Guida Tecnica:
Protocollo di Comunicazione Regolatori Digitali
Technical Guide:
Digital Regulators Communication Protocol

mecc alte

With Great Knowledge Comes Great Power

INTRODUZIONE

Il sistema implementa un sottoinsieme del protocollo Modbus ed in particolare implementa due comandi: 3 (read holding register) e 6 (preset single register). A questi due ne è aggiunto uno (non standard, che impiega un codice inutilizzato anche se non "libero"): 32, chiamato "get status", che restituisce N word (attualmente 4) che possono essere utilizzate per lo stato della macchina. Il motivo della presenza di questo comando è per avere maggiore flessibilità nei dati da scambiare ciclicamente e per non essere legati ad indirizzi specifici che possono non essere contigui.

Esiste anche un'altra eccezione allo standard che è dovuta ai limiti imposti dall'hardware dei regolatori digitali: per leggere il contenuto dei registri della zona allarmi è necessaria un'operazione in EEPROM (non essendo questi registri ricopiati in RAM), quindi il comando di lettura deve essere limitato ad una sola word per volta.

INTRODUCTION

The system implements a subgroup of the Modbus protocol and in particular two commands: 3 (read holding register) and 6 (preset single register). Another one (32, not standard, but using a "free" code) which returns N word (currently 4) has been added to these two, and can be used for the machine state. This command gives greater flexibility in cyclic exchange data, and lets to be not bound to specific addresses which may not be contiguous.

There is also another exception to the standard which is due to the limits set by the digital regulator hardware: an EEPROM operation is necessary for reading the contents of the alarm area registers (because these registers are not copied in RAM), therefore the read command should be limited to one single word at a time.

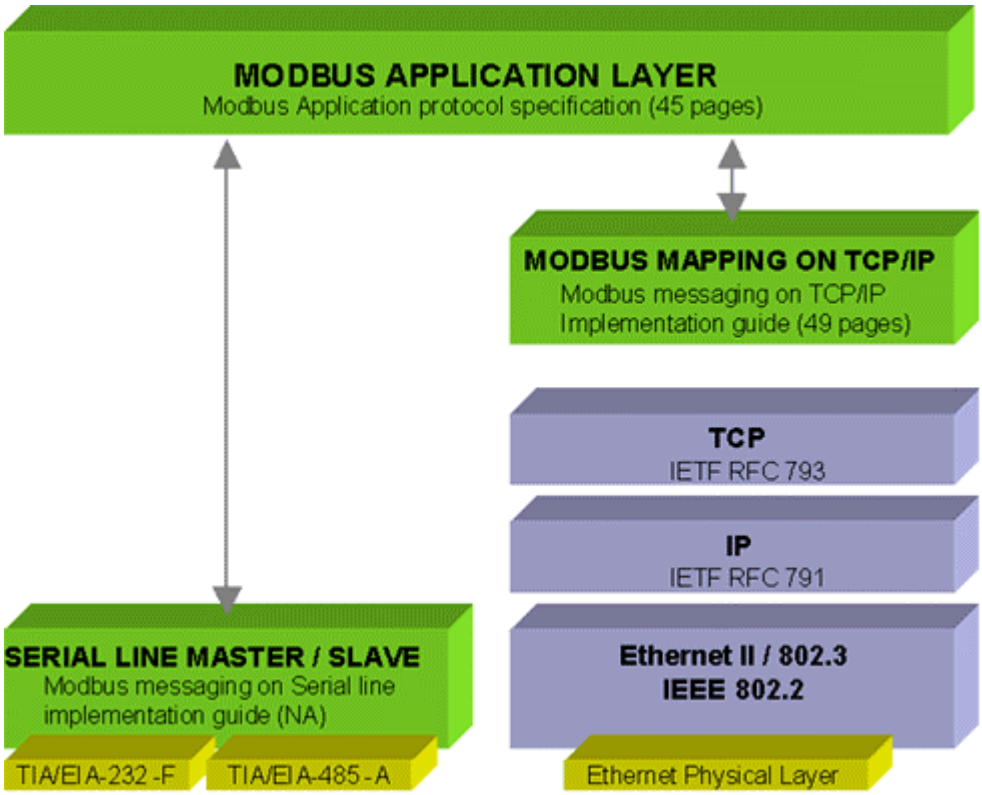

Fig. 1 - Struttura del protocollo MODBUS / Structure of the MODBUS protocol

Il protocollo Modbus prevede due possibili modalità per ricavare il sincronismo (e quindi riconoscere l'inizio e la fine dei pacchetti):

