

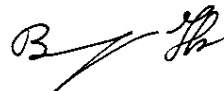



MANUALE UTENTE

Protocollo di comunicazione S.C.S. (in ASCII)

**CONVERTITORE VETTORIALE
PER MOTORI BRUSHLESS SINUSOIDALI
ALIMENTAZIONE CLASSE 400V (CVS)
ALIMENTAZIONE CLASSE 200V (CVS22)**

Valido per la release software **SW001_06**
Valido per la release software **SW007_xx**
Valido per la release software **SW035_xx.yy**

Mod. S04P01M05 Rev. 00 NT259_06	Rev. 06	Data: 09/03/2000	Pag. 1/17
Firme		Preparato da: Fabio Boninsegna 	Verificato da: Dario Sottocorna 

Sommario

CAPITOLO 1 : STRUTTURA DELLA RETE DI AZIONAMENTI	3
1.1 - Struttura fisica e stati della rete	3
1.2 - Struttura logica della rete	5
1.3 - Scambio delle informazioni tra Master e Slave	5
CAPITOLO 2 : STRUTTURA DEI MESSAGGI SCAMBIATI SULLA RETE	6
2.1 - Struttura delle frames	6
2.2 - Significato messaggi e campi delle frames standard	7
2.2.1 - Frame di tipo "A"	7
2.2.1.1 - Tabella significati dei codici frame tipo "A"	8
2.2.2 - Frame di tipo "B"	9
2.2.3 - Frame di tipo "C"	9
2.2.3.1 - Tabella significati dei codici frame tipo "C"	10
2.2.4 - Frame di tipo "D" (impostazione frame a pacchetti)	11
CAPITOLO 3 : ESECUZIONE D'OPERAZIONI E COMANDI	12
3.1 - Esecuzione comando "tipo azionamento"	12
3.2 - Scrittura di una variabile	13
3.3 - Lettura di una variabile	14
3.4 - Lettura di una variabile Input / Output (anche a bit)	15
3.5 - Lettura stato allarmi (anche a bit)	16
3.6 - Esecuzione di comandi diretti	17

Capitolo 1 : Struttura della rete di azionamenti

1.1 - Struttura fisica e stati della rete

Un singolo azionamento può comunicare in modo RS232, utilizzando questo protocollo, con una qualsiasi interfaccia seriale PC e/o PLC.

Considereremo invece il caso più generale d'utilizzo, in modo RS485, per una rete con più azionamenti.

Una rete di azionamenti ha una struttura a bus (di tipo *multi_drop*), in cui tutti i sistemi sono collegati in parallelo sulla stessa linea di trasmissione come illustrato in figura seguente: una utenza è il MASTER del sistema (host PC o supervisore) e le altre sono gli SLAVE (azionamenti).

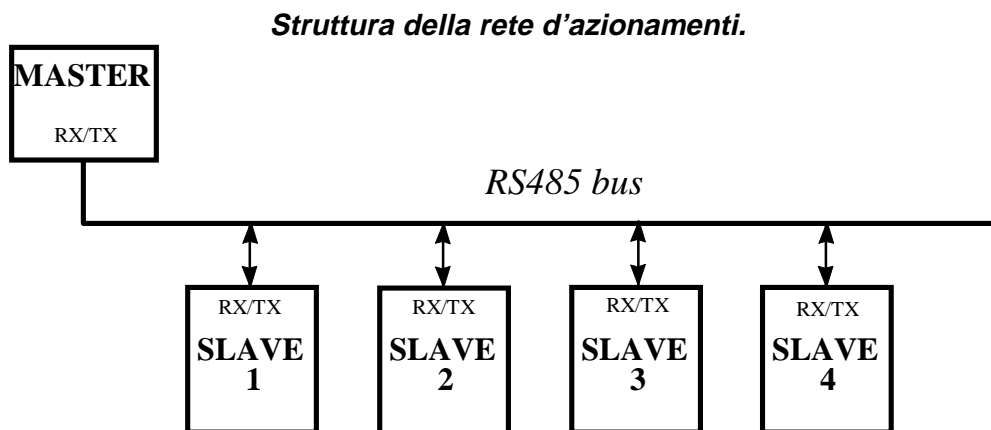
Il MASTER controlla il traffico dei dati sulla rete, inviando dei messaggi di interrogazione agli SLAVE e attendendo la loro risposta.

Inoltre il MASTER può inviare interrogazioni simultaneamente a tutti gli SLAVE (modo *broadcast*), ma in questo caso non riceve dagli SLAVE alcun tipo di risposta (per evitare conflitti sulla rete).

Ogni SLAVE è identificato da un indirizzo contenuto in PA00, che può variare da 00 a 99.

L'indirizzo 00 è riservato al MASTER per le interrogazioni di tipo *broadcast*.

Se ad uno SLAVE viene assegnato l'indirizzo 00, quest'ultimo ignorerà i tutti messaggi sulla linea seriale, anche quelli di tipo *broadcast*.

