

HSD

MANUALE PROTOCOLLO SERIALE RS485 per motori **SM137** e **SM140**

H5834D0007ITA
rev. 03
16/07/2010

Protocollo di comunicazione RS485



HSD S.p.A

registered office:
via Della Meccanica, 16
61122 Pesaro (PU) ITALY
loc. Chiusa di Ginestreto

factory headquarters:
piazzale Alfio De Simoni, sn
61122 Pesaro (PU) ITALY
tel.: (+39) 0721.205.211
fax: (+39) 0721.205.247
e-mail: supporthsd@hsd.it
web: www.hsd.it



Informazioni sulla pubblicazione

Copyright © 2010 HSD S.p.A.. Tutti i diritti riservati.

| Codice | Edizione | Revisione |
|---------------|----------|--------------|
| H5834D0007ITA | 1 | 1 (07, 2010) |

| Revisione | Redattore | Controllato | Approvato |
|-----------|-------------|-------------|-----------|
| 3 | A.Facondini | F.Toma | M.Venturi |

| Elenco degli aggiornamenti | | | |
|----------------------------|-----------------|-----------|--------------------|
| Revisione | Aggiunto | Eliminato | Modificato |
| 0 | Documento nuovo | | |
| 1 | | | Par 3.10 |
| 2 | | | Par 4.4, 4.5, 4.15 |
| 3 | | | Riferimenti |

Questo documento è stato realizzato da HSD esclusivamente per i propri clienti e contiene informazioni di proprietà riservata. Pertanto è vietata la riproduzione e/o divulgazione intera o parziale, in qualsiasi forma, senza l'autorizzazione scritta da HSD.

INDICE

| | | |
|----------|--|-----------|
| 1 | Installazione | 9 |
| 2 | Comunicazione seriale..... | 11 |
| 2.1 | Caratteri di controllo | 11 |
| 2.2 | Checksum del pacchetto trasmesso (CRC a 8 bit) | 11 |
| 2.3 | Lunghezza del pacchetto trasmesso | 11 |
| 2.4 | Indirizzo di nodo | 12 |
| 2.5 | Tempistiche | 12 |
| 3 | Comandi | 13 |
| 3.1 | Versioni del firmware | 13 |
| 3.2 | Uscita dallo stato di allarme | 13 |
| 3.3 | Motore fuori regolazione | 14 |
| 3.4 | Motore in regolazione | 14 |
| 3.5 | Stop | 15 |
| 3.6 | Assegnazione manuale della quota del motore (Azzeramento Manuale) | 16 |
| 3.7 | Assegnazione automatica della quota del motore (Azzeramento Automatico) | 17 |
| 3.8 | Movimento manuale con assegnamento della velocità (JOG) | 19 |
| 3.9 | Posizionamento alla quota assegnata (LINE) | 20 |
| 3.10 | Posizionamento alla quota assegnata con velocità assegnata (LINEVEL) | 21 |
| 3.11 | Richiesta della quota, della velocità e della coppia del motore | 23 |
| 4 | Comandi avanzati | 25 |
| 4.1 | Motore in EMERGENZA | 25 |
| 4.2 | Impostazione del valore di OVERRIDE | 25 |
| 4.3 | Lettura del valore di OVERRIDE | 26 |
| 4.4 | Modifica di un parametro dell'azionamento | 27 |
| 4.5 | Lettura di un parametro dell'azionamento | 29 |
| 4.6 | Salvataggio dei parametri | 30 |
| 4.7 | Taratura degli offset dei sensori di corrente | 31 |
| 4.8 | Azzeramento elettrico | 31 |
| 4.9 | Lettura dello stato di azzeramento | 32 |
| 4.10 | Lettura della distanza fra microinterruttore di azzeramento e tacca di zero dell'encoder | 33 |
| 4.11 | Ricezione della quota reale | 33 |
| 4.12 | Ricezione della quota teorica | 34 |
| 4.13 | Ricezione delle velocità teorica e reale | 34 |

| | | |
|----------|---|-----------|
| 4.14 | Ricezione dei valori di coppia teorica e reale | 35 |
| 4.15 | Letture dei messaggi di allarme o di Warning | 35 |
| 4.16 | Letture della versione software del firmware | 36 |
| 4.17 | Letture del tipo di motore e di bus di campo | 36 |
| 4.18 | Letture dello stato interno del motore | 37 |
| 4.19 | Campionamento di variabili | 38 |
| 4.20 | Campionamento variabili a frequenza assegnata | 38 |
| 4.21 | Letture dei dati campioni | 39 |
| 5 | Comandi obsoleti | 41 |
| 5.1 | Regolazione con attesa | 41 |
| 5.2 | Stop con attesa | 42 |
| 5.3 | Movimento manuale con assegnamento della velocità (Jog) | 42 |
| 5.4 | Modifica di un parametro dell'azionamento | 43 |
| 5.5 | Azzeramento elettrico con attesa | 43 |
| 5.6 | Campionamento di variabili | 44 |
| 6 | Parametri | 45 |
| 6.1 | Taratura dei parametri | 49 |
| 6.1.1 | Struttura del regolatore | 49 |
| 6.1.2 | Taratura della parte in feedforward del regolatore | 50 |
| 6.1.3 | Taratura della parte in retroazione del regolatore di velocità | 51 |
| 6.1.4 | Taratura della parte in feedforward del regolatore di posizione | 51 |
| 6.1.5 | Taratura della parte in feedforward del regolatore di posizione | 51 |
| A | Parametri di basso livello | 53 |
| B | Stati interni | 59 |
| C | Stato dell'azzeramento | 65 |
| D | Messaggi | 67 |
| D.1 | Messaggi di ALLARME | 67 |
| D.2 | Messaggi di WARNING | 70 |
| E | BitStato | 71 |
| F | Risposta a comando non valido | 73 |
| G | Ingressi digitali | 75 |

| | | |
|----------|--|-----------|
| G.1 | Introduzione | 75 |
| G.2 | Camme di extracorsa | 75 |
| G.3 | Microinterruttore di azzeramento | 75 |
| H | Interpolazione | 77 |
| I | ASSISTENZA..... | 79 |

1 Installazione

Per installare correttamente il motore SM137 o il motore SM140 occorre attuare le impostazioni descritte nel corrispondente Manuale Utente – Specifiche Tecniche e di Collegamento.

2 Comunicazione seriale

L'unità minima di trasmissione è costituita da un insieme di 10 bit:

- 1 bit di inizio sequenza (bit di start)
- 8 bit di dato
- 1 bit di fine sequenza (bit di stop)

| Start | Dato | | | | | | | | Stop |
|-------|------|--|--|--|--|--|--|-----|------|
| 0 | Lsb | | | | | | | Msb | 1 |

Il formato di trasmissione è il seguente:

1 start bit, 8 data bit, no parity, 1 stop bit

2.1 Caratteri di controllo

I motori SM137 e SM140 inviano comandi e richieste di dati in pacchetti composti da più caratteri. Ogni pacchetto inizia con il carattere di controllo **STX=0x02** (Inizio trasmissione) e termina con il carattere di controllo **ETX=0x1B** (Fine Trasmissione). I caratteri di controllo **STX, ETX, ESC** possono essere trasmessi come dati solo dopo essere stati trasformati in una sequenza di due caratteri come riportato nella seguente tabella:

| Caratteri da trasmettere | | Caratteri trasmessi | | |
|--------------------------|--------|---------------------|--------|--------------------------|
| STX | (0x02) | ESC | (0x1B) | 0xFF ⊕ STX (0xFD) |
| ETX | (0x03) | ESC | (0x1B) | 0xFF ⊕ ETX (0xFC) |
| ESC | (0x1B) | ESC | (0x1B) | 0xFF ⊕ ESC (0xE4) |

Il simbolo ⊕ indica l'operazione logica **XOR**.