- Modalità ASCII e utilizzo di caratteri di controllo (modo ASCII)
- Modalità binaria e utilizzo di pause, in cui la linea rimane inutilizzata per almeno il tempo di 3,5 caratteri (modo RTU)

Il protocollo utilizzato nei regolatori digitali Mecc Alte è uno "pseudo" RTU.

I regolatori implementano al loro interno una tabella di registri, tramite i quali è possibile accedere alle funzioni di controllo, comando o misura della scheda. I registri sono tutti a 16bit e assumono funzioni e significati diversi solo quando sono interpretati dal programma del regola

The Modbus protocol uses two possible modes for finding the synchronism (therefore recognising the beginning and end of the packets):

- ASCII mode and use of control characters (ASCII mode)
- binary mode and use of pauses, in which the line remains unused for at least 3.5 characters (RTU mode)

The Mecc Alte digital regulator uses RTU protocol.

The regulators have an internal table of registers, which can be used to access all the board control, drive or measure functions. The registers are all 16 bit and take on different functions and meanings only when interpreted by the digital regulator program. This means they do not have particular meanings at a communication

tore digitale, non hanno cioè significati particolari a livello di protocollo di comunicazione, il quale è utilizzato solo per il trasporto di word a 16 bit.

Il protocollo è di tipo master/slave e prevede un master ed un certo numero di slave. Anche se il campo indirizzo è un byte, è opportuno limitare comunque il numero di slave ad un numero massimo di 32, o anche minore. Il numero massimo può dipendere da molti fattori: tipo di driver di linea, baudrate, lunghezza dei collegamenti.

Il regolatore implementa uno slave il cui indirizzo è programmabile. Per realizzare una rete, e quindi necessario programmare ogni slave con un indirizzo diverso.
Le impostazioni della linea seriale sono fisse (non sono cioè programmabili) i valori sono 9600baud, 8bit, parità esclusa, 1 stop bit.

Errori

Vi sono varie cause di errore, o più in generale, di condizione di non perfetta comunicazione.

– La prima possibilità è che lo slave non sia connesso o sia guasto; in questo caso il master ritrasmette la richiesta dopo un timeout che normalmente è dell'ordine delle centinaia di millisecondi. Se gli slave hanno un tempo di latenza basso, il timeout può essere ridotto di conseguenza.

– Lo slave riceve un messaggio con errori; in questo caso il CRC fallisce e lo slave non risponde. Il master si comporta come nel caso precedente.

– Lo slave riceve il messaggio correttamente, ma la richiesta del master non può essere servita. In questo caso lo slave restituisce una risposta in cui al campo funzione viene sommato 128 (praticamente si imposta il bit più significativo del byte) ed il campo dati non contiene dati ma un byte che specifica il tipo di errore.

I codici di errore sono i seguenti:

1 Illegal FUNCTION
2 Illegal data address
3 Illegal data value
4 Slave device failure
5 Acknowledge
6 Slave busy
7 Negative Acknowledge
8 Memory parity error

Nei regolatori digitali Mecc Alte, al momento sono utilizzati solo i codici ModBus 2, 3 e 6

NOTA IMPORTANTE

Allo scopo di ridurre il carico computazionale al regolatore, dovuto alla comunicazione, il CSUM dei parametri calcolati in EEPROM è calcolato dal master.
Più in particolare, quando il master deve effettuare una scrittura nella zona parametri conservata in EEPROM, subito dopo, deve ricalcolare il CSUM e scriverlo nell'apposita locazione. Similmente, quando si fa l'upload dell'intera lista di parametri, il CSUM deve essere calcolato e scritto alla fine (una sola volta). Il regolatore stesso ricalcola e salva il CSUM solo nel caso, dopo il RESET, questo non corrisponda; in tal caso viene generato l'allarme "CheckSum Error".

protocol level, which is only used for transporting words at 16 bits.

