema	03 / 04		032D	3	0.1 VAh	131E	2	VAh
+kVAh1-C • Energia apparente capacitiva importata fase 1	03 / 04		0330	3	0.1 VAh	1320	2	VAh
+kVAh2-C • Energia apparente capacitiva importata fase 2	03 / 04		0333	3	0.1 VAh	1322	2	VAh
+kVAh3-C • Energia apparente capacitiva importata fase 3	03 / 04		0336	3	0.1 VAh	1324	2	VAh
+kVAh Σ -C • Energia apparente capacitiva importata di sistema	03 / 04		0339	3	0.1 VAh	1326	2	VAh
-kVAh1-C • Energia apparente capacitiva esportata fase 1	03 / 04		033C	3	0.1 VAh	1328	2	VAh
-kVAh2-C • Energia apparente capacitiva esportata fase 2	03 / 04		033F	3	0.1 VAh	132A	2	VAh

Parametro	Cod. di funzione (Hex)	Segno	INTERO			IEEE non disponibile per MODBUS TCP		
			Registro (Hex)	Word	U. M.	Registro (Hex)	Word	U. M.

VALORI DEI CONTATORI DI TARIFFA 2

-kVAh3-C • Energia apparente capacitiva esportata fase 3	03 / 04		0342	3	0.1 VAh	132C	2	VAh
-kVAh Σ -C • Energia apparente capacitiva esportata di sistema	03 / 04		0345	3	0.1 VAh	132E	2	VAh
+kvarh1-L • Energia reattiva induttiva importata fase 1	03 / 04		0348	3	0.1 varh	1330	2	varh
+kvarh2-L • Energia reattiva induttiva importata fase 2	03 / 04		034B	3	0.1 varh	1332	2	varh
+kvarh3-L • Energia reattiva induttiva importata fase 3	03 / 04		034E	3	0.1 varh	1334	2	varh
+kvarh Σ -L • Energia reattiva induttiva importata di sistema	03 / 04		0351	3	0.1 varh	1336	2	varh
-kvarh1-L • Energia reattiva induttiva esportata fase 1	03 / 04		0354	3	0.1 varh	1338	2	varh
-kvarh2-L • Energia reattiva induttiva esportata fase 2	03 / 04		0357	3	0.1 varh	133A	2	varh
-kvarh3-L • Energia reattiva induttiva esportata fase 3	03 / 04		035A	3	0.1 varh	133C	2	varh
-kvarh Σ -L • Energia reattiva induttiva esportata di sistema	03 / 04		035D	3	0.1 varh	133E	2	varh
+kvarh1-C • Energia reattiva capacitiva importata fase 1	03 / 04		0360	3	0.1 varh	1340	2	varh
+kvarh2-C • Energia reattiva capacitiva importata fase 2	03 / 04		0363	3	0.1 varh	1342	2	varh
+kvarh3-C • Energia reattiva capacitiva importata fase 3	03 / 04		0366	3	0.1 varh	1344	2	varh
+kvarh Σ -C • Energia reattiva capacitiva importata di sistema	03 / 04		0369	3	0.1 varh	1346	2	varh
-kvarh1-C • Energia reattiva capacitiva esportata fase 1	03 / 04		036C	3	0.1 varh	1348	2	varh
-kvarh2-C • Energia reattiva capacitiva esportata fase 2	03 / 04		036F	3	0.1 varh	134A	2	varh
-kvarh3-C • Energia reattiva capacitiva esportata fase 3	03 / 04		0372	3	0.1 varh	134C	2	varh
-kvarh Σ -C • Energia reattiva capacitiva esportata di sistema	03 / 04		0375	3	0.1 varh	134E	2	varh

VALORI DEI CONTATORI PARZIALI

+kWh Σ • Energia attiva importata di sistema	03 / 04		0400	3	0.1 Wh	1400	2	Wh
-kWh Σ • Energia attiva esportata di sistema	03 / 04		0403	3	0.1 Wh	1402	2	Wh
+kVAh Σ -L • Energia apparente induttiva importata di sistema	03 / 04		0406	3	0.1 VAh	1404	2	VAh
-kVAh Σ -L • Energia apparente induttiva esportata di sistema	03 / 04		0409	3	0.1 VAh	1406	2	VAh
+kVAh Σ -C • Energia apparente capacitiva importata di sistema	03 / 04		040C	3	0.1 VAh	1408	2	VAh
-kVAh Σ -C • Energia apparente capacitiva esportata di sistema	03 / 04		040F	3	0.1 VAh	140A	2	VAh
+kvarh Σ -L • Energia reattiva induttiva importata di sistema	03 / 04		0412	3	0.1 varh	140C	2	varh
-kvarh Σ -L • Energia reattiva induttiva esportata di sistema	03 / 04		0415	3	0.1 varh	140E	2	varh
+kvarh Σ -C • Energia reattiva capacitiva importata di sistema	03 / 04		0418	3	0.1 varh	1410	2	varh
-kvarh Σ -C • Energia reattiva capacitiva esportata di sistema	03 / 04		041B	3	0.1 varh	1412	2	varh