2.2 Checksum del pacchetto trasmesso (CRC a 8 bit)

Ogni pacchetto contiene un carattere di checksum (**CRC**) che viene trasmesso prima del carattere di controllo **ETX**.

| | | | | | |
|------------|-------|-----|-------|------------|------------|
| STX | Dato1 | ... | DatoN | CRC | ETX |
|------------|-------|-----|-------|------------|------------|

Il carattere **CRC** viene calcolato nel seguente modo:

$$CRC = 0xFF \oplus Datum1 \oplus Datum2 \oplus L \oplus DatumN$$

Il simbolo ⊕ indica l'operazione logica **XOR**.

N.B.: il carattere di checksum **CRC** viene calcolato sui byte di dati del comando prima che siano modificati qualora contengano i caratteri **STX, ETX, ESC**. I caratteri di controllo devono "mascherare" anche il **CRC**.

2.3 Lunghezza del pacchetto trasmesso

Il numero massimo di byte di dati che può essere trasmesso è 68; tale numero non tiene in considerazione i caratteri di controllo necessari alla trasmissione.

2.4 Indirizzo di nodo

Il secondo byte di ogni comando rappresenta un indirizzo di nodo assegnabile mediante il dip-switch di configurazione del motore.

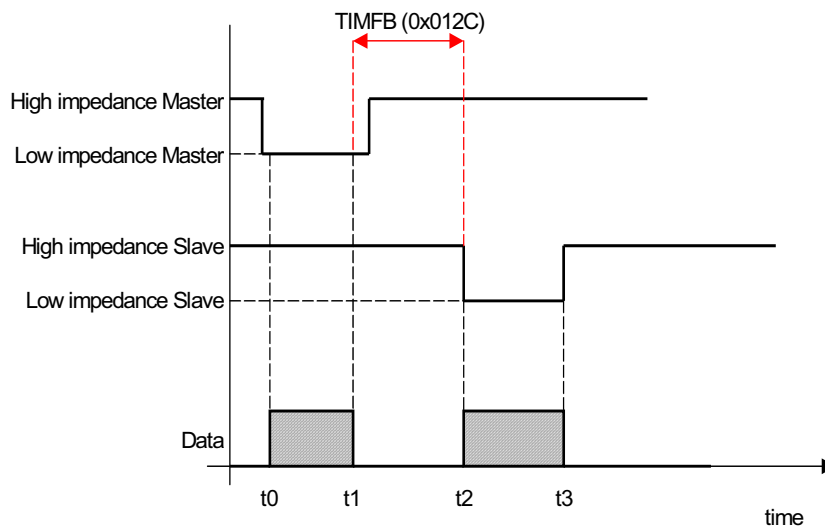
2.5 Tempistiche

La comunicazione con il motore avviene attraverso una linea seriale a 2 fili. Su questo tipo di connessione possono essere collegati più "enti"; in tal caso, un solo ente per volta può trasmettere mentre tutti, ad esclusione di chi trasmette, possono ricevere.

Ogni ente per poter trasmettere deve porre la linea di comunicazione in uno stato di "bassa impedenza" mentre ogni ente per poter ricevere deve porre la linea in uno stato di "alta impedenza". L'architettura è di tipo gerarchico e quindi sulla stessa linea è collegato un ente "master" (controllo numerico, personal computer, ...) e uno o più enti "slave" (i motori). È l'ente master a decidere con quale ente slave colloquiare.

La tempistica con cui gli enti master e slave pongono la linea negli stati di alta e bassa impedenza e quindi ricevono e trasmettono è riportata in Figura 1.

Figura 1 Tempistiche di innalzamento e abbassamento dell'impedenza della linea di comunicazione



Con riferimento alla Figura 1, si ha che:

- L'intervallo di tempo (t_2-t_1) è maggiore o uguale al tempo impostato nel parametro $TIMFB (0x12C)$
- Se l'ente master non trasmette di nuovo entro il tempo ($t_3 + TIMEOUTFB (0x12D)$), l'ente slave si pone in uno stato di allarme a meno che il valore del parametro $TIMEOUTFB$ non sia 0
- Il led del motore relativo alla comunicazione rimane acceso dall'istante t_0 all'istante t_3

3 Comandi

Nel pacchetto di risposta ai comandi sono sempre presenti 4 bit denominati BitStato il cui significato è riportato in Appendice E.

3.1 Versioni del firmware

Nelle successive sezioni, molti dei comandi descritti dipendono dalla versione del firmware ovvero dalla sigla di revisione dei Motori. Per distinguere una revisione C da una revisione B del motore SM137 è possibile controllare:

- I codice d'ordine
- la versione del firmware: se è minore o uguale a 110 il motore è una revisione B, se è maggiore o uguale a 110 è una revisione C

Il motore SM140 è stato commercializzato solo con versioni del firmware maggiori o uguali a 110.

3.2 Uscita dallo stato di allarme

N.B.: questo comando è utilizzabile sul motore SM140 e solo a partire dalla revisione C del motore SM137. Per motori SM137 di revisione precedente alla C è possibile utilizzare il comando "Motore fuori regolazione" (Sezione 3.3).

Il comando fa uscire il motore dallo stato di allarme.

| Byte | Comando | Risposta | |
|------|-----------------|-----------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdReset (0x9c) | CmdReset (0x9c) | |
| 05 | 0x00 | 0x00 | |
| 06 | 0x00 | 0x00 | |
| 07 | 0x00 | 0x00 | |
| 08 | CRC | CRC | |
| 09x | ETX (0x03) | ETX (0x03) | |

3 Comandi

3.3 Motore fuori regolazione

Il comando fa uscire il motore dallo stato di regolazione. Al di fuori dello stato di regolazione il motore non cerca di mantenere la posizione in cui è stato comandato.

| Byte | Comando | Risposta | |
|------|-----------------|-----------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | BitStato |
| 03 | 0x08 | 0x0 | |
| 04 | CmdNoReg (0x20) | CmdNoReg (0x20) | |
| 05 | 0x00 | 0x00 | |
| 06 | 0x00 | 0x00 | |
| 07 | 0x00 | 0x00 | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni:

- Il comando non ha effetto quando il motore sta effettuando:
 - l'azzeramento elettrico
 - la taratura degli offset dei sensori di corrente
- Il comando non viene accettato se:
 - il parametro **BIT_A** (sia veda l'Appendice **A**) ha il bit6 ad 1 e il motore è in allarme

3.4 Motore in regolazione

N.B.: questo comando è utilizzabile sul motore SM140 e solo a partire dalla revisione C del motore SM137.
Per motori SM137 di revisione precedente alla C è possibile utilizzare il comando "Regolazione con attesa" (Sezione 5.1).

Il comando porta il motore nello stato di regolazione. Nello stato di regolazione il motore cerca di mantenere la posizione in cui è stato comandato opponendosi ad eventuali carichi esterni.

| Byte | Comando | Risposta | |
|------|---------------|---------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | BitStato |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdReg (0xcc) | CmdReg (0xcc) | |
| 05 | 0x00 | 0x00 | |
| 06 | 0x00 | 0x00 | |
| 07 | 0x00 | 0x00 | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni:

Il comando ha effetto solo se il motore non è nello stato di regolazione.

