
Sommario

DEVICE MODBUS	3
1. Introduzione	3
1.1 Convenzioni	3
1.2 Dichiarazione device nella unit di configurazione	3
1.3 Tabelle “scambio dati” per il MODBUS	3
1.3.1 Dati letti dal device	3
1.3.2 Dati scrivibili da QCL e condivisibili in MODBUS	4
1.4 Descrizione funzionamento come master	5
1.4.1 Accesso dei dati input e coil	5
1.4.2 Modo di funzionamento MASTER	5
1.4.3 Richieste di lettura	6
1.4.4 Richieste di scrittura	7
1.4.5 Esempi	7
1.4.6 Schema a blocchi	8
1.4.7 Broadcast	8
1.4.8 Identificazione degli Slave	8
1.4.9 Estensione del protocollo	9
1.4.10 Comandi e parametri	9
1.5 Descrizione funzionamento come slave	12
1.5.1 Modo di funzionamento SLAVE	12
1.5.2 Schema a blocchi	13
1.5.3 Comandi e parametri	13
1.6 Migrazione dal device MODBUS01 al device MODBUS	15
Esempio:	16

DEVICE MODBUS

1. Introduzione

Il protocollo MODBUS® rappresenta un sistema di comunicazione industriale distribuita sviluppata da Gould-Modicon per integrare PLC's, computers, terminals, inverter e altre apparecchiature. MODBUS è un protocollo di comunicazione Master/Slave in cui solo un dispositivo può essere Master e controlla tutte le attività della linea seriale o TCPIP. Per quanto riguarda la linea seriale possono essere collegati fino ad un massimo 247 slave sulla stessa linea. Ad ogni dispositivo è assegnato un indirizzo che distingue esso da tutti gli altri dispositivi connessi.

Il device MODBUS permette di scegliere quale porta seriale, presente sullo strumento, usare per il collegamento. Le caratteristiche della trasmissione dati (nr. dispositivo fisico, baud rate, parità, stop bits) e del modo di trasmissione devono essere selezionate su ogni stazione e non possono essere modificate durante il funzionamento.

Con l'utilizzo di una linea seriale, ci sono due modi di trasmissione utilizzabili nel protocollo Modbus. Ambedue i modi permettono le stesse capacità di comunicazione. Il modo è selezionato dal master e deve essere unico per tutti i dispositivi connessi alla rete. I modi sono:

- ASCII (American Standard Code for Information Interchange)
- RTU, (Remote Terminal Unit.)

Il presente manuale si suddivide in due parti:

Parte I - Descrizione funzionamento come master

Parte II - Descrizione funzionamento come slave.

1.1 Convenzioni

Il protocollo di comunicazione modbus standard definisce 4 tipi di dati di scambio:

- INPUT STATUS = Ingressi digitali (Digital Inputs)
- COIL STATUS = Uscite digitali (Digital Outputs)
- INPUT REGISTER = Ingressi analogici (Analog Inputs)
- HOLDING REGISTER. = Uscite analogiche (Analog Outputs).

Nel corso del manuale si deve tener conto delle convenzioni presenti.

1.2 Dichiarazione device nella unit di configurazione

Dichiarazione del device utilizzato (device interno): nome del device assegnato, MODBUS, posizione dello slot e numero del device.

```
INTDEVICE
<nome_device>    MODBUS    TCamp    channel
...
```

Dove:

INTDEVICE	è la parola chiave che indica l'inizio della definizione di devices interni,
<nome_device>	è il nome del device,
MODBUS	è la parola chiave che identifica il device descritto in questo documento,
TCamp	è tempo di campionamento del device,
channel	è la definizione del tipo di canale di comunicazione collegato (0 = Prog, 1 = User, per le altre porte il numero identificativo dipende dal tipo hardware e firmware dello strumento utilizzato, 43 = Ethernet port per un modbus TCPIP).

Per quanto riguarda la dichiarazione dell'hardware utilizzato nella sezione "BUS" dell'unità di configurazione si dovrà fare riferimento al manuale firmware dell'hardware stesso.

Esempio

```
INTDEVICE
modbus           MODBUS    0002    1
```

1.3 Tabelle "scambio dati" per il MODBUS

1.3.1 Dati letti dal device

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Indice tabella
	15	14	13	12	11	10	9	8	7	6	5	4	6	2	1	0	Peso dei bit(2 ^N)
Registro Modbus																	
0001								iword1								2 ¹⁶	ilong1
0002								iword2								2 ⁰	
0003								iword3								2 ¹⁶	ilong2
0004								iword4								2 ⁰	
0005								iword5								2 ¹⁶	ilong3
0006								iword6								2 ⁰	
0007								iword7								2 ¹⁶	ilong4
0008								iword8								2 ⁰	
0009								iword9								2 ¹⁶	ilong5
0010								iword10								2 ⁰	
0011								iword11								2 ¹⁶	ilong6
0012								iword12								2 ⁰	
0013								iword13								2 ¹⁶	ilong7
0014								iword14								2 ⁰	
0015								iword15								2 ¹⁶	ilong8
0016								iword16								2 ⁰	
0017								iword17								2 ¹⁶	ilong9
0018								iword18								2 ⁰	
0019								iword19								2 ¹⁶	ilong10
0020								iword20								2 ⁰	
0021								iword21								2 ¹⁶	ilong11
0022								iword22								2 ⁰	
0023								iword23								2 ¹⁶	ilong12
0024								iword24								2 ⁰	
0025								iword25								2 ¹⁶	ilong13
0026								iword26								2 ⁰	
0027								iword27								2 ¹⁶	ilong14
0028								iword28								2 ⁰	
0029								iword29								2 ¹⁶	ilong15
0030								iword30								2 ⁰	
0031								iword31								2 ¹⁶	ilong16
0032								iword32								2 ⁰	

NOTA 1: la " iword1 " è la word più significativa della " ilong1 " (long = doppia word), la "iword3" è la word più significativa della " ilong2 ", ...

NOTA 2: il parametro " st_ibit0 " è il bit meno significativo (con peso 0) della " iword1 ", mentre il bit " st_ibit15 " è il bit più significativo (con peso 15) della " iword1 ".