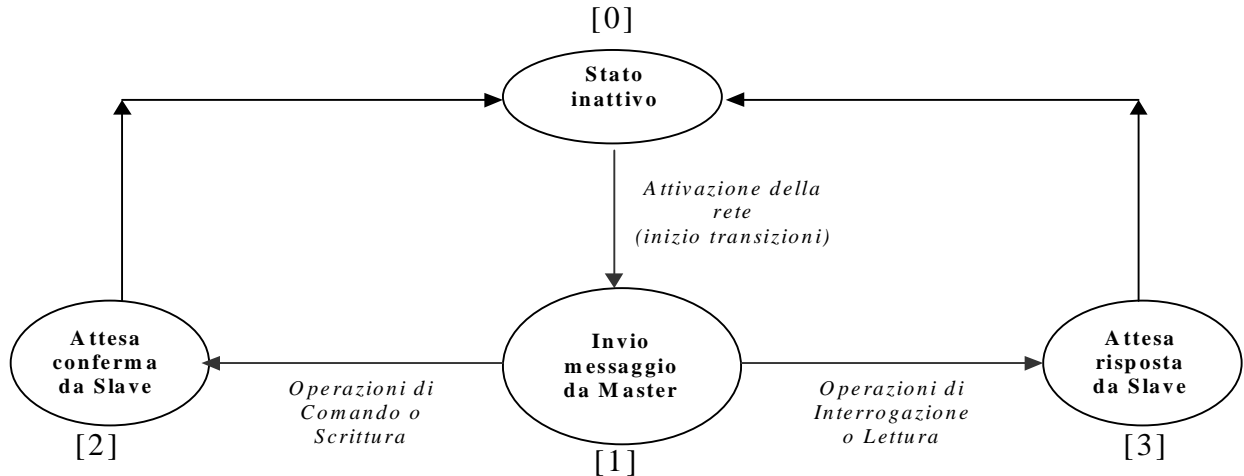


Il MASTER può inviare tre tipi d'interrogazione agli SLAVE:

- *Lettura*: il MASTER richiede un dato da uno SLAVE;
- *Scrittura*: il MASTER invia un dato ad uno SLAVE;
- *Comando*: il MASTER invia un comando diretto a uno SLAVE.

In modo *broadcast*, solamente le interrogazioni di Scrittura o Comando possono essere interpretate da tutti gli SLAVE, mentre quelle di Lettura sono ignorate.

Le interrogazioni di tipo *broadcast* non prevedono alcuna risposta da parte degli SLAVE.

Diagramma degli stati della rete.

La rete viene gestita secondo il diagramma degli stati rappresentato in figura precedente. Durante i quattro stati, il MASTER e gli SLAVE eseguono le seguenti operazioni:

0 - Stato inattivo: tutti gli SLAVE sono in ascolto sulla linea RS485 e attendono una frame. Nessuno SLAVE può arbitrariamente impegnare la rete se non è stato espressamente interrogato dal MASTER.

1 - Stato interrogazione: il MASTER invia un messaggio di interrogazione sulla linea RS485. Il messaggio di interrogazione può essere indirizzato ad uno SLAVE particolare o essere diretto a tutti gli SLAVE collegati alla rete (messaggio di tipo *broadcast*).

2 - Stato conferma: lo SLAVE che ha ricevuto un messaggio di tipo *comando* o *scrittura* risponde con un messaggio di conferma (che specifica se l'operazione è stata accettata oppure se esiste una condizione di errore). **Gli altri SLAVE attendono la fine della transazione**, cioè finché non è terminata la trasmissione del messaggio di risposta dello SLAVE interrogato.

3 - Stato risposta: lo SLAVE che ha ricevuto un messaggio di tipo *lettura* risponde inviando al MASTER il messaggio contenente il dato richiesto. **Gli altri SLAVE attendono la fine della transazione**, cioè finché non è terminata la trasmissione del messaggio di risposta dello SLAVE interrogato.

Ogni SLAVE risponde al MASTER solo quando riceve un interrogazione a lui destinata e ignora i messaggi inviati ad altri SLAVE.

In realtà ogni SLAVE riceve i messaggi di interrogazione inviati dal MASTER (che sono identificabili dal primo carattere del pacchetto), ne verifica il check-sum e poi controlla se l'indirizzo specificato nel messaggio corrisponde al proprio.

In caso d'errore di check-sum (BCC) o indirizzo non corrispondente, lo SLAVE non risponde ed attende l'invio da parte del MASTER della frame seriale successiva.

1.2 - Struttura logica della rete

I messaggi scambiati sulla rete tra il MASTER e gli SLAVE che supportano il protocollo SCS saranno organizzati secondo la struttura ISO/OSI [1].

Il protocollo descritto al precedente paragrafo definisce in modo esauriente i primi tre livelli ISO/OSI della rete:

Livello fisico:

- RS232 (SW14 = 1);
- RS485 (SW14 = 2) half-duplex;
- baudrate impostato in SW03;
- modalità asincrona;
- 10 bit (1 start, 8 data, no parità, 1 stop);

Livello di collegamento oppure modalità di sincronizzazione:

- la sincronizzazione è basata su un meccanismo del tipo *chiamata-risposta*, in cui per ogni messaggio inviato dal MASTER (escluso *broadcast* che non richiede risposta) si deve attendere una qualche risposta dallo SLAVE

Livello di rete:

- struttura e modalità di scambio dei messaggi.

I restanti 4 livelli definiscono l'utilizzo del campo DATI di ogni messaggio e saranno condensati in una unica specifica descritta al capitolo 2. Questa specifica, riassunta nei paragrafi seguenti, definirà le modalità d'interrogazione e comando degli SLAVE, secondo modalità simili a quelle di utilizzo di una semplice interfaccia utente strutturata a pagine in cui il campo **dati** consente di accedere alla voce desiderata.

1.3 - Scambio delle informazioni tra Master e Slave

Lo scambio d'informazioni tra MASTER e SLAVE avviene per mezzo del "campo dati" dei pacchetti. Il "campo dati" è una stringa di caratteri ASCII di lunghezza variabile, sino ad un massimo di 255.

Durante le operazioni di *Lettura* o *Scrittura* vengono scambiati tra MASTER e SLAVE i valori delle variabili interne degli azionamenti. L'informazione relativa viene codificata nella stringa del campo *dati* in base a quanto stabilito dalle caratteristiche delle variabili interne all'azionamento.

Poiché le informazioni scambiate durante le operazioni di *Lettura* o *Scrittura* sono di tipo numerico, la codifica del campo *dati* del pacchetto è analoga al modo di visualizzazione già utilizzato per il tastierino economico.

Esempio di dato contenuto in una variabile intera dell'azionamento (x = 1234):

- rappresentazione intera: 1234
- rappresentazione con una cifra decimale: 123.4 o 123,4
- rappresentazione esadecimale: 0x4D2

Durante la richiesta d'esecuzione comandi, il "campo dati" può contenere parametri numerici o messaggi. La codifica dei parametri e dei messaggi è definita per ogni tipo di comando.