The protocol is of the master/slave type, made up of one master and a certain number of slaves. Even if the address field is one byte, it is a good idea to limit the number of slaves to 32 or less. The maximum number can depend on many factors: the type of line driver, the baud rate, the length of the connections.

The regulator implements a slave with programmable address. To create a network, each slave should be programmed with a different address.
The serial line settings are fixed (they can therefore not be programmed) – the values are 9600baud, 8bit, parity excluded, 1 stop bit.

Errors

There are various causes of error or, more in general, of imperfect communication conditions.

- The first possibility is that the slave is not connected or is faulty. In this case the master retransmits the request after a timeout, which is normally set in hundredths of a millisecond. If the slaves have a low latency time, the timeout can be reduced as a result.

- The slave receives a message with errors. In this case the CRC fails and the slave does not respond. The master behaves as described above.

- The slave receives the message correctly, but the request from the master cannot be served. In this case the slave returns a reply in which 128 (practically the most important bit of the byte is set) and the data field which does not hold data but a byte that specifies the type of error are summed at the function field.

The error codes are as follows:

1 Illegal FUNCTION
2 Illegal data address
3 Illegal data value
4 Slave device failure
5 Acknowledge
6 Slave busy
7 Negative Acknowledge
8 Memory parity error

Only codes 2, 3 and 6 are used in the Mecc Alte digital regulators.

IMPORTANT NOTE

In order to reduce the computational load, caused by to communication, to the regulator, the CSUM of the parameters calculated in EEPROM is calculated by the master.
More in detail, when the master carries out a writing in the parameter area saved in EEPROM, it immediately recalculates the CSUM and writes it in the correct position. Similarly, when the whole parameter list is uploaded, the CSUM is written only once.
The regulator recalculates and saves the CSUM only if it does not correspond after resetting;in this case the alarm "CheckSum Error" is generated.

CONTENUTO DEI PACCHETTI MASTER/SLAVE

Comando 03: lettura di N words

CONTENTS OF THE MASTER/SLAVE PACKETS

Command 03: the reading of N words

MASTER		
Field Name	**(Hex)**	**Note**
Slave Address	11	Indirizzo del destinatario del comando / Address of the command receiver
Function	03	Codice del comando: 03 = lettura / Command code: 03 = reading
Starting Address Hi	00	Indirizzo di partenza delle word da leggere: h006B = d107
Starting Address Lo	6B	Starting address of the word to be read: h006B = d107
No. of words	00	Numero di word da leggere: h0003 = d3
No. of words	03	Number of word to be read: h0003 = d3
CRC H	xx	Codice di controllo per la correttezza dei dati, v. "Calcolo del CRC"
CRC L	xx	Control code for data correctness, refer to "Calculating the CRC"

L'unità master trasmette quindi il pacchetto: 11 03 00 6B 00 03 xx xx

Se ha correttamente ricevuto il comando, l'unità slave risponde ritrasmettendo "Slave Address" e "Function" come conferma del comando ricevuto, di seguito il numero di bytes e i dati richiesti.

The master unit therefore transmits the packet: 11 03 00 6B 00 03 xx xx

If it has correctly received the command, the slave unit replies by retransmitting "Slave Address" and "Function" to confirm the received command. The number of bytes and the requested data is given below.