1.3.2 Dati scrivibili da QCL e condivisibili in MODBUS

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Indice tabella
	15	14	13	12	11	10	9	8	7	6	5	4	6	2	1	0	Peso dei bit(2 ^N)
Registro Modbus																	
0001								oword1								2 ¹⁶	olong1
0002								oword2								2 ⁰	
0003								oword3								2 ¹⁶	olong2
0004								oword4								2 ⁰	
0005								oword5								2 ¹⁶	olong3
0006								oword6								2 ⁰	
0007								oword7								2 ¹⁶	olong4
0008								oword8								2 ⁰	
0009								oword9								2 ¹⁶	olong5
0010								oword10								2 ⁰	
0011								oword11								2 ¹⁶	olong6
0012								oword12								2 ⁰	
0013								oword13								2 ¹⁶	olong7
0014								oword14								2 ⁰	
0015								oword15								2 ¹⁶	olong8

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Indice tabella
	15	14	13	12	11	10	9	8	7	6	5	4	6	2	1	0	Peso dei bit(2 ⁿ)
0016								oword16								2 ⁰	
0017								oword17								2 ¹⁶	olong9
0018								oword18								2 ⁰	
0019								oword19								2 ¹⁶	olong10
0020								oword20								2 ⁰	
0021								oword21								2 ¹⁶	olong11
0022								oword22								2 ⁰	
0023								oword23								2 ¹⁶	olong12
0024								oword24								2 ⁰	
0025								oword25								2 ¹⁶	olong13
0026								oword26								2 ⁰	
0027								oword27								2 ¹⁶	olong14
0028								oword28								2 ⁰	
0029								oword29								2 ¹⁶	olong15
0030								oword30								2 ⁰	
0031								oword31								2 ¹⁶	olong16
0032								oword32								2 ⁰	

NOTA 1: la " oword1 " è la word più significativa della " olong1 " (long = doppia word), la "oword3" è la word più significativa della " olong2 ", ...

NOTA 2: il parametro " st_obit0 " è il bit meno significativo (con peso 0) della " oword1 ", mentre il bit " st_obit15 " è il bit più significativo (con peso 15) della " oword1 ".

1.4 Descrizione funzionamento come master

Il device MODBUS può utilizzare una porta seriale o TCP/IP per comunicare in modalità master (parametro *mode* = 0). Esso dà la possibilità (tramite parametri, stati e comandi) a tutti gli strumenti di poter colloquiare con qualsiasi apparecchiatura utilizzando protocollo Modbus.

Il device rende disponibile una area di scambio dati dove il QCL può scrivere o leggere i valori condivisi con gli slave della rete. Questa area è rappresentata da una lista di parametri del device suddivisa in due gruppi per differenziare il tipo di accesso dei dati da parte del device. La prima sezione è identificata come INPUT la seconda come OUTPUT. La sezione INPUT, per il funzionamento master, è l'area di deposito dei dati letti. Viene utilizzata quindi da tutte le funzioni di lettura (parametro *type* = 1, 2, 3 o 4). La sezione OUTPUT, nel funzionamento master, è l'area di deposito dei dati che il master deve trasferire agli slave. Viene utilizzata da tutte le funzioni di scrittura (parametro *type* = 5, 6, 15 o 16). Per semplificare lo scambio dati tra apparecchiature, la struttura dell'area di scambio include tipi di dato comunemente usati in QCL e cioè WORD, LONG e FLAG. Il tipo "BYTE" non compare perché il modbus utilizza solitamente la dimensione word per ogni indirizzo dei registri. Siccome il protocollo modbus seleziona i dati in base al numero di registro di accesso, mentre il QCL seleziona in base al simbolo del parametro, si crea una tabella di riferimento in cui ogni simbolo QCL trova il corrispondente numero del registro di accesso nel protocollo Modbus. Per ottenere una maggior flessibilità di indirizzamento, la stessa area di scambio dati può essere manipolata dal QCL con parametri di dimensione diversa. Tutti i simboli che compaiono nella stessa riga della tabella si riferiscono alle medesime locazioni di memoria.

1.4.1 Accesso dei dati input e coil

I dati tipo input o coil hanno nel protocollo modbus una dimensione a bit. Per la memorizzazione di queste informazioni nella tabella di scambio dati è definita la seguente modalità: ogni registro (e quindi ogni parametro word del device) rappresenta 16 elementi che vengono numerati da sinistra verso destra. Per le scritture o letture multiple quando i bit da leggere sono superiori a quelli contenuti nel registro l'operazione continua nel registro successivo.

Gli stati *st_ibit0...* e *st_obit1...* sono utili per agevolare l'accesso alle informazioni di tipo bit solo nel primo registro della tabella.

1.4.2 Modo di funzionamento MASTER

La modalità master viene impostata con il parametro *mode* = 0. La scheda genera messaggi modbus solamente su comando del QCL e non automaticamente. Per "formattare" la stringa del messaggio modbus vi sono una serie di parametri.

Mediante i due comandi SEND o SENDSYN si inviano la richiesta di lettura o scrittura allo slave, una serie di stati permettono una verifica sull'operazione eseguita. Vediamo in dettaglio:

1.4.2.1 Parametri per formattare il messaggio:

type = tipo di richiesta che si vuole processare (lettura o scrittura di word o di bit).

idcard = indirizzo dell'unità slave verso cui si vuole inviare la richiesta.

addr = indice da cui iniziare a leggere o da cui iniziare a scrivere nello slave in questione.

tabreg = per le operazioni di lettura (type = 1, 2, 3, 4), questo parametro indica la posizione nella "tabella dei dati letti" dove verrà memorizzato il dato letto sullo slave. Per le operazioni di scrittura (type = 5, 6, 15, 16), questo parametro indica da posizione nella "tabella dei dati da scrivere" dove si trova il dato da scrivere sullo slave. Sono previsti valori tra 1 e 512 per valori del parametro type = 1, 2, 5, 15 (scrittura e lettura di bit), mentre per valori type = 3, 4, 6, 16 (scrittura e lettura di word) i valori previsti sono tra 1 e 32.

num = numero di valori che si intendono leggere o scrivere (se il tipo richiesta type lo prevede). Sono previsti valori tra 1 e 128 per valori del parametro type = 1, 2, (lettura di bit), mentre per valori type = 3, 4, 16 (scrittura multipla e lettura di word) i valori previsti sono tra 1 e 32.

1.4.2.2 Comandi per inviare il messaggio:

SEND invia il messaggio di richiesta allo slave.

SENDSYC invia il messaggio di richiesta ad uno slave QEM di tipo MODBUS e segnala l'operazione conclusa (*st_sended*) solamente quando lo slave esegue il comando RECEIVED.