Note

- Il motore è effettivamente nello stato di regolazione solo dopo che il valore del bit `MASKBITCOMANDO_SM` è 1 (si veda l'Appendice C)
- Se il motore non è nello stato di regolazione è garantito che, nella risposta al comando, il valore del bit `MASKBITCOMANDO_SM` è 0

3.5 Stop

N.B.: questo comando è utilizzabile sul motore SM140 e solo a partire dalla revisione C del motore SM137.
 Per motori SM137 di revisione precedente alla C è possibile utilizzare il comando "Stop con attesa" (Sezione 5.2).

Il comando ferma il motore eseguendo una opportuna rampa di decelerazione ed interrompendo l'eventuale movimento in corso.

N.B.: il motore è dichiarato fermo (stato AXSTOP) quando la velocità teorica è 0 e quella reale è inferiore ai 12 rpm in valore assoluto.

| Byte | Comando | Risposta | |
|------|----------------|----------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdHold (0xbc) | CmdHold (0xbc) | |
| 05 | 0x00 | 0x00 | |
| 06 | 0x00 | 0x00 | |
| 07 | 0x00 | 0x00 | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni:

Il comando ha effetto solo se il motore è in movimento.

Note

- Quando il motore è "fermo" il valore del bit `MASKBITCOMANDO_SM` è 1 (si veda l'Appendice C)
- Se il motore non è fermo, è garantito che, nella risposta al comando, il valore del bit `MASKBITCOMANDO_SM` è 0

3.6 Assegnazione manuale della quota del motore (Azzeramento Manuale)

Il comando consente di assegnare la quota del motore con un valore prestabilito Q. Tale procedura è spesso indicata con i termini "azzeramento manuale".
Alla conclusione del comando, Q diviene la quota corrente del motore.

L'azzeramento non avviene in corrispondenza della tacca di zero dell'encoder ma è immediato.

In seguito a un azzeramento manuale lo stato di azzeramento del motore viene posto a 1 (parametro **AZZMAN**).

Il valore di azzeramento Q è un numero a 32 bit con segno indicato come segue:

| bit 31-24 | bit 23-16 | bit 15-8 | bit 7-0 |
|-----------|-----------|----------|---------|
| Q3 | Q2 | Q1 | Q0 |

| Byte | Comando | Risposta | |
|------|----------------|----------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | CmdMazz (0x01) | 0x0 | BitStato |
| 04 | Q1 | CmdMazz (0x01) | |
| 05 | Q0 | Q0 | |
| 06 | Q3 | Q3 | |
| 07 | Q2 | Q2 | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni:

Il comando ha effetto solo se il motore è nello stato di allarme, fuori regolazione o in regolazione.

Calcolo della quota di azzeramento

La quota di azzeramento Q deve essere espressa in conteggi encoder come nell'esempio seguente.

Esempio: calcolo dei componenti Q0, Q1, Q2, Q3 nel caso in cui siano:

- 6766.8176 mm la quota di azzeramento Q
- 0.32 mm/giro il passo meccanico del motore
- 800 conteggi/giro lo sviluppo encoder

$$Q = 6780,8465[mm] = \frac{6780,8465}{0,32} \times 800 = 16952116,25[Cnt] = 0 \times 0102AB34[Cnt]$$

| Q3 | Q2 | Q1 | Q0 |
|------|------|------|------|
| 0x01 | 0x02 | 0xAB | 0x34 |

3.7 Assegnazione automatica della quota del motore (Azzeramento Automatico)

Il comando consente di assegnare la quota del motore con il valore contenuto nel parametro `ORIG_AZZ`. Tale procedura è spesso indicata con i termini “azzeramento automatico”. Il valore contenuto nel parametro `ORIG_AZZ` diviene la quota corrente del motore in corrispondenza della tacca di zero dell’encoder.

È possibile effettuare l’azzeramento automatico in diversi modi in base al valore contenuto nel parametro `TIPOAZZ`.

TIPOAZZ = 0

Per poter effettuare l’azzeramento automatico è necessario che il motore stia effettuando un “movimento manuale” (Jog).

La quota di azzeramento viene assegnata alla posizione reale in cui si trova il motore nell’istante in cui, per la prima volta dopo che è stato inviato il comando di azzeramento automatico, viene rilevata la tacca di zero dell’encoder.

Lo stato di azzeramento, prima della rilevazione della tacca di 0 dell’encoder, è `SEARCHINGTACCA` (`0x0006`); il bit di stato `MASKBITCOMANDO_SM` ha valore 0.

Dopo la rilevazione della tacca di 0 dell’encoder lo stato di azzeramento diviene `AZZAUTO` (`0x0003`) e il bit di stato di `MASKBITCOMANDO_SM` assume il valore 1.

TIPOAZZ = 1

ATTENZIONE: Questo tipo di azzeramento automatico è attivabile solo per i motori di tipo SM140.

Il motore, che deve essere nello stato di regolazione (`AXSTOP`), utilizza solo il microinterruttore di azzeramento.

Le fasi dell’azzeramento sono le seguenti:

- il motore, ricevuto il comando di azzeramento, parte alla ricerca del microinterruttore di azzeramento alla velocità impostata nel parametro `VMAXAZZ`. Il segno del parametro individua il verso di movimento del motore.

Lo stato dell’azzeramento è `SEARCHINGMICRO` (codice `0x0002`); il bit `MASKBITCOMANDO_SM` viene posto a 0.

- il motore incontra il microinterruttore di azzeramento e lo oltrepassa (il livello logico del terzo ingresso è alto); si ferma e riparte a velocità `VAZZOUTMIC`.

Lo stato di azzeramento diviene `LEAVINGMICRO` (codice `0x0004`)

- il motore rilascia il microinterruttore di azzeramento (il livello logico del terzo ingresso diviene 0) la quota di azzeramento viene assegnata e il motore conclude l’azzeramento.

Lo stato di azzeramento diviene `AZZAUTO` (codice `0x0003`); Il bit di stato `ASKBITCOMANDO_SM` viene posto a 1

TIPOAZZ = 2

ATTENZIONE: Questo tipo di azzeramento automatico è attivabile solo per i motori di tipo SM140.

Il motore, che deve essere nello stato di regolazione (`AXSTOP`), utilizza sia il microinterruttore di azzeramento che la tacca di zero dell’encoder.

3 Comandi

Le fasi di azzeramento sono le seguenti:

- il motore, ricevuto il comando di azzeramento, parte alla ricerca del microinterruttore di azzeramento alla velocità impostata nel parametro **VMAXAZZ**. Il segno del parametro individua il verso di movimento del motore.

Lo stato dell'azzeramento è **SEARCHINGMICRO** (codice **0x0002**); il bit **MASKBITCOMANDO_SM** viene posto a 0.

- il motore incontra il microinterruttore di azzeramento e lo oltrepassa (il livello logico del terzo ingresso è 1); si ferma e riparte a velocità **VAZZOUTMIC**.

Lo stato di azzeramento diviene **LEAVINGMICRO** (codice **0x0004**)

- il motore rilascia il microinterruttore di azzeramento (il livello logico del terzo ingresso diventa basso) e si pone alla ricerca della tacca di zero dell'encoder mantenendo la stessa velocità.

Lo stato di azzeramento diviene **SEARCHINGTACCA** (codice **0x0006**).

- appena il motore incontra la tacca di zero dell'encoder, la quota di azzeramento viene assegnata e il motore conclude l'azzeramento.