Capitolo 2 : Struttura dei messaggi scambiati sulla rete

2.1 - Struttura delle frames

I messaggi scambiati sulla rete sono organizzati a *frames*, secondo diversi tipi di struttura dedicati ai diversi tipi di messaggi.

Le *frames* sono formate esclusivamente da caratteri ASCII-8 e sono costituite da un carattere di **start**, da un campo di **indirizzo** (2 caratteri), da un campo **comando** (4 caratteri), da un campo **dati o codice di errore o stato** (lunghezza variabile, da 0 a 255 caratteri), da un campo di **stop pacchetto** (EOT=04H) e da un campo di codice **errore BCC** (1 carattere).

Il pacchetto di risposta (da SLAVE a MASTER) non contiene il campo **comando**. È tuttavia mantenuto il campo indirizzo per preservare la consistenza dei dati.

Il codice di **errore BCC** è calcolato come somma modulo_256 del valore binario (codici ASCII) di tutti i caratteri comprendenti la frame, da quello di **start** a quello di **stop pacchetto** compreso.

Struttura della frame tipo "A".

carattere di start pacchetto	Indirizzo SLAVE		Codice COMANDO				Campo dati: caratteri ASCII (>31), di lunghezza variabile.				carattere di stop pacchetto	Codice di controllo	
ENQ= control E	0	1	P	A	3	4					EOT= control D	BCC	
	0	1	2	3	4	5	6	7	8	n	n+1	n+2	n+3

(a): Formato A = richiesta operazione da MASTER a SLAVE.

La **frame di tipo "A"** è inviata dal MASTER per iniziare una transazione sulla rete (scambio di messaggi con uno SLAVE) e contiene, oltre all'indirizzo dello SLAVE interrogato, un campo di **comando** e un campo di **dati** che possono servire allo SLAVE per eseguire il comando richiesto (per esempio l'impostazione di un parametro dell'azionamento).

Il campo **dati** può avere lunghezza variabile ed è terminato dal carattere EOT.

Queste frames sono usate per realizzare la sincronizzazione tra MASTER e SLAVE, cioè per determinare l'inizio di ogni transazione sulla rete.

Struttura della frame tipo "B".

carattere di start pacchetto	Indirizzo SLAVE		Campo dati: caratteri ASCII (>31), di lunghezza variabile.				carattere di stop pacchetto	Codice di controllo	
READ= control F	0	1					EOT= control D	BCC	
	0	1	2	3	4	n	n+1	n+2	n+3

(b): Formato B = risposta da SLAVE a MASTER.

La **frame di tipo "B"** è inviata dallo SLAVE al MASTER come risposta ad una richiesta di dati (interrogazione in lettura) e contiene, oltre al campo **indirizzo**, anche una stringa di lunghezza variabile (sino a 255 caratteri) contenente il dato richiesto e rappresentato in formato ASCII secondo la stessa modalità di visualizzazione utilizzata per il tastierino economico e secondo quanto stabilito dalle caratteristiche delle variabili stesse.

Questa frame viene usata dallo SLAVE anche come risposta ad un'operazione di scrittura, quando si eccedono i limiti impostati per la variabile modificata. Allora, il campo dati contiene l'indicazione del limite superato (inferiore o superiore) ed il valore del limite.

Struttura della frame tipo "C"

carattere di start pacchetto	Indirizzo SLAVE		Codice errore/stato: car. ASCII (>31), di lunghezza variabile.				carattere di stop pacchetto	Codice di controllo	
EOW= control G	0	1					EOT= control D	BCC	
0	1	2	3	4		n	n+1	n+2	n+3

(c): Formato C = risposta a TEST e conferme da SLAVE a MASTER.

La **frame di tipo "C"** è inviata dallo SLAVE al MASTER come risposta ad un comando o ad un'interrogazione in scrittura. Contiene, oltre al campo *indirizzo*, anche un codice d'errore (formato da due o tre caratteri). Con questo codice di errore lo SLAVE segnala al MASTER eventuali errori occorsi durante l'esecuzione del comando (per esempio l'impostazione di un parametro modificabile solo *off-line* con la regolazione attiva, la rappresentazione numerica incongruente con quanto stabilito dalle caratteristiche del parametro stesso, ecc.).

2.2 - Significato messaggi e campi delle frames standard

2.2.1 - Frame di tipo "A"

Le frame tipo "A" sono normalmente utilizzate dal Master per interrogare lo Slave.

- **Campo start [1 carattere]:** ENQ = CTRL+E = 0x05 Hex.
- **Campo indirizzo [2 caratteri]:** da 01a 99 (00 = comando broadcast).
- **Campo comando [4 caratteri]:** Il primo carattere specifica il tipo di operazione secondo la seguente tabella, mentre i restanti tre caratteri specificano informazioni aggiuntive (per esempio il numero di parametro su cui effettuare l'operazione):
- **Campo dati [da 0 a 255 caratteri]:** il contenuto del campo dati è sempre formato da caratteri ASCII e viene analizzato dall'azionamento che lo riceve, il quale risponde segnalando un errore se la stringa non contiene i dati nel formato richiesto per l'operazione. Il formato con cui i dati sono rappresentati è per mezzo di una stringa ASCII. Se vi sono decimali il separatore può essere sia "." che ",". Se vengono impostati più decimali di quelli definiti per tale grandezza, il dato viene arrotondato alla cifra più vicina con il numero di decimale definito (es 12.34 → 12.3; 12.36 → 12.4).
- **Campo stop [1 carattere]:** EOT = CTRL+D = 0x04 Hex.
- **Campo check-sum (BCC) [1 carattere]:** somma modulo_256 di tutti i caratteri presenti fino al campo stop compreso.

I comandi di memorizzazione e di calibrazione utilizzano le stesse funzioni che possono essere richiamate anche per mezzo del tastierino economico.

Il comando di TEST può essere usato per testare lo stato durante l'esecuzione di un comando di memorizzazione o calibrazione.