1.4.2.3 Variabili di Stato per diagnostica operazione:

Quando l'operazione di trasmissione messaggio è conclusa, viene impostata a 1 la variabile di stato *st_sended*. Se è stato utilizzato il comando SEND lo stato viene posto a 1 quando il master riceve la conferma del messaggio inviato. Se invece è stato utilizzato nel master il comando SENDSYC allora questo stato verrà impostato a 1 quando lo slave QEM avrà eseguito il comando RECEIVED. In caso di errore, invece, questo stato viene posto comunque a 1 (dopo il tempo *toutsyc*) ma il parametro *err* segnala la causa dell'errore (vedere la descrizione di questo parametro per i dettagli).

Il comando SENDSYC si rende necessario quando si vuole sincronizzare la scrittura del master di un pacchetto dati con la lettura degli stessi dati da parte dello slave (può essere usato quando sia master che slave sono Qmove).

1.4.3 Richieste di lettura

Read Coil

Funzione di lettura a bit. Si vogliono leggere i bit di un certo slave (*idcard*). Viene impostato il parametro *type* = 1.

Bisogna impostare *addr* per indicare l'indirizzo del bit nello slave da cui si intende iniziare a leggere. Il parametro *num* indica il numero di bit che si intendono leggere. Bisogna impostare *tabreg* per indicare l'indice del bit nella "tabella dei dati letti" del master da cui si vuole che siano scritti i bit letti. I valori letti saranno salvati nella "tabella dei dati letti". Si raccomanda di soddisfare sempre le seguenti due condizioni:

$(addr + num - 1) \leq 512$ e $(tabreg + num - 1) \leq 512$ altrimenti la richiesta non potrà essere eseguita (segnalazione mediante *wcmd*).

Read Input

Funzione di lettura a bit. Si vogliono leggere i bit di un certo slave (*idcard*). La funzione è identica alla "Read Coil". Viene impostato il parametro *type* = 2.

Il significato delle due funzioni si distingue a livello di protocollo MODBUS. Alcuni slave possono mettere a disposizione delle zone di memoria differenti a seconda se i bit letti sono Coil o Input. Nel caso di slave Qmove le due richieste hanno lo stesso effetto.

Read Holding Register

Funzione di lettura a word. Si vogliono leggere le word di un certo slave (*idcard*). Viene impostato il parametro *type* = 3.

Bisogna impostare *addr* per indicare l'indirizzo della word nello slave da cui si intende iniziare a leggere. Il parametro *num* indica il numero di word che si intendono leggere. Bisogna impostare *tabreg* per indicare l'indice della word nella "tabella dei dati letti" del master da cui si vuole che siano scritte le word lette. I valori letti verranno salvati nella "tabella dei dati letti". Si raccomanda di soddisfare sempre le seguenti due condizioni:

$(addr + num - 1) \leq 32$ e $(tabreg + num - 1) \leq 32$ altrimenti la richiesta non potrà essere eseguita (segnalazione mediante *wcmd*).

Read Input Register

Funzione di lettura ingressi a word. Si vogliono leggere i bit di un certo slave (*idcard*). La funzione è identica alla "Read Holding Register". Viene impostato il parametro *type* = 4.

Il significato delle due funzioni si distingue a livello di protocollo MODBUS. Alcuni slave possono mettere a disposizione delle

zone di memoria differenti a seconda se le word lette sono Holding o Input. Nel caso di slave Qmove le due richieste hanno lo stesso effetto.

1.4.4 Richieste di scrittura

Force coil

Funzione di scrittura bit. Si vogliono scrivere (forzare) i bit di un certo slave (*idcard*). Viene impostato il parametro *type* = 5. Bisogna impostare *addr* per indicare l'indirizzo del bit nello slave in cui si intende scrivere. Bisogna impostare *tabreg* per indicare l'indice del bit nella "tabella dei dati da scrivere" in cui è contenuto il valore da scrivere.

Force register

Funzione di scrittura singola word. Si vuole scrivere (forzare) la word di un certo slave (*idcard*). Viene impostato il parametro *type* = 6.

Bisogna impostare *addr* per indicare l'indirizzo della word nello slave in cui si intende scrivere. Bisogna impostare *tabreg* per indicare l'indice della word nella "tabella dei dati da scrivere" in cui è contenuto il valore da scrivere.

Force multiple coil

Funzione di scrittura bit multipli. Si vogliono scrivere (forzare) i bit di un certo slave (*idcard*). Viene impostato il parametro *type* = 15.

Bisogna impostare *addr* per indicare l'indirizzo del bit nello slave in cui si intende scrivere. Si deve impostare in *num* il numero di bit da scrivere. Bisogna impostare *tabreg* per indicare l'indice del bit nella "tabella dei dati da scrivere" da cui iniziano i valori da scrivere. Si raccomanda di soddisfare sempre le seguenti due condizioni:

$(addr + num - 1) \leq 512$ e $(tabreg + num - 1) \leq 512$ altrimenti la richiesta non potrà essere eseguita (segnalazione mediante *wcmd*).

Force multiple register

Funzione di scrittura word multiple. Si vogliono scrivere (forzare) le word di un certo slave (*idcard*). Viene impostato il parametro *type* = 16.

Bisogna impostare *addr* per indicare l'indirizzo della word nello slave in cui si intende scrivere. Si deve impostare in *num* il numero di word da scrivere. Bisogna impostare *tabreg* per indicare l'indice della word nella "tabella dei dati da scrivere" da cui iniziano i valori da scrivere. Si raccomanda di soddisfare sempre le seguenti due condizioni:

$(addr + num - 1) \leq 32$ e $(tabreg + num - 1) \leq 32$ altrimenti la richiesta non potrà essere eseguita (segnalazione mediante *wcmd*).