VALORI DI BILANCIO

kWh Σ • Energia attiva di sistema	03 / 04	X	041E	3	0.1 Wh	1414	2	Wh
kVAh Σ -L • Energia apparente induttiva di sistema	03 / 04	X	0421	3	0.1 VAh	1416	2	VAh
kVAh Σ -C • Energia apparente capacitiva di sistema	03 / 04	X	0424	3	0.1 VAh	1418	2	VAh
kvarh Σ -L • Energia reattiva induttiva di sistema	03 / 04	X	0427	3	0.1 varh	141A	2	varh
kvarh Σ -C • Energia reattiva capacitiva di sistema	03 / 04	X	042A	3	0.1 varh	141C	2	varh

Descrizione registro	Cod. di funzione (Hex)	INTERO		Significato valori
		Registro (Hex)	Word	
DATI CONTATORE E MODULO DI COMUNICAZIONE				
Numero seriale	03 / 04	0500	5	10 car. ASCII (\$00÷\$FF)
Modello	03 / 04	0505	1	\$03=6A trifase/4fili \$06=6A trifase/3fili \$08=80A trifase/4fili \$0A=80A trifase/3fili \$0C=80A monofase/2fili
Tipo	03 / 04	0506	1	\$02=MID
Firmware release	03 / 04	0507	1	Convertire il valore di lettura formato Hex in valore Decimale. es. \$66=102 =rel. 1.02
Hardware version	03 / 04	0508	1	Convertire il valore di lettura formato Hex in valore Decimale. es. \$64=100 =rev. 1.00
Riservato	03 / 04	0509	2	
Tariff in use	03 / 04	050B	1	\$01=tariffa 1 \$02=tariffa 2
Primary/secondary value	03 / 04	050C	1	\$00=primario \$01=secondario
Codice di errore	03 / 04	050D	1	\$00=nessuno \$01=ordine delle fasi errato
Valore TA (solo in caso di contatore modello 6A trifase)	03 / 04	050E	1	\$0001÷\$2710
Riservato	03 / 04	050F	2	
Valore di fondoscala corrente (FSA)	03 / 04	0511	1	\$00=1A \$01=5A \$02=80A
Modalità d’inserzione	03	0512	1	\$01=3-fasi/4 fili/3 correnti \$02=3-fasi/3 fili/2 correnti \$03=monofase
Indirizzo MODBUS (non disponibile in caso di LAN GATEWAY)	03	0513	1	\$01÷\$F7
Modalità MODBUS (non disponibile in caso di LAN GATEWAY)	03	0514	1	\$00=7E2 (ASCII) \$01=8N1 (RTU)
Velocità di comunicazione (non disponibile in caso di LAN GATEWAY)	03	0515	1	\$01=300 bps \$02=600 bps \$03=1200 bps \$04=2400 bps \$05=4800 bps \$06=9600 bps \$07=19200 bps \$08=38400 bps \$09=57600 bps \$0A=115200 bps

Descrizione registro	Cod. di funzione (Hex)	INTERO		Significato valori
		Registro (Hex)	Word	
DATI CONTATORE E MODULO DI COMUNICAZIONE				
Stato dei contatori parziali	03	0517	1	<p>Convertire il valore di lettura formato Hex in valore Binario. es. \$0003= 0000000000000011</p> <p>Ogni bit corrisponde allo stato di un contatore parziale.</p> <p>0=disabilitato 1=attivo</p> <p>0000000000000011 ←</p> <p>Iniziare a leggere la stringa di bit seguendo la direzione della freccia. Il primo bit corrisponde allo stato del primo contatore nell’elenco:</p> <p>1) +kWhΣ PAR 2) -kWhΣ PAR 3) +kVAhΣ-L PAR 4) -kVAhΣ-L PAR 5) +kVAhΣ-C PAR 6) -kVAhΣ-C PAR 7) +kvarhΣ-L PAR 8) -kvarhΣ-L PAR 9) +kvarhΣ-C PAR 10) -kvarhΣ-C PAR</p> <p>Gli ultimi sei bit della stringa sono riservati.</p> <p>Nell’esempio indicato, solo i contatori +kWhΣ PAR e -kWhΣ PAR sono attivi.</p>