2.2.1.1 - Tabella significati dei codici frame tipo "A"

Primo carattere	Altri 3 caratteri	Letture e scrittura parametri standard
I	Nxx	Letture stato ingresso logico numero xx (0 ÷ 31).
I	A00	Letture complessiva ingressi logici (32 bit).
O	Uxx	Letture stato uscita logica numero xx (00 ÷ 31).
O	A00	Letture complessiva uscite logiche (32 bit).
A	0xx	Letture registro ALLARME (16bit) numero xx (00 ÷ 02).
A	Lxx	Letture allarme numero xx (00 ÷ 47).
M	xxx	Letture MISURA (grandezza interna) numero xxx.
S	xxx	Letture SWITCH numero xxx.
P	Axx	Letture PARAMETRO normale numero xx.
P	Xxx	Letture PARAMETRO asse elettrico numero xx.
B	xxx	Scrittura SWITCH numero xxx.
W	Axx	Scrittura PARAMETRO normale numero xx.
W	Xxx	Scrittura PARAMETRO asse elettrico numero xx.
Letture e scrittura parametri posizionate		
P	Zxx	Letture PARAMETRO posizionate numero xx.
Q	xxx	Letture QUOTA numero xxx in modalit� quote (SW31=0).
W	Zxx	Scrittura PARAMETRO posizionate numero xx.
Y	xxx	Scrittura QUOTA numero xxx in modalit� quote (SW31=0).
Comandi diretti standard		
E	ESV	Salva impostazioni dell'azionamento in EEPROM.
E	ERD	Ripristina impostazioni dell'azionamento da EEPROM.
E	PDF	Ripristina impostazioni default dell'azionamento da EPROM.
F	IOF	Calibrazione automatica offset di corrente.
F	EOF	Calibrazione automatica offset ingressi encoder.
F	ROF	Calibrazione automatica offset ingressi resolver.
F	1OF	Calibrazione automatica offset ingresso analogico IN1.
F	2OF	Calibrazione automatica offset ingresso analogico IN2.
F	3OF	Calibrazione automatica offset ingresso analogico IN3.
F	AOF	Calibrazione automatica offset tutti ingressi analogici.
F	1FP	Calibrazione automatica fondoscala + ingresso analogico IN1.
F	2FP	Calibrazione automatica fondoscala + ingresso analogico IN2.
F	3FP	Calibrazione automatica fondoscala + ingresso analogico IN3.
F	AFP	Calibrazione automatica fondoscala + tutti ingressi analogici.
F	1FN	Calibrazione automatica fondoscala - ingresso analogico IN1.
F	2FN	Calibrazione automatica fondoscala - ingresso analogico IN2.
F	3FN	Calibrazione automatica fondoscala - ingresso analogico IN3.
F	AFN	Calibrazione automatica fondoscala - tutti ingressi analogici.
F	STU	Calibrazione automatica anello di corrente (self-tuning).
F	RES	Centratura automatica resolver.
R	STA	Reset software degli allarmi.
T	EST	Test dello stato dell'azionamento in seguito a un comando.
T	YPE	Richiesta tipo azionamento (CVS, CVA, taglie, opzioni).
Comandi diretti posizionate		
Z	GOH	Start homing posizionate.
Z	STH	Stop homing posizionate.
Z	GOP	Start posizionamento.
Z	STP	Stop posizionamento.

2.2.2 - Frame di tipo "B"

La frame tipo "B" è la classica risposta dello Slave al Master alla richiesta di lettura di un parametro. Infatti, il campo dati contiene il valore corrente del parametro richiesto.

- **Campo start [1 carattere]:** READ = CTRL+F = 0x06 Hex;
- **Campo indirizzo [2 caratteri]:** da 01 a 99;
- **Campo dati [da 0 a 255 caratteri]:** il campo dati contiene una stringa che rappresenta un dato di tipo numerico (risposta a una lettura). Il campo dati può contenere anche il messaggio di errore relativo al superamento dei limiti impostabili per una variabile durante un operazione di scrittura. In questo caso il campo dati ha il seguente formato:

L+ xxxxx: superamento limite superiore, che vale xxxxx;
L- xxxxx: superamento limite inferiore, che vale xxxxx;

- **Campo stop [1 carattere]:** EOT = CTRL+D = 0x04 Hex.
- **Campo check-sum (BCC) [1 carattere]:** somma modulo_256 di tutti i caratteri presenti fino al campo stop compreso.

2.2.3 - Frame di tipo "C"

La frame tipo 'C' è la risposta dello Slave al Master durante e/o alla fine di comandi diretti oppure alla scrittura di un parametro.

- **Campo start [1 carattere]:** EOW = CTRL+G = 0x07 Hex;
- **Campo indirizzo [2 caratteri]:** da 01 a 99;
- **Campo dati [2, 3 o 9 caratteri]:** che codificano diversi tipi di messaggi in risposta a un comando o operazione di scrittura (vedi tabella seguente);
- **Campo stop [1 carattere]:** EOT = CTRL+D = 0x04 Hex.
- **Campo check-sum (BCC) [1 carattere]:** somma modulo_256 di tutti i caratteri presenti fino al campo stop compreso.

In caso di risposta ad un comando di **TEST**, il campo errore può avere lunghezza variabile ed è così codificato:

< comando precedente [4 caratteri] > : < stato [4 caratteri] >

Il campo "comando precedente" contiene l'ultimo comando eseguito e il campo "stato" contiene un indicatore che specifica lo stato durante l'esecuzione del comando:

- **BUSY** → comando in esecuzione
- **DONE** → comando eseguito
- **FAIL** → comando fallito: richiedere lo stato degli allarmi per avere più indicazioni sulle cause della mancata esecuzione (Rif. NT214 cap 7).