Lo stato di azzeramento diviene **AZZAUTO** (codice **0x0003**); Il bit di stato **MASKBITCOMANDO_SM** viene posto a 1.

Solo per questo tipo di azzeramento può essere significativo il comando **CmdGetDistMicroZero** che ritorna la distanza tra il microinterruttore di azzeramento e la tacca di zero dell'encoder del motore, relativa all'ultimo azzeramento effettuato.

Se l'azzeramento automatico viene interrotto, lo stato di azzeramento del motore diviene **NOAZZ** (codice **0x0000**).

| Byte | Comando | Risposta | |
|------|---------------|---------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdAzz (0x38) | CmdAzz (0x38) | |
| 05 | 0x00 | 0x00 | |
| 06 | 0x00 | 0x00 | |
| 07 | 0x00 | 0x00 | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni

Quando **TIPOAZZ=0** il comando ha effetto solo se il motore sta effettuando un movimento manuale (Jog). Per gli altri tipi di azzeramento il comando ha effetto solo se il motore è nello stato **AXSTOP**.

Note

La risposta al comando è immediata. Per sapere quando il motore ha concluso l'azzeramento automatico, effettuando una fermata e tornando in regolazione, occorre monitorare il bit di stato **MASKBITCOMANDO_SM** (Appendice C).

Dopo l'invio del comando, se il motore non ha terminato l'azzeramento, viene garantito che, nella risposta al comando, il valore del bit di stato **MASKBITCOMANDO_SM** è 0.

I possibili stati di azzeramento sono riportati in Appendice C.

3.8 Movimento manuale con assegnamento della velocità (JOG)

N.B.: questo comando è utilizzabile sul motore SM140 e solo a partire dalla revisione C del motore SM137.

Il comando consente di attivare il movimento manuale del motore assegnandone la velocità con un valore prestabilito v . Tale procedura è spesso indicata con il termine “Jog”.

Il valore di velocità v è un numero a 16 bit con segno, indicato come segue:

| bit 15-8 | bit 7-0 |
|----------|---------|
| VH | VL |

Il segno di v indica il verso di rotazione del motore:

- il segno “+” indica una rotazione verso quote crescenti
- il segno “-” indica una rotazione verso quote decrescenti.

L'unità di misura della velocità è espressa in giri al minuto.

| Byte | Comando | Risposta |
|------|----------------|----------------|
| 01 | STX (0x02) | STX (0x02) |
| 02 | Nodo | Nodo |
| 03 | 0x08 | 0x0 BitStato |
| 04 | CmdJogN (0xe0) | CmdJogN (0xe0) |
| 05 | 0x00 | 0x00 |
| 06 | VH | VH |
| 07 | VL | VL |
| 08 | CRC | CRC |
| 09 | ETX (0x03) | ETX (0x03) |

Condizioni

Il comando ha effetto solo se il motore è nello stato di regolazione oppure sta effettuando un movimento manuale con assegnamento della velocità (Jog).

Esempio: calcolo dei valori VH e VL per attivare un movimento del motore con una velocità di 185 mm/min, essendo il passo meccanico del motore pari a 0.32 mm/giro.

$$(-185) \left((mm)/(min) \Leftrightarrow \frac{-185}{0,32} \right) - 578[rpm] = 0 \times FDBE[rpm]$$

| VH | VL |
|------|------|
| 0xFD | 0xBE |

3 Comandi

3.9 Posizionamento alla quota assegnata (LINE)

Il comando consente di attivare un posizionamento del motore alla quota assegnata Q espressa in conteggi.

Il valore di posizionamento Q è un numero a 32 bit indicato come segue:

| bit 31-24 | bit 23-16 | bit 15-8 | bit 7-0 |
|-----------|-----------|----------|---------|
| Q3 | Q2 | Q1 | Q0 |

| Byte | Comando | Risposta | |
|------|----------------|----------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | CmdTraj (0x02) | 0x0 | BitStato |
| 04 | Q1 | CmdTraj (0x02) | |
| 05 | Q0 | Q0 | |
| 06 | Q3 | Q3 | |
| 07 | Q2 | Q2 | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni

- Il comando ha effetto solo se il motore è in regolazione ed è stato azzerato in modo manuale o in modo automatico
- La variazione della quota impostabile con il comando `CmdTraj` è al massimo di $0x3FFFFFF=67108863$ conteggi encoder, in valore assoluto. Se la variazione di quota impostata è superiore viene segnalato il messaggio di WARNING $0x400B$ (ALMOVTOOLONG), per maggiori dettagli si veda tabella a pag. 70.
Esempio: se la quota attuale è 1000, il movimento quotato deve essere compreso tra -67107863 e 67109863

Note

- la risposta al comando è immediata. Per sapere quando il motore ha concluso il posizionamento, tornando nello stato di regolazione, occorre monitorare il bit di stato `MASKBITCOMANDO_SM`
- nella risposta al comando viene garantito che, se il motore non ha ancora raggiunto la quota obiettivo, il bit di stato `MASKBITCOMANDO_SM` è 0
- il motore ha terminato il posizionamento se la quota reale ha raggiunto la quota obiettivo teorica con la precisione richiesta dai parametri impostati

Calcolo della quota di posizionamento.

La quota di posizionamento Q deve essere espressa in conteggi encoder come nell'esempio seguente.

Esempio: calcolo dei componenti Q0, Q1, Q2, Q3 nel caso in cui siano: 1150.75 mm la quota di posizionamento Q, 0.32 mm/giro il passo meccanico del motore e 500 conteggi/giro lo sviluppo encoder.

$$Q = 1150,75[mm] = \frac{1150,75}{0,32} \times 500 = 1798046,875[Cnt] = 0 \times 01B6F9E[Cnt]$$

| Q3 | Q2 | Q1 | Q0 |
|------|------|------|------|
| 0x01 | 0x1B | 0x6F | 0x9E |

3.10 Posizionamento alla quota assegnata con velocità assegnata (LINEVEL)

N.B.: questo comando è utilizzabile sul motore SM140 e solo a partire dalla revisione C del motore SM137.

Il comando consente di attivare un posizionamento del motore alla quota desiderata Q , espressa in conteggi, con velocità assegnata non negativa V , espressa in rpm.

Il valore di posizionamento Q è un numero a 32 bit indicato come segue:

| bit 31-24 | bit 23-16 | bit 15-8 | bit 7-0 |
|-----------|-----------|----------|---------|
| Q3 | Q2 | Q1 | Q0 |

Il valore di velocità V è un numero a 16 bit indicato come segue:

| bit 15-8 | bit 7-0 |
|----------|---------|
| VH | VL |

| Byte | Comando | Risposta |
|------|------------------|------------------|
| 01 | STX (0x02) | STX (0x02) |
| 02 | Nodo | Nodo |
| 03 | 0x00 | 0x0 BitStato |
| 04 | CmdTrajVel(0xc8) | CmdTrajVel(0xc8) |
| 05 | 0x00 | 0x00 |
| 06 | Q1 | Q1 |
| 07 | Q0 | Q0 |
| 08 | Q3 | 0x00 |
| 09 | Q2 | CRC |
| 10 | VH | ETX |
| 11 | VL | |
| 12 | CRC | |
| 13 | ETX (0x03) | |