1.4.5 Esempi

1) Impostazioni modbus e apertura del canale di comunicazione di tipo seriale

```

modbus.mode = 0           ; modbus master
modbus.prot = 1           ; RTU
modbus.brate = 57600      ; baudrate
modbus.stopb = 1         ; stop bits
modbus.par = 0            ; parity
modbus.tout = 100        ; timeout

modbus.OPENCOM
tm_opencom = 200         ; timeout for OPENCOM command
WAIT modbus.st_opencom OR tm_opencom
IF NOT modbus.st_opencom
  OpenComError = 1       ; com not opened
ENDIF

```

2) Impostazioni modbus e apertura del canale di comunicazione di tipo TCPIP

```

modbus.mode = 0           ; modbus master
modbus.prot = 2           ; TCPIP
modbus.tout = 100        ; timeout

tcpip_port = 502         ; number of tcpip port (default 502)
modbus.stopb = (tcpip_port ANDB &H0000FF00) / 256
modbus.par = tcpip_port ANDB &H000000FF

modbus.OPENCOM
tm_opencom = 500         ; waiting time to have a correct response
WAIT tm_opencom
IF NOT modbus.st_opencom
  OpenComError = 1       ; com not opened
ENDIF

```

3) Lettura del registro input nr.7 dello slave id 67

```

modbus.idcard = 67

```

```

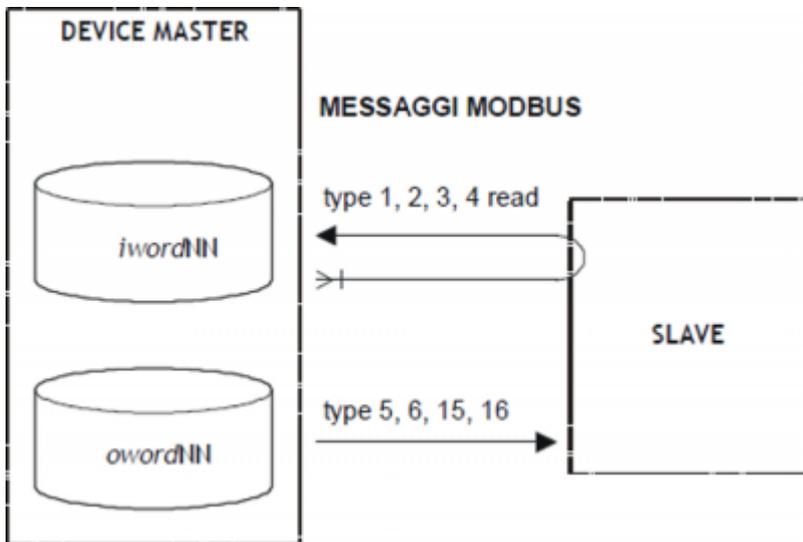
modbus.addr = 7
modbus.tabreg = 7
modbus.num = 1
modbus.type = 4
modbus.SEND
WAIT modbus.st_sended
IF (modbus.err)
CALL ErroreModbus
ENDIF
slVelocita = modbus.iword7
    
```

4) Lettura dei holding register dal nr.600 al nr.606 dello slave id 8
 Siccome l'area di scambio dati QCL non supporta l'indirizzo 600, si utilizza il parametro tabreg per salvare i dati letti all'indirizzo 10

```

modbus.idcard = 8
modbus.addr = 600
modbus.num = 6
modbus.tabreg = 10
modbus.type = 3
modbus.SEND
WAIT modbus.st_sended
IF (modbus.err)
CALL ErroreModbus
ENDIF
swTemp1 = modbus.iword10
slVelocita = modbus.iling6
slPosiz = modbus.iling7
swTemp1 = modbus.iword15
    
```

1.4.6 Schema a blocchi



1.4.7 Broadcast

Il Broadcast è un messaggio speciale che viene inviato dal Master e viene ricevuto contemporaneamente da tutti gli slave connessi. La funzionalità broadcast (idcard = 0) è supportata solamente in modo master e se viene usato un messaggio con type = 5, 6, 15 e 16 (solo scrittura). In questa modalità lo stato st_sended viene attivato appena terminata la trasmissione e non alla ricezione della conferma dallo slave. Infatti non esiste nessuna risposta degli slave per questo tipo di messaggi. Non è possibile utilizzare il comando SENDSYC.

1.4.8 Identificazione degli Slave

La scheda quando funziona in modalità Master permette, con un speciale comando, di riconoscere se lo slave collegato è di tipo QEM. In caso affermativo determina anche il modello di slave connesso. Questo permette di realizzare una diagnostica sulla corretta configurazione della rete. Non è possibile estendere questo controllo a tutti i tipi di slave esistenti perché il protocollo non prevede una modalità standard di riconoscimento slave.

Il comando READSTYPE eseguito dopo aver opportunamente selezionato lo slave (tramite parametro idcard) invia allo slave una richiesta modbus nr.17 per il riconoscimento. Lo slave risponderà con un suo valore predefinito e potrà essere letto nel parametro stype del master. Nel caso lo slave non sia prodotto da QEM srl, il parametro avrà valore -1 ad indicare che lo slave è connesso ma non è possibile identificarlo.

Per il device MODBUS lo slave ID è diverso a seconda dello slave QEM utilizzato. Per una codifica degli slave in questo senso vedere il manuale Firmware dello slave utilizzato.

Esempio d'identificazione dello slave nr.12:

```

modbus.idcard = 12
modbus.READSTYPE
WAIT modbus.st_sended
    
```

```

IF (modbus.err)
  sbConfig = 3           ; error: slave 12 not connected
ELSE
  IF (modbus.stype EQ 1) ; Board CAM-AG-98
    sbConfig = 1         ; configuration OK
  ELSE
    sbConfig = 2         ; error in network configuration
  ENENDIF
ENDIF

```

1.4.9 Estensione del protocollo

Il protocollo di comunicazione modbus standard definisce 4 tipi di dati di scambio:

- INPUT STATUS,
- COIL STATUS,
- INPUT REGISTER,
- HOLDING REGISTER.

Questi 4 tipi trattano i segnali tipo Digital Inputs, Digital Outputs, Analog Inputs, and Analog Outputs. I valori dei registri (sia INPUT REGISTERS che HOLDING REGISTERS), sono definiti come valori a 16 bit, ognuno con un unico indirizzo, (esempio INPUT REGISTER #1, INPUT REGISTER #2, INPUT REGISTER #3, ecc.).

Per permettere di supportare tipi di dato di dimensione maggiore di 16 bit, è stata progettata una estensione al protocollo per trattare valori di dimensione long 32 bit. L'implementazione di questa funzionalità mantiene la formattazione del messaggio standard modbus e utilizza il contenuto di due indirizzi consecutivi per rappresentare un valore 32 bit. Quando questa estensione è attiva (wider = 1), a seconda dell'indirizzo utilizzato lo slave risponderà con un valore a 16 bit o 32 bit per ogni richiesta di registro. Se l'indirizzo è superiore a 5000 verrà risposto con valore a 32 bit, se inferiore il protocollo rimane invariato con risposta a 16 bit.

1.4.10 Comandi e parametri

1.4.10.1 Simbologia adottata

Il nome del parametro, stato o comando viene riportato alla sinistra della tabella.