2.2.3.1 - Tabella significati dei codici frame tipo "C"

Codice	Significato
OK	Operazione di scrittura conclusa correttamente.
CA	Comando diretto accettato.
EC	Comando non riconosciuto o errore di sintassi.
ES	Comando non accettato per lo stato azionamento non corretto.
EU	Identificatore di campo comando non ammesso es. "PQ00".
E1	Variabile richiesta disabilitata .
E2	Tentativo di modifica variabile OFF-LINE nello stato di regolazione attiva.
E3	Variabile modificabile fornendo la Password Utente.
E4	Variabile modificabile fornendo la Password SCS.
E5	Modifica bloccata per tutte le variabili (SW00 = 1).
E6	Tentativo di modifica a parametro solo visualizzabile.
E7	Campo abilitazione del parametro non riconosciuto.
E8	Richiesto un numero d'indice di una variabile non esistente.
E9	Conflitto di comandi simultanei.
E10	Password Utente errata (vedi PA98).
E11	Password SCS errata (vedi PA99).
E12	Variabile ad uso esclusivo CVS.
E13	Variabile ad uso esclusivo CVA.
E14	Errore di formato nel campo dati.
E20	Comando non riconosciuto.
	Significato codici per posizionario
EZ0	Azionamento non in modo posizionario (SW9 diverso da 1).
EZ1	Tentativo di modifica valore quota minore del finecorsa SW.
EZ2	Tentativo di modifica valore quota maggiore del finecorsa SW.
EZ3	Quota attuale alla presenza di un finecorsa HW.
EZ4	Quota attuale alla presenza di un finecorsa SW.
EZ5	Asse in movimento.
EZ6	Asse già in STOP e fermo.
EZ7	Posizionamento in corso.
EZ8	Homing in corso.
EZ9	STOP asincrono posizionamento disabilitato (SW35=0).
EZ10	Tentativo di impostazione quote in modalità a pacchetti.
EZ11	Tentativo di impostazione pacchetti in modalità quote.

2.2.4 - Frame di tipo "D" (impostazione frame a pacchetti)

Il pacchetto è una frame complessa, dove è possibile leggere o scrivere con una unica comunicazione seriale più parametri contemporaneamente.

Attualmente esiste solo il **pacchetto tipo '1'** valido solo con opzione OI_SP per posizionatore.

La differenza rispetto alle frames già descritte (che comunque sono sempre attive) è l'aggiunta di un nuovo tipo di frame 'D' che viene gestita dal Master, mentre lo Slave risponde sempre con frame tipo 'B' e 'C' già descritte.

- **Campo start [1 carattere]:** ENQ = CTRL+E = 0x05 Hex.
- **Campo indirizzo [2 caratteri]:** da 01 a 99 (00 = comando *broadcast*).
- **Campo comando [4 caratteri]:** Il primo carattere specifica il tipo di operazione secondo la seguente tabella, mentre i restanti tre caratteri specificano il numero di pacchetto su cui effettuare l'operazione:

Primo carattere	Altri 3 caratteri	Definizione
1	xxx	Lettura o impostazione pacchetto numero xxx in modalità a pacchetti (SW31=1).

- **Campo dati [da 0 a 255 caratteri]:** il contenuto del campo dati è sempre formato da caratteri ASCII e viene analizzato dall'azionamento che lo riceve, il quale risponde segnalando un errore se la stringa non contiene i dati nel formato richiesto per l'operazione. Il formato con cui i dati sono rappresentati è per mezzo di una stringa ASCII. **Se vi sono decimali il separatore può essere sia "." che ",".** Se vengono impostati più decimali di quelli definiti per tale grandezza, il dato viene arrotondato alla cifra più vicina con il numero di decimale definito (es. 12.34 → 12.3; 12.36 → 12.4).

N.B. Nella modalità a pacchetti (SW31 = 1), se il campo dati è vuoto, il Master richiede semplicemente la lettura del pacchetto numero "xxx".

Esempio di struttura della frame tipo "D" in lettura pacchetto numero 23.

carattere di start pacchetto	Indirizzo SLAVE		Codice COMANDO				carattere di stop pacchetto	Codice di controllo
ENQ= control E	0	1	1	0	2	3	EOT= control D	BCC
0	1	2	3	4	5	6	7	8

(a): Formato D = richiesta lettura pacchetto da MASTER a SLAVE.

Mentre nel caso d'impostazione pacchetto, sempre con il comando '1xxx', il campo dati ha il seguente formato:

<valore quota>:<valore velocità>:<tipo rampa>:<valore accelerazione> :

Almeno uno di questi quattro campi deve esistere ed il numero del separatore di campo ":" deve essere sempre pari a quattro.

Esempio di struttura della frame tipo "D" in impostazione pacchetto numero 0.

carattere di start pacchetto	Indirizzo SLAVE		Codice COMANDO				Campo dati: caratteri ASCII (>31), di lunghezza variabile.												carattere di stop pacchetto	Codice di controllo
ENQ= control E	0	1	1	0	0	0	2	0	.	5	:	:	2	:	4	.	8	:	EOT= control D	BCC
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

(a): Formato D = richiesta di impostazione di pacchetto da MASTER a SLAVE.

La figura precedente illustra come impostare il pacchetto 0 con campo *valore quota* (20.5mm), nessun campo *valore velocità* (predefinita), campo *tipo rampa* (2=rampa S) e campo *valore accelerazione* (4.8 sec).

- **Campo stop [1 carattere]:** EOT = CTRL+D = 0x04 Hex.
- **Campo check-sum (BCC) [1 carattere]:** somma modulo_256 di tutti i caratteri presenti fino al campo stop compreso.