DATI MODULO DI COMUNICAZIONE

Numero seriale	03 / 04	0518	5	10 car. ASCII (\$00-\$FF)
Riservato	03 / 04	051D	2	
Versione firmware	03 / 04	051F	1	Convertire il valore di lettura formato Hex in valore Decimale. es. \$66=102 =rel. 1.02
Versione hardware	03 / 04	0520	1	Convertire il valore di lettura formato Hex in valore Decimale. es. \$64=100 =rev. 1.00

Descrizione registro	Cod. di funzione (Hex)	Registro (Hex)	Significato valori
PINZE			
Eventi di allarme	01	0000	<p>40 pinze</p> <p>Byte 1 - tensione fuori dal limite UV3 UV2 UV1 UVΣ OV3 OV2 OV1 OVΣ </p> <p>Byte 2 - tensione di linea fuori dal limite COM RES UV23 UV31 UV12 OV23 OV31 OV12 </p> <p>Byte 3/4 - corrente fuori dal limite RES RES RES RES RES RES UI1 UI3 UI2 UI1 UIΣ OI1 OI3 OI2 OI1 OIΣ </p> <p>Byte 5 - frequenza fuori dal limite RES RES RES RES RES RES RES F </p> <p>LEGENDA UV=tensione sotto al limite OV=tensione sopra al limite UI=corrente sotto al limite OI=corrente sopra al limite F=frequenza fuori dal limite COM=comunicazione in corso RES=bit riservato a 0</p> <p>NOTA: i valori di soglia di tensione, corrente e frequenza possono cambiare a seconda del modello di contatore abbinato. Fare riferimento alla tabella riportata qui sotto.</p>

TENSIONE NOMINALE DEL CONTATORE	SOGLIE DEI PARAMETRI			
	TENSIONE DI FASE disponibile solo per contatori mod. 2-4 fili	TENSIONE DI LINEA non disponibile per contatore mod. 2 fili	CORRENTE	FREQUENZA
A	UV _{L-N} : Vnom -20% OV _{L-N} : Vnom +20%	UV _{L-L} : Vnom * $\sqrt{3}$ -20% OV _{L-L} : Vnom * $\sqrt{3}$ +20%	UI: Valore di corrente di start (Ist) OI: Valore di fondo scala (FS)	F bassa: 45Hz F alta: 65Hz
B				
C	UV _{L-N} : 230V -20% OV _{L-N} : 240V +20%	UV _{L-L} : 400V -20% OV _{L-L} : 415V +20%		
D				

5.2 Registri di scrittura (Codice di funzione \$10)

Descrizione registro	Cod. di funzione (Hex)	INTERO		Valori programmabili
		Registro (Hex)	Word	
DATI CONTATORE E MODULO DI COMUNICAZIONE				
Indirizzo MODBUS (non disponibile in caso di LAN GATEWAY)	10	0513	1	\$01÷\$F7
Modalità MODBUS (non disponibile in caso di LAN GATEWAY)	10	0514	1	\$00=7E2 (ASCII) \$01=8N1 (RTU)
Velocità di comunicazione (non disponibile in caso di LAN GATEWAY)	10	0515	1	\$01=300 bps \$02=600 bps \$03=1200 bps \$04=2400 bps \$05=4800 bps \$06=9600 bps \$07=19200 bps \$08=38400 bps \$09=57600 bps \$0A=115200 bps
Riservato	10	0516	1	
Stato dei contatori parziali	10	0517	1	Byte 1-selezione del contatore parziale: \$00=+kWhΣ PAR \$01=-kWhΣ PAR \$02=+kVAhΣ -L PAR \$03=-kVAhΣ -L PAR \$04=+kVAhΣ -C PAR \$05=-kVAhΣ -C PAR \$06=+kvarhΣ -L PAR \$07=-kvarhΣ -L PAR \$08=+kvarhΣ -C PAR \$09=-kvarhΣ -C PAR \$0A=tutti i contatori parziali Byte 2-azione su contatore/i parziale/i: \$01=avviare \$02=fermare \$03=azzerare es. avvia il contatore +kWhΣ PAR 00=+kWhΣ PAR 01=start valore finale da impostare: 0001



NOTA

I registri di scrittura \$0513, \$0514, \$0515 consentono di programmare i parametri di comunicazione del modulo.



FRER SRL

Viale Europa, 12 • 20093 Cologno Monzese (MI) • ITALY • Tel. : +39 02 27302828 • Fax : +39 02 25391518 • www.frer.it