R

Indica se il relativo parametro o stato è ritentivo (al momento dell'inizializzazione del device mantiene lo stato precedentemente definito), oppure lo stato che assume al momento dell'inizializzazione del device.

Se il device non necessita di inizializzazione il campo "R" indica il valore che il parametro o stato assume all'accensione della scheda.

R = Ritentivo

0 = Al momento dell'inizializzazione del device il valore viene forzato a zero.

1 = Al momento dell'inizializzazione del device il valore viene forzato a uno.

- = Al momento dell'inizializzazione del device viene presentato il valore significativo.

D

Indica la dimensione del parametro.

F = Flag

B = Byte

W = Word

L = Long

S = Single Float

Condizioni

Vengono descritte tutte le condizioni necessarie affinché il parametro sia considerato corretto o perché il comando venga accettato.

In alcuni casi vengono specificati dei valori limite per l'accettazione del parametro: se vengono introdotti dei valori esterni ai limiti impostati, il dato viene comunque accettato; pertanto devono essere previsti opportuni controlli dell'applicativo tali da garantire il corretto funzionamento.

Per l'esecuzione di un comando, tutte le relative condizioni devono necessariamente essere soddisfatte; in caso contrario il comando non viene eseguito.

A

Indica la modalità di accesso.

R = Read (lettura).

W = Write (scrittura).

RW = Read / Write.

1.4.10.2 Parametri

Nome	D	R	A	Condizioni	Descrizione
mode	B	R	R/W	-	Mode (0 ÷ 2) Definisce se la scheda deve funzionare come master o come slave. 0 = master.
prot	B	R	R/W	-	Protocol (0 ÷ 1) Definisce il tipo di protocollo Modbus da utilizzare. 0 = ASCII, 1 = RTU, 2 = TCP/IP
wider	B	R	R/W	-	Wide Registers (0 ÷ 1) Indica se utilizzare l'estensione del protocollo per i registri a 32bit. Vedere capitolo "Estensione del protocollo": 0 = normale, 1 = protocollo esteso.
idcard	W	R	R/W	-	Identification Card (0 ÷ 255) Nel funzionamento come master è il numero del dispositivo al quale s'intende trasmettere. Nel modo master è valida anche l'impostazione zero per il funzionamento broadcast.
type	B	1	R/W	-	Type Definisce il tipo di richiesta che il master deve eseguire. Valori ammessi: 1, 2, 3, 4, 5, 6, 15, 16
addr	L	1	R/W	-	Address (1 ÷ 65535) Definisce l'indirizzo utilizzato dal master per definire quale sia il dato da leggere o scrivere.
tabreg	W	1	R/W	-	Destination (1 ÷ 512) Definisce l'indirizzo nelle tabelle dei dati in cui scrivere il dato letto o in cui trovare il dato da scrivere.
num	W	1	R/W	-	Number (1 ÷ 512) È il numero d'elementi da scrivere o leggere nella richiesta che compone il master.
brate	L	R	R/W	-	Baud rate Baud rate della seriale. Valori validi: 4800, 9600, 19200, 38400, 57600, 115200.
stopb	B	R	R/W	-	Stop bit Valori validi: 1, 2.
par	B	R	R/W	-	Parity (0 ÷ 2) 0 = none (nessuna), 1 = odd (dispari), 2 = even (pari).
tout	W	R	R/W	-	Timeout (0 ÷ 9999 msec) Per la modalità master è il tempo massimo in cui lo slave deve rispondere. Impostandolo a zero il Timeout è disabilitato.
toutsyc	W	R	R/W	-	Time out synchronize (0 ÷ 9999 msec) Utilizzato solamente per la modalità master è il tempo massimo che può trascorrere tra l'invio del comando SENDSYN e l'esecuzione del comando RECEIVED nello slave QEM.
iword1÷32	W	0	R	-	Input Word nr. (1Input Long nr. (1÷16)32)
st_ibit0÷15	F	0	R	-	Input bit nel parametro iword1
ilong1÷16	L	0	R	-	Input Long nr. (1÷16)
oword1÷32	W	0	R/W	-	Output Word nr. (1÷32)
st_obit0÷15	F	0	R/W	-	Output bit nel parametro oword1
olong1÷16	L	0	R/W	-	Output Long nr. (1÷16)

La possibilità di comunicare con il protocollo TCP/IP è stata sviluppata in un tempo successivo rispetto alla nascita del device MODBUS. Per questo motivo l'impostazione della porta e dell'indirizzo IP viene eseguita mediante alcuni parametri che hanno un significato diverso.

Per configurare il numero della porta TCP/IP è necessario andare a scrivere sui parametri *stopb* e *par*. Rispettivamente :

- **stopb** = tcpip_port / &H00000100
- **par** = tcpip_port ANDB &H000000FF

Per impostare lo slave TCP/IP a cui si vuole eseguire la richiesta è necessario andare a scrivere sul parametro *brate*. Un indirizzo IP può essere espresso come TcpAddr3.TcpAddr2.TcpAddr1.TcpAddr0 (per esempio 192.168.0.1).

- **brate** = (TcpAddr3 * &H01000000) + (TcpAddr2 * &H00100000) + (TcpAddr1 * &H00001000) + TcpAddr0