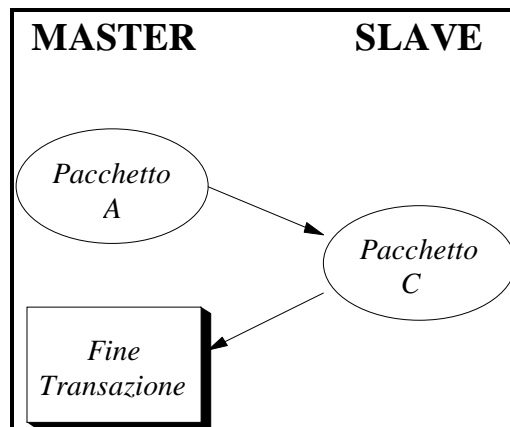
Capitolo 3 : Esecuzione d'operazioni e comandi

In questo capitolo sono descritte le operazioni eseguibili per mezzo del protocollo seriale, intese come sequenza di pacchetti scambiati tra MASTER e SLAVE sulla rete.

Le operazioni descritte qui di seguito presuppongono che il MASTER chiami uno SLAVE che comunque risponde. Se ciò non avviene (es. lo SLAVE chiamato non è collegato alla rete) allora, dopo aver atteso un certo tempo, il MASTER può iniziare una nuova transazione.

Gli altri SLAVE vengono quindi sbloccati non solo dai pacchetti B, C o D ma anche dalla trasmissione di pacchetti A (solo in caso di timeout dopo un'interrogazione).

3.1 - Esecuzione comando "tipo azionamento"



Questo comando viene utilizzato per conoscere le caratteristiche di sistema dell'azionamento.

La risposta, nel campo dati, ha la seguente codifica:

<tipo>:<versione>:<corrente>:<tensione>:<opzioni>

dove: *tipo* : azionamento tipo CVS o CVA

versione : numero e release software presente nell'azionamento

corrente : valore della taglia di corrente dell'azionamento

tensione : valore della taglia di tensione dell'azionamento (500Vdc o 1000Vdc)

opzioni : eventuali opzioni presenti

(attualmente: 0000 = nessun'opzione,
0003 = posizionatore modalità quote,
0007 = posizionatore modalità pacchetti)

➤ Esempio di richiesta "**tipo**" da MASTER per l'azionamento num. 20 (frame A):

(ASCII) = "[CHR\$(5)] [2 0] [T Y P E] [CHR\$(4)] [CHR\$(BCC)]"

(Hex) = 05 32 30 54 59 50 45 04 AD

➤ Risposta da azionamento SLAVE num. 20 (frame C):

(ASCII) = "[CHR\$(7)] [2 0] [C V S : S W 0 0 7 . 0 0 : 2 5 . 0

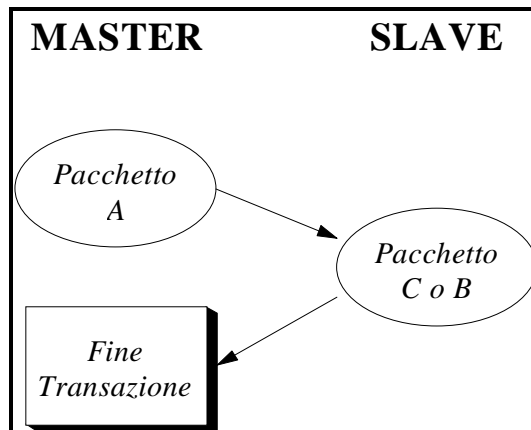
(Hex) = 07 32 30 43 56 53 3A 53 57 30 30 37 2E 30 30 3A 32 35 2E 30

: 1 0 0 0 . 0 : 0 0 0 3] [CHR\$(4)] [CHR\$(BCC)]"

3A 31 30 30 30 2E 30 3A 30 30 30 33 04 87

Cioè, driver CVS con software release 007.00, fondoscala di corrente 25A, fondoscala di tensione 1000.0V, presenza della opzione OI_SP in modo posizionatore a quote.

3.2 - Scrittura di una variabile



Messaggi d'errore utilizzati (pacchetto C): OK, E1...E9, ecc.
 Il pacchetto "B" viene utilizzato come risposta per segnalare il superamento dei limiti impostabili per la variabile richiesta.

➤ Esempio di "scrittura" da MASTER del parametro PA10 (tempo rampa accelerazione) con 50.7 secondi, per l'azionamento num. 06 (frame A):
 (ASCII) = "[CHR\$(5)] [0 6] [W A 1 0] [5 0 . 7] [CHR\$(4)] [CHR\$(BCC)]"
 (Hex) = 05 30 36 57 41 31 30 35 30 2E 37 04 32

➤ Risposta da azionamento SLAVE num. 06 (frame C):
 (ASCII) = "[CHR\$(7)] [0 6] [O K] [CHR\$(4)] [CHR\$(BCC)]"
 (Hex) = 07 30 36 4F 4B 04 0B
 Cioè, modifica PA10 corretta ed eseguita.

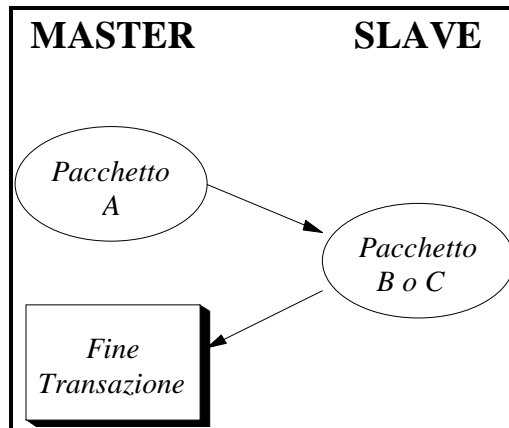
Note particolari sulla scrittura variabili.

Sono presenti nel CVS alcune variabili (vedi SW25, SW26, SW27) che devono essere impostate in esadecimale, antepoendo al valore Hex i caratteri '0x':

➤ Esempio di "scrittura" da MASTER dello switch SW26 (maschera 2 abilitazione allarmi) con il valore 580 Hex, sull'azionamento num. 03 (frame A):
 (ASCII) = "[CHR\$(5)] [0 3] [B 0 2 6] [0 x 5 8 0] [CHR\$(4)] [CHR\$(BCC)]"
 (Hex) = 05 30 33 42 30 32 36 30 78 35 38 30 04 8B

➤ Risposta da azionamento SLAVE num. 03 (frame C):
 (ASCII) = "[CHR\$(7)] [0 3] [E 4] [CHR\$(4)] [CHR\$(BCC)]"
 (Hex) = 07 30 33 45 34 04 E7
 Cioè, tentativo di modifica SW26 senza password SCS.