SLAVE		
Field Name	**(Hex)**	**Note**
Slave Address	11	Ritrasmette "Slave Address" e "Function" = risposta corretta al comando
Function	03	Retransmits "Slave Address" and "Function" = correct reply to command
Byte Count	06	Numero di bytes che verranno trasmessi: h06 = d6 Number of bytes that will be transmitted: h06 = d6
Data Hi (Register 107)	02	1ª word all'indirizzo h006B (d107): h020B
Data Lo (Register 107)	2B	1st word at the h006B (d107) address: h020B
Data Hi (Register 108)	00	2ª word all'indirizzo h006C (d108): h0000
Data Lo (Register 108)	00	2nd word at the h006C (d108) address: h0000
Data Hi (Register 109)	00	3ª word all'indirizzo h006D (d109): h00064
Data Lo (Register 109)	64	3rd word at the h006D (d109) address: h00064
CRC H	xx	3ª word all'indirizzo h006D (d109): h00064
CRC L	xx	Codice di controllo per la correttezza dei dati, v. "Calcolo del CRC" Control code for data correctness, refer to "Calculating the CRC"

L'unità slave trasmette quindi il pacchetto: 11 03 06 02 2B 00 00 00 64 xx xx

Se il comando non è stato correttamente ricevuto, l'unità slave risponde

The slave unit therefore transmits the following packet: 11 03 06 02 2B 00 00 00 64 xx xx

If the command is not received correctly, the slave replies

SLAVE		
Field Name	**(Hex)**	**Note**
Slave Address	11	Ritrasmette "Slave Address" come risposta al comando Retransmits "Slave Address" in reply to the command
Function with error	83	Viene sommato h80 = d128 al comando ricevuto (h03 + h80 = h83) h80 = d128 is summed to the received command (h03 + h80 = h83)
Error code	nn	Codice errore (v. pag. 3) / Error code (see page 3)
CRC H	xx	Codice di controllo per la correttezza dei dati, v. "Calcolo del CRC"
CRC L	xx	Control code for data correctness, refer to "Calculating the CRC"

L'unità slave trasmette quindi il pacchetto: 11 83 nn xx xx

The slave unit therefore transmits the following packet: 11 83 nn xx xx

Comando 06: Scrittura di una word

Command 06: Writing a word

MASTER		
Field Name	**(Hex)**	**Note**
Slave Address	11	Indirizzo del destinatario del comando / Address of the command receiver
Function	06	Codice del comando: 06 = scrittura / Command code: 06 = writing
Register Address Hi	00	Indirizzo della word da scrivere: h0001 = d1
Register Address Lo	01	Address of the word to be written: h0001 = d1
Preset Data Hi	00	Dato da scrivere: h0003 = d3
Preset Data Lo	03	Datum to be written: h0003 = d3
CRC H	xx	Codice di controllo per la correttezza dei dati, v. "Calcolo del CRC"
CRC L	xx	Control code for data correctness, refer to "Calculating the CRC"

L'unità master trasmette quindi il pacchetto: 11 06 00 01 00 03 xx xx

The master unit therefore transmits the following packet: 11 06 00 01 00 03 xx xx

Se ha correttamente ricevuto il comando, l'unità slave risponde ritrasmettendo l'intero pacchetto

If it receives the command correctly, the slave unit replies by retransmitting the whole packet

SLAVE		
Field Name	**(Hex)**	**Note**
Slave Address	11	Indirizzo del destinatario del comando / Address of the command receiver
Function	06	Codice del comando: 06 = scrittura / Command code: 06 = writing
Register Address Hi	00	Indirizzo della word da scrivere: h0001 = d1
Register Address Lo	01	Address of the word to be written: h0001 = d1
Preset Data Hi	00	Dato da scrivere: h0003 = d3
Preset Data Lo	03	Datum to be written: h0003 = d3
CRC H	xx	Codice di controllo per la correttezza dei dati, v. "Calcolo del CRC"
CRC L	xx	Control code for data correctness, refer to "Calculating the CRC"

L'unità slave trasmette quindi il pacchetto: 11 06 00 01 00 03 xx xx

The slave unit therefore transmits the following packet: 11 06 00 01 00 03 xx xx

Comando 32: get status

Command 32: get status

MASTER		
Field Name	**(Hex)**	**Note**
Slave Address	11	Indirizzo del destinatario del comando / Address of the command receiver
Function	20	Codice del comando: 20 = scrittura (h20 = d32) Command code: 20 = writing (h20 = d32)
Starting Address Hi	00	Senza significato
Starting Address Lo	00	Without meaning
No. of words	00	Numero fisso (4) di word di stato da leggere: h0004 = d4
No. of words	04	Fixed number (4) of state words to be read: h0004 = d4
CRC H	xx	Codice di controllo per la correttezza dei dati, v. "Calcolo del CRC"
CRC L	xx	Control code for data correctness, refer to "Calculating the CRC"