1.4.10.3 Variabili

Nome	D	R	A	Condizioni	Descrizione
err	B	0	R	-	<p>Errors Indica se sono stati riscontrati errori nel protocollo. In modalità master il parametro viene aggiornato in coincidenza dell'attivazione dello stato <i>st_sended</i>. I valori da 1 a 49 sono quelli inviati dallo slave. I valori superiori a 50 sono generati internamente dal master. I primi 8 valori sono normalizzati nel protocollo e sono: 0 = comunicazione avvenuta senza errori, 1 = ILLEGAL FUNCTION, 2 = ILLEGAL DATA ADDRESS, 3 = ILLEGAL DATA VALUE, 4 = SLAVE DEVICE FAILURE, 5 = ACKNOWLEDGE, 6 = SLAVE DEVICE BUSY, 7 = NEGATIVE ACKNOWLEDGE, 8 = MEMORY PARITY ERROR, 50 = UNKNOWN RECEIVED ERROR lo slave ha risposto con un codice superiore a 49, 51 = TIMEOUT lo slave non ha risposto entro il tempo programmato nel parametro tout, 52 = INVALID ANSWER, 52 = INVALID FUNCTION ANSWER - Tipo funzione non supportata, 53 = CHECKSUM ANSWER - Il checksum ricavato non corrisponde, 54 = TRUNCATED ANSWER - Messaggio troppo corto, 55 = INVALID ID ANSWER - Id slave non corrispondente, 56 = INVALID_MBAP_TRANSACTION_ID - modbus TCP/IP, <i>transaction ID</i> non corretto, 57 = INVALID_MBAP_PROTOCOL_ID - modbus TCP/IP, <i>protocol ID</i> diverso da zero, 70 = TIMEOUT SYNCHRONIZED lo slave QEM non hanno eseguito il comando RECEIVED entro il tempo programmato nel parametro toutsync. In modalità slave il parametro non viene mai aggiornato.</p>
serr	B	0	R/W	-	<p>Serial Errors Indica se sono stati riscontrati errori nella comunicazione seriale. Il parametro viene aggiornato ad ogni errore rilevato. Il valore permane fino: - ad un successivo errore; - alla scrittura di uno dei seguenti parametri: mode, prot, wider, brate, stopb, par; - ad una scrittura sullo stesso (qualsiasi valore lo azzera). 0 = nessun errore, 1 = parity error, 2 = framing error, 3 = overrun error.</p>
stype	W	0	R	-	<p>Slave type Indica il tipo di slave connesso. Il parametro viene aggiornato quando il parametro <i>st_sended</i> diventa 1 e se era stato eseguito un comando READSTYPE</p>

1.4.10.4 Stati

Nome	D	R	A	Condizioni	Descrizione
st_sended	F	0	R	-	<p>Sended Stato utilizzato solo in modalità master. L'attivazione indica il completamento della trasmissione di un messaggio. Lo stato viene resettato con i comandi SEND o SENDSYN.</p>
st_opencom	F	0	R	-	<p>Open communication port L'attivazione indica che il device sta impegnando la porta di comunicazione seriale. Per settare questo stato usare il comando OPENCOM, per resettarlo CLOSECOM.</p>
wdata	F	0	R	-	<p>Warning Data Questo bit segnala che è stato tentato un inserimento di un valore non valido in un parametro.</p>
wcmd	F	0	R	-	<p>Warning Command Questo bit segnala che è non è stato eseguito un comando perché mancano le condizioni necessarie.</p>

1.4.10.5 Comandi

I comandi a disposizione per gestire il device sono elencati sotto in ordine di priorità decrescente. Il device esegue tutti i comandi ricevuti entro lo stesso tempo di campionamento iniziando da quello con la priorità maggiore. Per esempio se il device riceve nello stesso tempo di campionamento i comandi CLOSECOM e OPENCOM, per primo esegue il comando OPENCOM e poi quello di CLOSECOM lasciando perciò la porta di comunicazione chiusa.

Nome	D	R	A	Condizioni	Descrizione
SEND	-	-	-	mode = 0 0 < num ≤ 128 st_sended = 1 st_opencom = 1	<p>Sended Determina la trasmissione del messaggio verso lo slave selezionato.</p>

Nome	D	R	A	Condizioni	Descrizione
SENDSYC	-	-	-	mode = 0 0 < num ≤ 128 st_sended = 1 st_opencom = 1	Send synchronize Utilizzabile solo in modo master determina la trasmissione del messaggio verso uno slave QEM con la sincronizzazione.
READSTYPE	-	-	-	mode = 0 st_sended = 1 st_opencom = 1	Read slave type Richiesta lettura informazione type sullo slave specificato in idcard. La risposta verrà segnalata sul parametro stype quando lo stato st_sended diventa 1.
CLRWDATA	-	-	-	-	Clear Warning Data Azzera la segnalazione del parametro wdata.
CLRWCMD	-	-	-	-	Clear Warning Command Azzera la segnalazione del parametro wcmd.
OPENCOM	-	-	-	st_opencom = 0	Open Serial communication \ Apre la comunicazione seriale (il device quindi impegna la porta di comunicazione). Lo stato st_opencom diventa 1.
CLOSECOM	-	-	-	-	Close Serial communication Chiude la comunicazione seriale (il device quindi non impegna più la porta di comunicazione). Lo stato st_opencom diventa 0.

1.5 Descrizione funzionamento come slave

1.5.1 Modo di funzionamento SLAVE

La modalità slave viene impostata con il parametro mode = 1 o mode = 2. Bisogna introdurre nei parametri il codice dell'ID slave tramite il parametro idcard.

Impostazioni modbus e apertura del canale di comunicazione di tipo seriale

```

modbus.mode = 1 ; modbus slave
modbus.prot = 1 ; RTU
modbus.brates = 57600 ; baudrate
modbus.stopb = 1 ; stop bits
modbus.par = 0 ; parity
modbus.idcard = 1 ; slave id

modbus.OPENCOM
tm_opencom = 100 ; timeout for OPENCOM command
WAIT modbus.st_opencom OR tm_opencom
IF NOT modbus.st_opencom
    OpenComError = 1 ; com not opened
ENDIF
    
```

Impostazioni modbus e apertura del canale di comunicazione di tipo TCP/IP

```

modbus.mode = 1 ; modbus slave
modbus.prot = 2 ; TCP/IP

tcpip_port = 502 ; number of tcpip port (default 502)
modbus.stopb = (tcpip_port ANDB &H0000FF00) / 256
modbus.par = tcpip_port ANDB &H000000FF

modbus.OPENCOM
tm_opencom = 100 ; waiting time to have a correct response
WAIT (modbus.st_opencom OR timerOpencom)
IF NOT modbus.st_opencom
    OpenComError = 1 ; com not opened
ENDIF
    
```

Le operazioni che spettano all'applicativo QCL sono essenzialmente due:

- aggiornare i parametri tipo owordNN, olongNN (che verranno poi letti dal master)
- processare i parametri tipo iwordNN, ilongNN (che sono stati scritti dal master)

1.5.1.1 Aggiornamento dei parametri tipo owordNN

Per aggiornare i parametri tipo owordNN il QCL deve scrivere i valori nei parametri quindi inviare il comando VALIDATE. L'utilizzo del comando si rende necessario per rendere disponibili alle letture del master i nuovi dati nello stesso istante, evitando così che possano venire letti solamente parte dei parametri aggiornati. Quindi l'assegnazione di un valore ad un parametro device NON SIGNIFICA che esso sia immediatamente disponibile per le letture del master. Sarà disponibile solamente dopo il comando VALIDATE.