3.3 - Lettura di una variabile



N.B.: se lo SLAVE non è in grado di rispondere con un pacchetto B (es. si richiede una variabile non definita o non disponibile) allora invia direttamente un pacchetto C per terminare la transazione.

Se il MASTER rileva un errore sul dato ricevuto, può ripetere la transazione inviando un nuovo pacchetto A. La transazione termina comunque con il pacchetto B o C.

➤ Esempio di “lettura” da MASTER del parametro asse elettrico AX2 (numeratore rapporto) per azionamento num. 13 (frame A):
 (ASCII) = “[CHR\$(5)] [1 3] [P X 0 2] [CHR\$(4)] [CHR\$(BCC)]”
 (Hex) = 05 31 33 50 58 30 32 04 77

➤ Risposta da azionamento SLAVE num. 13 (frame B):
 (ASCII) = “[CHR\$(6)] [1 3] [5 0 0 0] [CHR\$(4)] [CHR\$(BCC)]”
 (Hex) = 06 31 33 35 30 30 30 04 33
 Cioè, AX2 contiene il valore 5000.

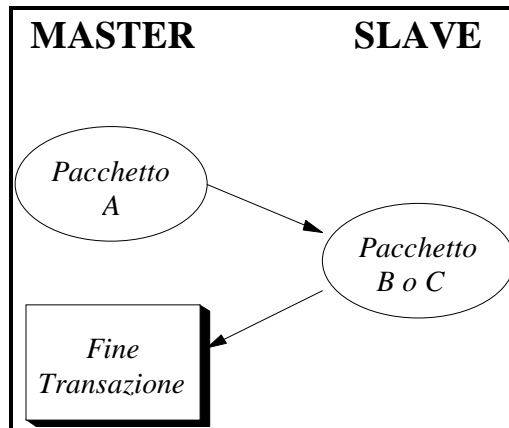
Note particolari sulla lettura di variabili.

Sono presenti nel CVS alcune variabili (vedi GR34, SW25, SW26, SW27) che sono gestite in esadecimale, infatti viene anteposto al valore Hex i caratteri ‘0x’:

➤ Esempio di “lettura” da MASTER della grandezza 34 sull’azionamento numero 69 (con opzione OI-SP) (frame A):
 (ASCII) = “[CHR\$(5)] [6 9] [M 0 3 4] [CHR\$(4)] [CHR\$(BCC)]”
 (Hex) = 05 36 39 4D 30 33 34 04 5C

➤ Risposta da azionamento SLAVE num. 69 (frame B):
 (ASCII) = “[CHR\$(6)] [6 9] [0 x 3] [CHR\$(4)] [CHR\$(BCC)]”
 (Hex) = 06 36 39 30 78 33 04 54
 Cioè, asse fermo ed in posizione sull’azionamento num. 69.

3.4 - Lettura di una variabile Input / Output (anche a bit)



N.B.: se lo SLAVE non è in grado di rispondere con un pacchetto B (es. si richiede un I/O non definito o non disponibile) allora invia direttamente un pacchetto C per terminare la transazione. Se il MASTER rileva un errore sul dato ricevuto, può ripetere la transazione inviando un nuovo pacchetto A. La transazione termina comunque con il pacchetto B o C.

➤ Esempio di “lettura” da MASTER di tutti gli input digitali sull’azionamento num. 21 (frame A):
 (ASCII) = “[CHR\$(5)] [2 1] [I A 0 0] [CHR\$(4)] [CHR\$(BCC)]”
 (Hex) = 05 32 31 49 41 30 30 04 56

➤ Risposta da azionamento SLAVE num. 21 (frame B):
 (ASCII) = “[CHR\$(6)] [2 1] [0 x 3 2 F 6 0 0] [CHR\$(4)] [CHR\$(BCC)]”
 (Hex) = 06 32 31 30 78 33 32 46 36 30 30 04 56
 Cioè, lo stato completo (32bit) dell’input digitale del CVS num. 21.

➤ Esempio di “lettura” da MASTER di tutti gli output digitali sull’azionamento num. 2 (frame A):
 (ASCII) = “[CHR\$(5)] [0 2] [O A 0 0] [CHR\$(4)] [CHR\$(BCC)]”
 (Hex) = 05 30 32 4F 41 30 30 04 5B

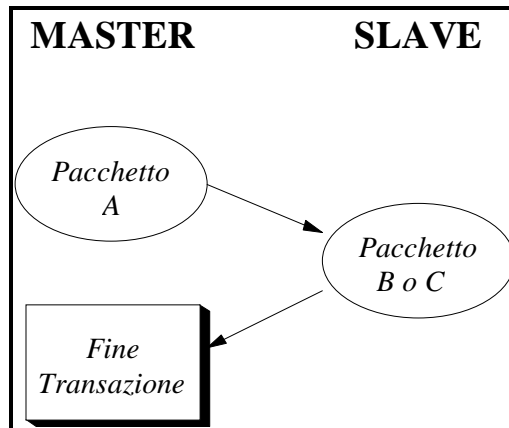
➤ Risposta da azionamento SLAVE num. 02 (frame B):
 (ASCII) = “[CHR\$(6)] [0 2] [0 x A B C] [CHR\$(4)] [CHR\$(BCC)]”
 (Hex) = 06 32 31 30 78 41 42 43 04 DA
 Cioè, lo stato completo (32bit) dell’output digitale del CVS numero 2.