Condizioni

- il comando ha effetto solo se il motore è in regolazione o in fase di esecuzione di un movimento quotato con velocità assegnata
- nei casi in cui ci siano dei limiti cinematici per i quali il movimento comandato non è effettuabile (ad esempio la quota obiettivo inviata è troppo vicina alla quota corrente per consentire un'adeguata rampa di decelerazione) la risposta al comando è di tipo **CmdNACK**
- non è consentito al motore di invertire la velocità rispetto al movimento precedente. Ad esempio: supponiamo che il motore si trovi ad una quota $Q_i=0.0$ mm e venga comandato un movimento alla quota obiettivo $Q_f=500.0$ mm. Supponiamo inoltre che nel momento in cui il motore arriva alla quota intermedia $Q_t=400.0$ mm sia comandato un movimento alla nuova quota obiettivo $Q_{f1}=100.0$ mm. Dato che il movimento verso Q_{f1} comporta una inversione di velocità, tale movimento non può essere effettuato e la risposta al comando è di tipo **CmdNACK**
- Il motore deve essere già stato azzerato in modo manuale o in modo automatico prima dell'invio del comando **CmdTrajVel**
- La variazione della quota impostabile con il comando **CmdTraj** è al massimo di **0x3FFFFFF=67108863** conteggi encoder, in valore assoluto. Se la variazione di quota impostata è superiore viene segnalato il messaggio di WARNING **0x400B (ALMOVTOOLONG)**, per maggiori dettagli si veda tabella a pag. 70.
Esempio: se la quota attuale è 1000, il movimento quotato deve essere compreso tra -67107863 e 67109863

Note

- la risposta al comando è immediata. Per sapere quando il motore ha concluso il posizionamento tornando nello stato di regolazione occorre monitorare il bit di stato **MASKBITCOMANDO_SM**
- nella risposta al comando viene garantito che, se il motore non ha raggiunto la quota obiettivo, il bit di stato **MASKBITCOMANDO_SM** è 0
- il motore ha terminato il posizionamento se la quota reale ha raggiunto la quota obiettivo teorica con la precisione richiesta dai parametri impostati

Calcolo della quota obiettivo e della velocità di posizionamento.

Negli esempi seguenti sono calcolate: la quota obiettivo in conteggi encoder e la velocità di posizionamento in giri/min.

Esempio: calcolo dei componenti Q0, Q1, Q2, Q3, essendo:

- 1150.75 mm la quota di posizionamento Q
- 0.32 mm/giro il passo meccanico del motore
- 500 conteggi/giro lo sviluppo encoder

$$Q = 1150,75[mm] = \frac{1150,75}{0,32} \times 500 = 1798046,875[cnt] = 0 \times 01B6F9E[cnt]$$

| Q3 | Q2 | Q1 | Q0 |
|------|------|------|------|
| 0x01 | 0x1B | 0x6F | 0x9E |

Esempio: calcolo dei valori VH e VL per una velocità di 150 mm/min, essendo il passo meccanico del motore pari a 0.32 mm/giro.

$$150[\text{mm}/\text{mn}] \Leftrightarrow \frac{150}{0,32} = 468,75(\text{rpm}) = 0 \times 01D4[\text{rpm}]$$

| VH | VL |
|------|------|
| 0x01 | 0xD4 |

3.11 Richiesta della quota, della velocità e della coppia del motore

Mediante questo comando è possibile richiedere al motore:

- la quota reale
- la quota teorica
- la velocità reale e la velocità teorica
- la coppia reale e la coppia teorica

| Byte | Comando | Risposta | |
|------|------------|------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | 0x00 | D1 | |
| 05 | CRC | D0 | |
| 06 | ETX | D3 | |
| 07 | | D2 | |
| 08 | | CRC | |
| 09 | | ETX (0x03) | |

Condizioni

Il comando ha effetto in qualsiasi condizione.

Note

- il significato dei dati ritornati dipende dall'ultimo comando richiesto, come riportato nella seguente tabella
- all'accensione del motore, come ultimo comando richiesto, viene considerato `CmdGetPos`

| Ultimo comando richiesto | Descrizione | Valore | Unità di misura |
|--------------------------|--|--------------------------|-----------------|
| CmdGetPos | È ritornata la quota reale del motore | Qr = D3-D2-D1-D0 | conteggi |
| CmdGetPosT | È ritornata la quota teorica del motore | Qt = D3-D2-D1-D0 | conteggi |
| CmdGetVel | Sono ritornate la velocità teorica e quella reale del motore | Vr = D1-D0 Vt = D3-D2 | Q15 |
| CmdGetTor | Sono ritornate la coppia teorica e quella reale del motore | Tr = D1-D0 Tt = D3-D2 | Q15 |

Esempio 1: supponendo che l'ultimo comando richiesto sia CmdGetPosT, il passo meccanico del motore sia 0.32 mm/giro, lo sviluppo encoder 800 cnt/giro e il motore restituisca la quota teorica mediante i seguenti valori:

| D3 | D2 | D1 | D0 |
|------|------|------|------|
| 0x09 | 0xC1 | 0x10 | 0xAB |

la quota teorica in mm è ricavata nel seguente modo:

$$Q = 0 \times 09C110AB[cnt] = 163647659[cnt] = \frac{163647659}{800} \times 0,32[mm] = 65459,0636[mm]$$

Esempio 2: supponendo che l'ultimo comando richiamato sia CmdGetVel, il passo meccanico del motore sia 0.32 mm/giro, lo sviluppo encoder 800 cnt/giro e il motore restituisca la velocità teorica e la velocità reale mediante i seguenti valori:

| D3 | D2 | D1 | D0 |
|------|------|------|------|
| 0x09 | 0xC1 | 0x10 | 0xAB |

le velocità teorica e reale sono ricavate nel seguente modo:

$$Vt = 0 \times 09C1[Q15] = 2497[Q15] = \frac{2497}{2^{15}} \times 8000 \times 0,32[mm/min] = 195,0781[mm/min]$$

$$Vr = 0 \times 10AB[Q15] = 4267[Q15] = \frac{4267}{2^{15}} \times 8000 \times 0,32[mm/min] = 333,3594[mm/min]$$

4 Comandi avanzati

4.1 Motore in EMERGENZA

Porta il motore nello stato di EMERGENZA. Lo stato di emergenza è simile allo stato di NON REGOLAZIONE tranne per il fatto che dallo stato di emergenza non è possibile ritornare direttamente nello stato di regolazione.

Per riportare il motore nello stato di regolazione occorre riportare il motore nello stato di non regolazione.

| Byte | Comando | Risposta | |
|------|-----------------|-----------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdEmerg (0x90) | CmdEmerg (0x90) | |
| 05 | 0x00 | 0x00 | |
| 06 | 0x00 | 0x00 | |
| 07 | 0x00 | 0x00 | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni

Il comando ha effetto in qualsiasi condizione.

4.2 Impostazione del valore di OVERRIDE

N.B.: questo comando è utilizzabile sul motore SM140 e solo a partire dalla revisione C del motore SM137.

Mediante questo comando è possibile variare percentualmente la velocità di un movimento del motore. Il parametro inviato è la percentuale di velocità del motore e può assumere valori compresi fra 0 e 200.

| Byte | Comando | Risposta | |
|------|--------------------|----------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdSetOverr (0xd0) | CmdSetOverr (0xd0) | |
| 05 | 0x00 | 0x00 | |
| 06 | OH | Parte alta override | |
| 07 | OL | Parte bassa override | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

4 Comandi avanzati

Condizioni

- il comando ha effetto in qualsiasi momento; se un movimento è già in esecuzione, il motore effettuerà una rampa per raggiungere la nuova velocità

Esempio: se si vuole ridurre del 10% la velocità dei movimenti è sufficiente inviare al motore un valore di override pari a 90.