Esempio

```

modbus.olong1 = anpos.speed
modbus.oword1 = npezzi
modbus.oword2 = npezzi_tot
modbus.oword3 = anpos.dir
modbus.olong2 = anpos.posit
modbus.VALIDATE
    
```

1.5.1.2 Processo dei parametri tipo iwordNN

Per processare i parametri tipo *ywordNN*, invece, il QCL deve attendere che lo stato *st_msgrx* venga attivato. Questo significa che un messaggio di scrittura è stato inviato dal master. Per i messaggi di lettura nessun segnale viene riferito ai parametri QCL. Il QCL poi deve prelevare i valori dall'area scambio dati e al termine deve inviare il comando RECEIVED, solo allora lo stato *st_msgrx* viene disattivato.

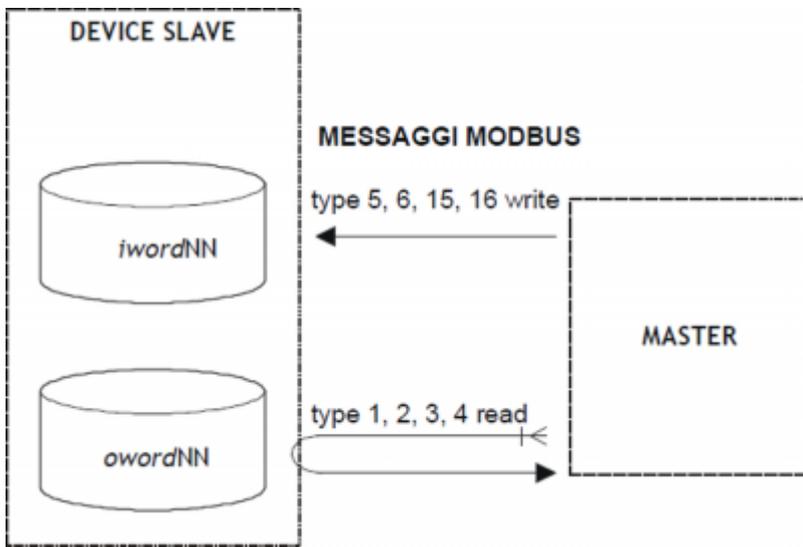
Esempio

```
WAIT modbus.st_msgrx
anpos.speed = modbus.ulong1
anpos.posit = modbus.ulong2
npezzi = modbus.ushort1
RECEIVED modbus
```

Naturalmente non esiste nessun controllo per evitare che durante la lettura dei parametri *ywordNN* il master erroneamente esegua un'altra scrittura, prima che il codice del progetto presente nello slave finisca il processo. Inoltre il codice per la manipolazione del device modbus nell'applicativo è consigliato che sia raccolto in un unico task in modo da non creare accessi contemporanei ai device che potrebbero generare delle condizioni non valide in ogni singolo contesto.

Tutte le richieste di lettura che arrivano dal master MODBUS fanno riferimento ai dati di tipo *ywordNN* accedendo quindi alla stessa area dati. Lo slave mette a disposizione i valori che il master leggerà nei parametri di tipo *ywordNN*. Tutte le richieste di scrittura che arrivano dal master MODBUS fanno riferimento ai dati di tipo *ywordNN* accedendo quindi alla stessa area dati. Lo slave mette a disposizione i parametri *ywordNN* dove il master potrà scrivere i dati.

1.5.2 Schema a blocchi



1.5.3 Comandi e parametri

1.5.3.1 Simbologia adottata

Il nome del parametro, stato o comando è riportato alla sinistra della tabella.

R

Indica se il relativo parametro o stato è ritentivo (al momento dell'inizializzazione del device mantiene lo stato precedentemente definito), oppure lo stato che assume al momento dell'inizializzazione del device. Se il device non necessita di inizializzazione il campo "R" indica il valore che il parametro o stato assume all'accensione della scheda.

R = Ritentivo

0 = Al momento dell'inizializzazione del device il valore è forzato a zero.

1 = Al momento dell'inizializzazione del device il valore è forzato ad uno.

- = Al momento dell'inizializzazione del device è presentato il valore significativo.

D

Indica la dimensione del parametro.

F = Flag

B = Byte

W = Word

L = Long

S = Single Float

Condizioni

Sono descritte tutte le condizioni necessarie affinché il parametro sia considerato corretto o perché il comando venga accettato.

In alcuni casi sono specificati dei valori limite per l'accettazione del parametro: se sono introdotti dei valori esterni ai limiti impostati, il dato è comunque accettato; pertanto devono essere previsti opportuni controlli dell'applicativo tali da garantire il corretto funzionamento.

Per l'esecuzione di un comando, tutte le relative condizioni devono necessariamente essere soddisfatte; in caso contrario il comando non è eseguito.

A

Indica la modalità d'accesso.

R = Read (lettura).

W = Write (scrittura).

RW = Read / Write.

1.5.3.2 Parametri

Nome	D	R	A	Condizioni	Descrizione
mode	B	R	R/W	-	<p>Mode (0 ÷ 2) Definisce se la scheda deve funzionare come master o come slave. 1 = slave con area di scambio dati fissa; 2 = slave senza area di scambio dati fissa: ad ogni scrittura o lettura eseguita dal master viene messo a 1 il flag <i>st_msgrx</i>. Contemporaneamente i parametri <i>type</i>, <i>addr</i> e <i>num</i> indicano le caratteristiche del messaggio ricevuto. - Type indica il tipo di funzione che vuole fare il master (lettura o scrittura), - addr indica l'indirizzo dell'inizio del blocco di dati scritti o letti, - num indica il numero di dati letti o scritti (a bit o a byte a seconda del valore di <i>type</i>). Il device controllerà, in ricezione, solamente <i>num</i> (size di messaggio) che non sia superiore a 32word ma non più l'indirizzo che può essere ora qualsiasi. Se è una scrittura (lo si capisce dal valore di <i>type</i> se è 5,6,15,16) vengono riportati i dati nelle variabili di scambio input (sempre a partire dalla prima word) ed il device attende il comando RECEIVED. Se è una lettura il device attende che il QCL depositi i dati nelle word variabili di scambio output ed attende il comando VALIDATE.</p>
prot	B	R	R/W	-	<p>Protocol Definisce il tipo di protocollo Modbus da utilizzare. 0 = ASCII, 1 = RTU, 2 = TCP/IP.</p>
wider	B	R	R/W	-	<p>Wide Registers Indica se utilizzare l'estensione del protocollo per i registri a 32bit. Vedere capitolo "Estensione del protocollo". 0 = normale, 1 = protocollo esteso.</p>
idcard	W	R	R/W	-	<p>Identification Card (1 ÷ 255 per slave) Nel funzionamento come slave è il numero che identifica l'apparecchio nella rete.</p>
brate	L	R	R/W	-	<p>Baud rate Baud rate della seriale. Valori validi: 4800, 9600, 19200, 38400, 57600, 115200.</p>
stopb	B	R	R/W	-	<p>Stop bit Valori validi: 1, 2.</p>
par	B	R	R/W	-	<p>Parity (0 ÷ 2) 0 = none, 1 = even, 2 = odd.</p>
rdelay	W	R	R/W	-	<p>Reply delay (0 ÷ 9999 msec) È il tempo di attesa prima di trasmettere la risposta.</p>
iword1÷32	W	0	R	-	Input Word nr. (1Input Long nr. (1÷16)32)
st_ubit0÷15	F	0	R	-	Input bit nel parametro iword1
ilong1÷16	L	0	R	-	Input Long nr. (1÷16)
oword1÷32	W	0	R/W	-	Output Word nr. (1÷32)
st_obit0÷15	F	0	R/W	-	Output bit nel parametro oword1
olong1÷16	L	0	R/W	-	Output Long nr. (1÷16)