➤ Esempio di “lettura” da MASTER dello stato dell’input digitale 13 sull’azionamento num. 21 (frame A):
 (ASCII) = “[CHR\$(5)] [2 1] [I N 1 3] [CHR\$(4)] [CHR\$(BCC)]”
 (Hex) = 05 32 31 49 4E 31 33 04 67

➤ Risposta da azionamento SLAVE num. 21 (frame B):
 (ASCII) = “[CHR\$(6)] [2 1] [O N] [CHR\$(4)] [CHR\$(BCC)]”
 (Hex) = 06 32 31 4F 4E 04 DA

Cioè, l’input digitale 13 del CVS num. 21 è attivo (**ON**). In caso contrario il campo dati conterrà **OFF**.

3.5 - Lettura stato allarmi (anche a bit)



N.B.: se lo SLAVE non è in grado di rispondere con un pacchetto B (es. si richiede un allarme non definito o non disponibile) allora invia direttamente un pacchetto C per terminare la transazione.

Se il MASTER rileva un errore sul dato ricevuto, può ripetere la transazione inviando un nuovo pacchetto A. La transazione termina comunque con il pacchetto B o C.

➤ Esempio di “lettura” da MASTER di tutti gli allarmi di tipo 2 (analogici) sull’azionamento num. 01 (frame A):
 (ASCII) = “[CHR\$(5)] [0 1] [A 0 0 2] [CHR\$(4)] [CHR\$(BCC)]”
 (Hex) = 05 30 31 41 30 30 32 04 3D

➤ Risposta da azionamento SLAVE num. 01 (frame B):
 (ASCII) = “[CHR\$(6)] [0 1] [0 x 0] [CHR\$(4)] [CHR\$(BCC)]”
 (Hex) = 06 30 31 30 78 30 04 43

Cioè, nessun allarme.

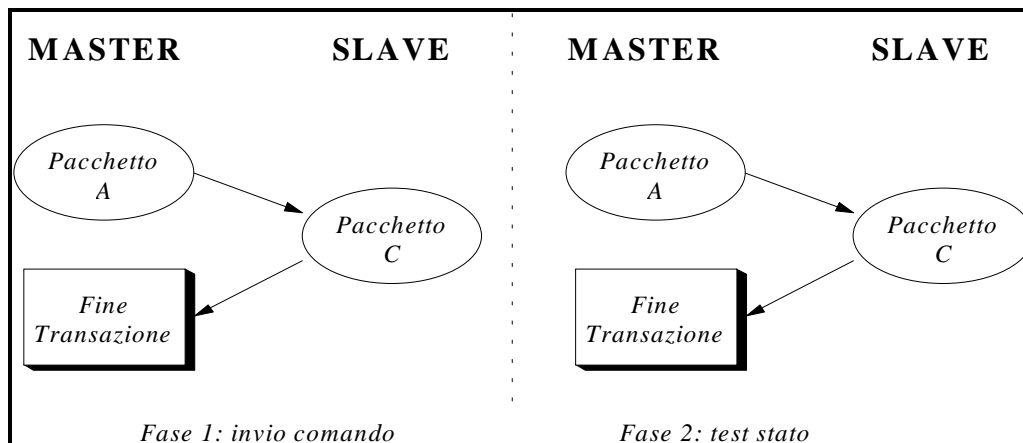
➤ Esempio di “lettura” da MASTER dell’allarme num. 4 (pastiglie termiche) sull’azionamento num. 03 (frame A):
 (ASCII) = “[CHR\$(5)] [0 3] [A L 0 4] [CHR\$(4)] [CHR\$(BCC)]”
 (Hex) = 05 30 33 41 4C 30 34 04 5D

➤ Risposta da azionamento SLAVE num. 03 (frame B):
 (ASCII) = “[CHR\$(6)] [0 3] [O F F] [CHR\$(4)] [CHR\$(BCC)]”
 (Hex) = 06 30 33 4F 46 46 04 48

Cioè, allarme 4 è disattivo (**OFF**). In caso contrario il campo dati conterrà **ON**.

Esiste anche, nel caso di lettura allarmi tipo ALxx, **una risposta del campo dati tipo ‘Act’**: questo significa che l’allarme è attivo ma non abilitato dalla propria maschera (vedi SW25, SW26, SW27).

3.6 - Esecuzione di comandi diretti



Lo SLAVE risponde subito con un pacchetto C, segnalando se il comando è stato accettato per le condizioni attuali di funzionamento. Se il comando è accettato, allora la sua esecuzione inizia immediatamente dopo l'invio del pacchetto.

Il MASTER può testare lo stato dello SLAVE con il comando TEST.

Messaggi d'errore utilizzati:

Risposta al comando: CA, ES, EC;

Risposta al test dello stato comando: (FAIL / DONE / BUSY).

N.B. Se esiste errore (stato "FAIL") si può, con richiesta di lettura registri di allarme, avere il dettaglio del tipo di mal funzionamento.

➤ Esempio di "comando diretto" da MASTER di calibrazione automatica PI corrente (self-tuning) sull'azionamento num. 05 (frame A):
 (ASCII) = "[CHR\$(5)] [0 5] [F S T U] [CHR\$(4)] [CHR\$(BCC)]"
 (Hex) = 05 30 35 46 53 54 55 04 B0

➤ Risposta da azionamento SLAVE num. 05 (frame C):
 (ASCII) = "[CHR\$(7)] [0 5] [C A] [CHR\$(4)] [CHR\$(BCC)]"
 (Hex) = 07 30 35 43 41 04 F4

Cioè, comando self-tuning accettato ed inizio immediato Calibrazione PI.

➤ Esempio di "comando diretto" da MASTER salvataggio parametri su EEPROM per azionamento num. 11 (frame A):
 (ASCII) = "[CHR\$(5)] [1 1] [E E S V] [CHR\$(4)] [CHR\$(BCC)]"
 (Hex) = 05 31 31 45 45 53 56 04 9E

➤ Risposta da azionamento SLAVE num. 11 (frame C):
 (ASCII) = "[CHR\$(7)] [1 1] [E S] [CHR\$(4)] [CHR\$(BCC)]"
 (Hex) = 07 31 31 45 53 04 05

Cioè, comando Save non accettato per stato azionamento non corretto (probabilmente OnLine).