90 [%] 0x005A [%]

| OH | OL |
|------|------|
| 0x00 | 0x5A |

4.3 Lettura del valore di OVERRIDE

N.B.: questo comando è utilizzabile sul motore SM140 e solo a partire dalla revisione C del motore SM137.

Mediante questo comando è possibile leggere il valore di override corrente del motore.

| Byte | Comando | Risposta | |
|------|--------------------|----------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdGetOverr (0xd8) | CmdGetOverr (0xd8) | |
| 05 | 0x00 | 0x00 | |
| 06 | 0x00 | Parte alta override | |
| 07 | 0x00 | Parte bassa override | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni

Il comando può essere inviato in qualsiasi momento.

4.4 Modifica di un parametro dell'azionamento

N.B.: questo comando è utilizzabile sul motore SM140 e solo a partire dalla revisione C del motore SM137.

Mediante questo comando è possibile modificare i parametri dell'azionamento. La codifica dei parametri modificabili è riportata nella Sezione 6.

Occorre distinguere l'invio di un parametro a 16 bit dall'invio di un parametro a 32 bit.

Invio di parametri a 16 bit

Indichiamo con **PS0** e **PS1** la parte bassa e la parte alta del valore da assegnare al parametro a 16 bit PS, rispettivamente, come riportato nella seguente tabella.

| bit 15-8 | bit 7-0 |
|----------|---------|
| PS 1 | PS 0 |

Il comando da inviare è il seguente:

| Byte | Comando | Risposta | |
|------|---|------------------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdChgParN (0xb8) | CmdChgParN (0xb8) | |
| 05 | 0x00 | 0x00 | |
| 06 | Parte alta codice parametro | Parte alta codice parametro | |
| 07 | Parte bassa codice parametro | Parte bassa codice parametro | |
| 08 | Parte alta valore parametro(PS ₁) | CRC | |
| 09 | Parte alta valore parametro(PS ₀) | ETX (0x03) | |
| 10 | CRC | | |
| 11 | ETX (0x03) | | |

Invio di parametri a 32 bit

Indichiamo con **PL0**, **PL1**, **PL2** e **PL3** i byte che contengono il valore da assegnare al parametro a 32 bit PL, come riportato nella seguente tabella.

| bit 31-24 | bit 23-16 | bit 15-8 | bit 7-0 |
|-----------|-----------|----------|---------|
| PL 3 | PL 2 | PL 1 | PL 0 |

4 Comandi avanzati

Il comando da inviare è il seguente:

| Byte | Comando | Risposta | |
|------|------------------------------|------------------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdChgParN (0xb8) | CmdChgParN (0xb8) | |
| 05 | 0x00 | 0x00 | |
| 06 | Parte alta codice parametro | Parte alta codice parametro | |
| 07 | Parte bassa codice parametro | Parte bassa codice parametro | |
| 08 | PL ₁ | CRC | |
| 09 | PL ₀ | ETX (0x03) | |
| 10 | PL ₃ | | |
| 11 | PL ₂ | | |
| 12 | CRC | | |
| 13 | ETX (0x03) | | |

Invio contemporaneo di più parametri

È possibile assegnare i valori di più parametri contemporaneamente. Ad esempio: supponiamo di voler modificare un parametro a 16 bit e 2 parametri a 32 bit; in tal caso occorre inviare il seguente comando:

| Byte | Comando | Risposta | |
|------|--|------------------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdChgParN (0xb8) | CmdChgParN (0xb8) | |
| 05 | 0x00 | 0x00 | |
| 06 | Parte alta codice parametro | Parte alta codice parametro | |
| 07 | Parte bassa codice parametro | Parte bassa codice parametro | |
| 08 | Parte alta valore parametro (PS ₁) | CRC | |
| 09 | Parte alta valore parametro (PS ₀) | ETX (0x03) | |
| 10 | Parte alta codice parametro | | |
| 11 | Parte bassa codice parametro | | |
| 12 | PL ₁ | | |
| 13 | PL ₀ | | |
| 14 | PL ₃ | | |
| 15 | PL ₂ | | |
| 16 | Parte alta codice parametro | | |
| 17 | Parte bassa codice parametro | | |
| 18 | PL ₂ | | |
| 19 | PL ₀ | | |
| 20 | PL ₃ | | |
| 21 | PL ₂ | | |
| 22 | CRC | | |
| 23 | ETX (0x03) | | |

Condizioni

Il comando ha effetto solo se il motore è nello stato di allarme, fuori regolazione o in regolazione. Il limite sul numero di parametri che si possono inviare è costituito dalla lunghezza massima del pacchetto che è 68 byte.

4.5 Lettura di un parametro dell'azionamento

N.B.: questo comando è utilizzabile sul motore SM140 e solo a partire dalla revisione C del motore SM137.

Mediante questo comando è possibile leggere i parametri dell'azionamento. La codifica dei parametri che è possibile leggere è riportata nella Sezione 6.

Occorre distinguere la lettura di un parametro a 16 bit da quella di un parametro a 32 bit.

Letture di parametri a 16 bit

Indichiamo rispettivamente con **PS0** e **PS1** la parte bassa e la parte alta del valore del parametro a 16 bit PS che si vuole leggere, come riportato nella seguente tabella.

| bit 15-8 | bit 7-0 |
|----------|---------|
| PS 1 | PS 0 |

Il comando da inviare è il seguente:

| Byte | Comando | Risposta | |
|------|------------------------------|--|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdGetParN (0xc0) | CmdGetParN (0xc0) | |
| 05 | 0x00 | 0x00 | |
| 06 | Parte alta codice parametro | Parte alta codice parametro | |
| 07 | Parte bassa codice parametro | Parte bassa codice parametro | |
| 08 | 0x00 | Parte alta valore parametro (PS ₁) | |
| 09 | 0x00 | Parte alta valore parametro (PS ₀) | |
| 10 | CRC | CRC | |
| 11 | ETX (0x03) | ETX (0x03) | |

Letture di parametri a 32 bit

Indichiamo con **PL0**, **PL1**, **PL2** e **PL3** i byte che contengono il valore da leggere relativo al parametro a 32 bit PL, come riportato nella seguente tabella.

| bit 31-24 | bit 23-16 | bit 15-8 | bit 7-0 |
|-----------|-----------|----------|---------|
| PL 3 | PL 2 | PL 1 | PL 0 |

4 Comandi avanzati

Il comando da inviare è il seguente:

| Byte | Comando | Risposta | |
|------|------------------------------|------------------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdGetParN (0xc0) | CmdGetParN (0xc0) | |
| 05 | Parte alta codice parametro | Parte alta codice parametro | |
| 06 | Parte bassa codice parametro | Parte bassa codice parametro | |
| | 0x00 | PL ₁ | |
| 07 | 0x00 | PL ₀ | |
| 08 | 0x00 | PL ₃ | |
| 09 | 0x00 | PL ₂ | |
| 10 | CRC | CRC | |
| 11 | ETX (0x03) | ETX (0x03) | |

Condizioni

Il comando può essere inviato in qualsiasi momento.

4.6 Salvataggio dei parametri

N.B.: questo comando è utilizzabile sul motore SM140 e solo a partire dalla revisione C del motore SM137.

Il comando attua il salvataggio dei parametri del motore sulla flash memory. Alla successiva riaccensione il motore riparte con i parametri precedentemente salvati.

| Byte | Comando | Risposta | |
|------|---------------------|---------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdSaveParFl (0xe8) | CmdSaveParFl (0xe8) | |
| 05 | 0x00 | 0x00 | |
| 06 | 0x00 | 0x00 | |
| 07 | 0x00 | 0x00 | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni

N.B.: sulla flash memory è possibile attuare un numero finito di scritture. Per questo motivo è opportuno non utilizzare questo comando troppo frequentemente.