L'unità master trasmette quindi il pacchetto: 11 20 00 00 00 04 xx xx

The master unit therefore transmits the following packet: 11 20 00 00 00 04 xx xx

Se ha correttamente ricevuto il comando, l'unità slave risponde ritrasmettendo "Slave Address" e "Function" come conferma del comando ricevuto, di seguito il numero di bytes e le 4 word di stato

If it receives the command correctly, the slave unit replies by transmitting "Slave Address" and "Function" to confirm command reception, followed by the number of bytes and the 4 state words.

SLAVE		
Field Name	**(Hex)**	**Note**
Slave Address	11	Ritrasmette "Slave Address" e "Function" = risposta corretta al comando Restransmits "Slave Address" and "Function" = correct reply to command
Function	20	
Byte Count	08	Numero di bytes che verranno trasmessi: h08 = d8, 8 bytes = 4 words Number of bytes that will be transmitted: h08 = d8, 8 bytes = 4 words
Data Hi (W0)	02	1ª word: h020B
Data Lo (W0)	2B	
Data Hi (W1)	00	2ª word: h0000
Data Lo (W1)	00	
Data Hi (W2)	00	3ª word: h00064
Data Lo (W2)	64	
Data Hi (W3)	00	3ª word: h00064
Data Lo (W3)	64	
CRC H	xx	Codice di controllo per la correttezza dei dati, v. "Calcolo del CRC" Control code for data correctness, refer to "Calculating the CRC"
CRC L	xx	

L'unità slave trasmette quindi il pacchetto: 11 20 08 02 2B 00 00 00 64 00 64 xx xx

Calcolo del CRC

Dettagli relativi al tipo di CRC utilizzato possono essere reperiti nel documento che descrive lo standard Modbus (scaricabile dal sito). Di seguito è riportato il codice C del calcolo.

```
/* ------------------------------ */
#define GENERATORE      (0xA001)

/*  Il  crc  ha  i  byte  invertiti;  dovrebbe  essere
giusto così */
WORD get_crc16 (unsigned int * msg, WORD size )
{     WORD crc = 0xFFFF, nshift, fai;
      for ( ; size--; )
      {     crc ^= (WORD) *msg++;
            for (nshift = 8; nshift--;)
            {     fai = crc & 0x0001;
                  crc >>= 1;
                  if (fai)
                        crc ^= GENERATORE;
            }
      }
      fai = ( crc << 8) & 0xFF00;
      fai = fai + (( crc >> 8) & 0x00FF) ;
      return (fai);
}

/* ------------------------------ */
```

La routine è utilizzata sia dal master sia dallo slave.

The slave unit therefore transmits the following packet: 11 20 08 02 2B 00 00 00 64 00 64 xx xx

Calculating the CRC

Details regarding the type of CRC used can be found in the document that describes the standard Modbus (this document can be downloaded from the site). The calculation C code is given below.

```
/* ------------------------------ */
#define GENERATORE      (0xA001)

/* The crc has swapped bytes; this should be correct
*/
WORD get_crc16 (unsigned int * msg, WORD size )
{     WORD crc = 0xFFFF, nshift, fai;
      for ( ; size--; )
      {     crc ^= (WORD) *msg++;
            for (nshift = 8; nshift--;)
            {     fai = crc & 0x0001;
                  crc >>= 1;
                  if (fai)
                        crc ^= GENERATORE;
            }
      }
      fai = ( crc << 8) & 0xFF00;
      fai = fai + (( crc >> 8) & 0x00FF) ;
      return (fai);
}

/* ------------------------------ */
```

The routine is used by both the master and the slave.