1.5.3.3 Variabili

Nome	D	R	A	Condizioni	Descrizione
serr	B	0	R/W	-	Serial Errors Indica se sono stati riscontrati errori nella comunicazione seriale. Il parametro viene aggiornato ad ogni errore rilevato. Il valore permane fino ad un successivo errore o ad una scrittura sullo stesso con il QCL. 0 = nessun errore, 1 = parity error, 2 = framing error, 3 = overrun error.

1.5.3.4 Stati

Nome	D	R	A	Condizioni	Descrizione
st_msgrx	F	0	R	-	Message received Stato utilizzato solo in modalità slave. Indica che un messaggio del master ha eseguito una scrittura nell'area scambio dati QCL. Lo stato viene resettato con il comando RECEIVED.
st_opencom	F	0	R	-	Open communication port L'attivazione indica che il device sta impegnando la porta di comunicazione seriale. Per settare questo stato usare il comando OPENCOM, per resettarlo CLOSECOM.
wdata	F	0	R	-	Warning Data Questo bit segnala che è stato tentato un inserimento di un valore non valido in un parametro.
wcmd	F	0	R	-	Warning Command Questo bit segnala che non è stato eseguito un comando perchè mancano le condizioni necessarie.

1.5.3.5 Comandi

I comandi a disposizione per gestire il device sono elencati sotto in ordine di priorità decrescente. Il device esegue tutti i comandi ricevuti entro lo stesso tempo di campionamento iniziando da quello con la priorità maggiore. Per esempio se il device riceve nello stesso tempo di campionamento i comandi CLOSECOM e OPENCOM, per primo esegue il comando OPENCOM e poi quello di CLOSECOM lasciando perciò la porta di comunicazione chiusa.

Nome	D	R	A	Condizioni	Descrizione
RECEIVED	-	-	-	st_msgrx = 1 mode = 1, 2 st_opencom = 1	Received Utilizzato solamente in modalità slave. Indica che l'applicativo QCL ha processato le informazioni che il master ha inviato
VALIDATE	-	-	-	mode = 1, 2 st_opencom = 1	Validate Indica che i parametri di scambio dati QCL diventano disponibili al master.
CLRWDATA	-	-	-	-	Clear Warning Data Azzerla la segnalazione del parametro wdata
CLRWCMD	-	-	-	-	Clear Warning Command Azzerla la segnalazione del parametro wcmd
OPENCOM	-	-	-	st_opencom = 0	Open Serial communication Apre la comunicazione seriale (il device quindi impegna la porta di comunicazione). Lo stato <i>st_opencom</i> diventa 1.
CLOSECOM	-	-	-	-	Close Serial communication Chiude la comunicazione seriale (il device quindi non impegna più la porta di comunicazione). Lo stato <i>st_opencom</i> diventa 0.

1.6 Migrazione dal device MODBUS01 al device MODBUS

Il primo aspetto da tenere presente per la migrazione è il fatto che il device MODBUS01 è esterno mentre il MODBUS è interno. Questo implica che la dichiarazione del device deve essere eseguita nella sezione "INTDEVICE" dell'unità di configurazione del progetto. Inoltre, questo comporta la scelta, durante la definizione del device MODBUS, del tempo di campionamento. Il valore del tempo di campionamento che mantiene inalterate le prestazioni della comunicazione tra i due device è pari a 5 ms. Bisogna inoltre, in fase di definizione, selezionare la seriale da impegnare.

Il device MODBUS ha aumentato il numero di variabili di scambio (da 16 word a 32 word) ma questo non influenza il funzionamento per le impostazioni che utilizzano al massimo 16 word come sono quelle pensate per il device MODBUS01. Il device MODBUS ha limitato la variabile "num" a 128, mentre per il device MODBUS01 il limite è 256. Questo si è reso necessario per permettere un'ottimizzazione del tempo di campionamento. La scelta è stata presa in considerazione del fatto che tali valori venivano impiegati unicamente per la gestione dei Coils, ma che perdeva di logica la lettura/scrittura di più di 128 coils consecutivi.

Il device MODBUS consente di controllare (mediante 2 nuovi comandi) l'impegno della seriale di comunicazione da parte del device. Questa nuova funzionalità è interessante nell'eventualità che in un progetto Qview siano stati dichiarati altri devices che possono utilizzare la stessa seriale.

Se un applicativo era stato scritto per il device MODBUS01 bisogna aggiungere il comando OPENCOM (eseguito anche solo una

volta) come mostrato nel seguente esempio.

Esempio:

```
; impostazione dei parametri di comunicazione
modbus.mode = 0
modbus.brate = 38400
...
modbus.OPENCOM
WAIT modbus.st_opencom
...
modbus.idcard = 8
modbus.addr = 600
modbus.num = 6
modbus.tabreg = 10
modbus.type = 3
modbus.SEND
WAIT modbus.st_sended
IF (modbus.err)
    CALL ErroreModbus
ENDIF
```

Documento generato automaticamente da **Qem Wiki** - <https://wiki.qem.it/>

Il contenuto wiki è costantemente aggiornato dal team di sviluppo, è quindi possibile che la versione online contenga informazioni più recenti di questo documento.