- il comando può essere inviato solo se il motore è nello stato di allarme

4.7 Taratura degli offset dei sensori di corrente

Il comando attua la taratura degli offset dei sensori di corrente. Questa operazione viene effettuata automaticamente dal motore all'accensione.

| Byte | Comando | Risposta | |
|------|------------------|------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdAdcOff (0x28) | CmdAdcOff (0x28) | |
| 05 | 0x00 | 0x00 | |
| 06 | 0x00 | 0x00 | |
| 07 | 0x00 | 0x00 | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni

Il comando ha effetto solo se il motore è in allarme.

Note

La risposta al comando è data solo al termine della scrittura.

4.8 Azzeramento elettrico

N.B.: questo comando è utilizzabile sul motore SM140 e solo a partire dalla revisione C del motore SM137.

Il comando azzerla la posizione elettrica del rotore. Questa operazione viene effettuata automaticamente dal motore la prima volta che viene portato in regolazione.

Vi sono diversi tipi di azzeramento: si consiglia di utilizzare sempre il tipo 0 essendo l'unico che non implica un movimento del motore.

| Byte | Comando | Risposta | |
|------|-------------------------|-------------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdAzzEl (0xc4) | CmdAzzEl (0xc4) | |
| 05 | 0x00 | 0x00 | |
| 06 | 0x00 | 0x00 | |
| 07 | Tipo azzeramento (0x00) | Tipo azzeramento (0x00) | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni

Il comando ha effetto solo se il motore è fuori regolazione.

4 Comandi avanzati

Note

- la risposta al comando è immediata.

Per sapere quando il motore ha terminato l'azzeramento elettrico occorre monitorare il bit di stato `MASKBITCOMANDO_SM`. Solo dopo che il motore ha terminato l'azzeramento elettrico il valore del bit `MASKBITCPMANDO_SM` diventa 1

4.9 Lettura dello stato di azzeramento

N.B.: questo comando è utilizzabile sul motore SM140 e solo a partire dalla revisione C del motore SM137.

Il comando serve per leggere lo stato dell'azzeramento del motore. Il valore ritornato è compreso fra quelli riportati nella seguente tabella:

| | |
|--|---|
| <code>NOAZZ (0x0000)</code> : | il motore non è azzerato |
| <code>AZZMAN (0x0001)</code> : | sul motore è stato attuato un azzeramento manuale |
| <code>AZZAUTO (0x0003)</code> : | sul motore è stato attuato un azzeramento automatico |
| <code>SEARCHINGMICRO (0x0002)</code> : | il motore sta effettuando l'azzeramento automatico; in particolare è in atto la fase di ricerca del microinterruttore di azzeramento |
| <code>LEAVINGMICRO (0x0004)</code> : | il motore sta effettuando l'azzeramento automatico: in particolare è in atto la fase di rilascio del microinterruttore di azzeramento |
| <code>SEARCHINGTACCA (0x0006)</code> : | il motore sta effettuando l'azzeramento automatico: in particolare è in atto la fase di ricerca della tacca di zero dell'encoder |

| Byte | Comando | Risposta |
|------|-----------------------------------|-----------------------------------|
| 01 | <code>STX (0x02)</code> | <code>STX (0x02)</code> |
| 02 | Nodo | Nodo |
| 03 | <code>0x08</code> | <code>0x0</code> BitStato |
| 04 | <code>CmdGetStatAzz (0x64)</code> | <code>CmdGetStatAzz (0x64)</code> |
| 05 | <code>0x00</code> | <code>0x00</code> |
| 06 | <code>0x00</code> | Parte alta stato azzeramento |
| 07 | <code>0x00</code> | Parte bassa stato azzeramento |
| 08 | CRC | CRC |
| 09 | <code>ETX (0x03)</code> | <code>ETX (0x03)</code> |

Condizioni

Il comando può essere inviato in qualsiasi momento.

4.10 Lettura della distanza fra microinterruttore di azzeramento e tacca di zero dell'encoder

N.B.: questo comando è utilizzabile solamente sul motore SM140

Il comando consente di rilevare la distanza tra la tacca di zero dell'encoder e il microinterruttore di azzeramento. La distanza D è espressa in conteggi encoder ed è ritornata su 4 byte:

| bit 31-24 | bit 23-16 | bit 15-8 | bit 7-0 |
|-----------|-----------|----------|---------|
| D3 | D 2 | D 1 | D 0 |

| Byte | Comando | Risposta |
|------|----------------------------|----------------------------|
| 01 | STX (0x02) | STX (0x02) |
| 02 | Nodo | Nodo |
| 03 | 0x08 | 0x0 BitStato |
| 04 | CmdGetDistMicroZero (0x5c) | CmdGetDistMicroZero (0x5c) |
| 05 | 0x00 | 0x00 |
| 06 | 0x00 | D1 |
| 07 | 0x00 | D0 |
| 08 | CRC | D3 |
| 09 | ETX (0x03) | D2 |
| 10 | | CRC |
| 11 | | ETX (0x03) |

Condizioni

- questo comando può essere inviato in qualsiasi momento
- l'uso di questo comando è utile solo nel caso in cui sia stato effettuato un azzeramento automatico di tipo 2 (TIPOAZZ=2); azzeramento con microinterruttore e tacca di zero)

4.11 Ricezione della quota reale

Il comando impone al motore di rispondere, ad ogni invio del comando nullo, con la quota reale dell'asse. L'unità di misura del valore inviato è il numero di conteggi.

| Byte | Comando | Risposta |
|------|------------------|------------------|
| 01 | STX (0x02) | STX (0x02) |
| 02 | Nodo | Nodo |
| 03 | 0x08 | 0x0 BitStato |
| 04 | CmdGetPos (0x68) | CmdGetPos (0x68) |
| 05 | 0x00 | 0x00 |
| 06 | 0x00 | 0x00 |
| 07 | 0x00 | 0x00 |
| 08 | CRC | CRC |
| 09 | ETX (0x03) | ETX (0x03) |

Condizioni

Il comando può essere inviato in qualsiasi condizione.

4 Comandi avanzati

4.12 Ricezione della quota teorica

Il comando impone al motore di rispondere, ad ogni invio del comando nullo, con la quota teorica dell'asse. L'unità di misura del valore ritornato è il numero di conteggi.

| Byte | Comando | Risposta | |
|------|-------------------|-------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdGetPosT (0x98) | CmdGetPosT (0x98) | |
| 05 | 0x00 | 0x00 | |
| 06 | 0x00 | 0x00 | |
| 07 | 0x00 | 0x00 | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

4.13 Ricezione delle velocità teorica e reale

Il comando impone al motore di rispondere, ad ogni invio del comando nullo, con i valori di velocità teorica e reale. L'unità di misura dei valori inviati è [Q15] (si veda l'Appendice D).

| Byte | Comando | Risposta | |
|------|------------------|------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdGetVel (0x70) | CmdGetVel (0x70) | |
| 05 | 0x00 | 0x00 | |
| 06 | 0x00 | 0x00 | |
| 07 | 0x00 | 0x00 | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni

Il comando può essere inviato in qualsiasi condizione.