RIFERIMENTI / REFERENCES	
Internet address	**Description**
http://www.modbus.org	Modbus-IDA is a group of independent users and suppliers of automation devices that seeks to drive the adoption of the Modbus communication protocol suite and the evolution to address architectures for distributed automation systems across multiple market segments. Modbus-IDA will also provide the infrastructure to obtain and share information about the protocols, their application and certification to simplify implementation by users resulting in reduced costs.
http://www.modicon.com	This is the site of the original inventors of the MODBUS protocol, Modicon, now known as Schneider Automation. You'll find useful MODBUS product-related information.
http://www.automationcorner.com	This site contains software solutions industry links, and a library of industrial information.
http://ethernet.industrial-networking.com	Very useful site - a mine of information on using Ethernet in industrial applications.
http://www.codeguru.com/network/mod_rssim.html	Modbus serial RTU simulator. Compiles with Visual C++ 6.0, and runs on Windows 2000 and probably 95/98.
http://www.industrialethernet.com	The web-site of the newly formed Association of Industrial Vendors and Users that use Ethernet in an industrial environment. The site is still under development, but is worth keeping an eye on.
http://www.geocities.com/pbmcrae42/	Example C code for linux modbus rtu comms
http://pes.free.fr/	LibModbus is a simple modbus toolbox for Linux. This library contents master, slave and serial port configuration routines.
http://jamod.sourceforge.net/	Java Modbus Library. This library was implemented by Dieter Wimberger

INDICE DELLA REVISIONE / REVISION HISTORY		
Rev.	**Data / Date**	**Descrizione / Description**
rev.00	07/11	Initial release

Mecc Alte SpA
Via Roma
20 – 36051 Creazzo
Vicenza – ITALY
T: +39 0444 396111
F: +39 0444 396166
E: mecc-alte-spa@meccalte.it

After sale service email:
sat2@meccalte.it

France
Mecc Alte International S.A.
Z.E.La Gagnerie
16330 ST.Amant De Boixe
T: 0545/397562
F: 0545/398820
E: mecc.alte@meccalte.fr

After sale service email:
philippe.denis@meccalte.fr

India
Mecc Alte India PVT LTD
Plot NO: 1, Sanaswadi-Talegaon
Dhamdhere Road
Taluka: Shirur, District: Pune - 412208
Maharashtra, India
T: +91 2137 619600
F: +91 2137 619699
E: sales@meccalte.in

United Kingdom
Mecc Alte U.K. LTD
6 Lands' End Way
Oakham
Rutland
T: +44 01572/771160
F: +44 01572/771161
E: gen@meccalte.co.uk

After sale service email:
rod.marshall@meccalte.co.uk

Germany
Mecc Alte Generatoren GmbH
Ensener Weg 21
D-51149 Köln
T: 02203/503810
F: 02203/503796
E: info@meccalte.de

After sale service email:
service@meccalte.de

Far East
Mecc Alte (F.E.) PTE LTD
19 Kian Teck Drive
Singapore 628836
T: +65 62 657122
F: +65 62 653991
E: enquiry@meccalte.com.sg

After sale service email:
enquiry@meccalte.com.sg

Spain
Mecc Alte España S.A.
C/ Rio Taibilla, 2
Polig. Ind. Los Valeros
03178 Benijofar (Alicante)
T: 096/6702152
F: 096/6700103
E: gerencia@meccalte.es

After sale service email:
serviciotecnico@meccalte.es

U.S.A. and Canada
Mecc Alte Inc.
1229 Adam Drive
McHenry IL, 60051 (USA)
T: 815 344 0530
F: 815 344 0535
E: sales@meccalte.us

After sale service email:
sales@meccalte.us

Australia
Mecc Alte Alternators PTY LTD
10 Duncan Road, PO Box 1046
Dry Creek, 5094, South Australia
T: +61 (0)8 8349 8422
F: +61 (0)8 8349 8455
E: sales@meccalte.com.au

China
Mecc Alte Alternator Haimen LTD
755 Nanhai East Rd
Jiangsu HEDZ 226100 PRC
T: +86 0513 82325758
F: +86 0513 82325768
E: sales@meccalte.cn

www.meccalte.com