4.14 Ricezione dei valori di coppia teorica e reale

Il comando impone al motore di rispondere, ad ogni invio del comando nullo, con i valori di coppia teorica e reale. L'unità di misura dei valori inviati è [Q15] (si veda l'Appendice D).

| Byte | Comando | Risposta | |
|------|------------------|------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdGetTor (0x78) | CmdGetTor (0x78) | |
| 05 | 0x00 | 0x00 | |
| 06 | 0x00 | 0x00 | |
| 07 | 0x00 | 0x00 | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni

Il comando può essere inviato in qualsiasi condizione.

4.15 Lettura dei messaggi di allarme o di Warning

Il comando consente la lettura del messaggio (di allarme o di warning) presente sull'azionamento. Per la codifica dei messaggi si veda l'Appendice D.

Una volta letto, il messaggio viene cancellato.

| Byte | Comando | Risposta | |
|------|--------------------|------------------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdGetAlarm (0x60) | CmdGetAlarm (0x60) | |
| 05 | 0x00 | 0x00 | |
| 06 | 0x00 | Parte alta codice messaggio | |
| 07 | 0x00 | Parte bassa codice messaggio | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni

Il comando può essere inviato in qualsiasi condizione.

4 Comandi avanzati

4.16 Lettura della versione software del firmware

Il comando consente di leggere la versione del firmware del motore.

| Byte | Comando | Risposta | |
|------|------------------|----------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdGetVer (0x80) | CmdGetVer (0x80) | |
| 05 | 0x00 | 0x00 | |
| 06 | 0x00 | Parte alta versione | |
| 07 | 0x00 | Parte bassa versione | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni

Il comando può essere inviato in qualsiasi condizione.

Note

I dati ritornati nei byte **Parte alta versione** e **Parte bassa versione** sono in formato esadecimale pertanto devono essere convertiti in decimale.

Esempio: supponiamo che vi siano i dati 0x00 e 0x67 nei byte **Parte alta versione** e **Parte bassa versione** rispettivamente. In tal caso la versione del firmware del motore è 0x0067=103.

4.17 Lettura del tipo di motore e di bus di campo

N.B.: questo comando è utilizzabile sul motore SM140 e solo a partire dalla revisione C del motore SM137.

Il comando consente la lettura del protocollo di comunicazione del motore.

| Byte | Comando | Risposta | |
|------|-------------------|-----------------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdGetType (0xa4) | CmdGetType (0xa4) | |
| 05 | 0x00 | 0x00 | |
| 06 | 0x00 | Parte alta tipo protocollo | |
| 07 | 0x00 | Parte bassa tipo protocollo | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni

Il comando può essere inviato in qualsiasi condizione.

Risposta

Il significato del valore ritornato è riportato nella seguente tabella:

| Byte | Comando | Protocollo di comunicazione |
|------|---------|-----------------------------|
| 0x00 | SM137 | ENET-X |
| 0x01 | SM137 | RS-485 |
| 0x02 | SM137 | CAN |
| | | |
| 0x10 | SM140 | ENET-X |
| 0x11 | SM140 | RS-485 |
| 0x12 | SM140 | CAN |

4.18 Lettura dello stato interno del motore

Il comando permette di leggere lo stato interno del motore.

Per la codifica degli stati interni del motore si faccia riferimento all'Appendice B.

| Byte | Comando | Risposta |
|------|---------------------|--------------------------|
| 01 | STX (0x02) | STX (0x02) |
| 02 | Nodo | Nodo |
| 03 | 0x08 | 0x0 BitStato |
| 04 | CmdGetSmStat (0xA8) | CmdGetSmStat (0xA8) |
| 05 | 0x00 | 0x00 |
| 06 | 0x00 | Parte alta codice stato |
| 07 | 0x00 | Parte bassa codice stato |
| 08 | CRC | CRC |
| 09 | ETX (0x03) | ETX (0x03) |

Condizioni

Il comando può essere inviato in qualsiasi condizione.

4.19 Campionamento di variabili

N.B.: questo comando è utilizzabile sul motore SM140 e solo a partire dalla revisione C del motore SM137.

Il comando abilita il campionamento di due variabili interne al firmware.

| Byte | Comando | Risposta | |
|------|---------------------|---------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdSampleVar (0xb4) | CmdSampleVar (0xb4) | |
| 05 | 0x00 | 0x00 | |
| 06 | Codice variabile 1 | Codice variabile 1 | |
| 07 | Codice variabile 2 | Codice variabile 2 | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni

Il comando può essere inviato in qualsiasi condizione.
La codifica delle variabili è riportata nelle appendici .

4.20 Campionamento variabili a frequenza assegnata

N.B.: questo comando è utilizzabile sul motore SM140 e solo a partire dalla revisione C del motore SM137.

Il comando abilita il campionamento di due variabili interne al firmware; la frequenza di campionamento è assegnata come parametro.

| Byte | Comando | Risposta | |
|------|---------------------------------|------------------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdSampleMem (0xac) | CmdSampleMem (0xac) | |
| 05 | Parte alta indirizzo var. 1 | Parte alta indirizzo var. 1 | |
| 06 | Parte bassa indirizzo var. 1 | Parte bassa indirizzo var. 1 | |
| 07 | Parte alta indirizzo var. 2 | Parte alta indirizzo var. 2 | |
| 08 | Parte bassa indirizzo var. 2 | CRC | |
| 09 | Parte alta t. di campionamento | ETX (0x03) | |
| 10 | Parte bassa t. di campionamento | | |
| 11 | CRC | | |
| 12 | ETX (0x03) | | |

Condizioni

Il comando può essere inviato in qualsiasi condizione.

4.21 Lettura dei dati campioni

Scarica i valori campionati dal motore.

| Byte | Comando | Risposta | |
|---------|-------------------|-------------------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdGetSamp (0xA0) | CmdGetSamp (0xA0) | |
| 05 | 0x00 | 0x00 | |
| 06 | 0x00 | Parte alta numero campioni | |
| 07 | 0x00 | Parte bassa numero campioni | |
| 08 | CRC | Parte alta campione 1 var. 1 | |
| 09 | ETX (0x03) | Parte bassa campione 1 var. 1 | |
| 10 | | Parte alta campione 1 var. 2 | |
| 11 | | Parte bassa campione 1 var. 2 | |
| 12 | | Parte alta campione 2 var. 1 | |
| 13 | | Parte bassa campione 2 var. 1 | |
| 14 | | Parte alta campione 2 var. 2 | |
| 15 | | Parte bassa campione 2 var. 2 | |
| ... | | ... | |
| 8+4 N | | Parte alta campione N var. 1 | |
| 8+4 N+1 | | Parte bassa campione N var. 1 | |
| 8+4 N+2 | | Parte alta campione N var. 2 | |
| 8+4 N+3 | | Parte bassa campione N var. 2 | |
| 8+4 N+4 | | 0xFF | |
| 8+4 N+5 | | 0xFF | |
| 8+4 N+6 | | CRC | |
| 8+4 N+7 | | ETX (0x03) | |

Condizioni

Il comando può essere inviato in qualsiasi condizione.

Il valore massimo di N è costituito dalla lunghezza massima del buffer (si veda la Sezione 2.3 di questo manuale).

5 Comandi obsoleti

ATTENZIONE: I comandi descritti nelle sezioni successive sono obsoleti ma mantenuti per compatibilità con le versioni precedenti del software.

SE NE SCONSIGLIA L'USO!

5.1 Regolazione con attesa

Il comando porta il motore nello stato di regolazione.

Nello stato di regolazione il motore cerca di mantenere la posizione in cui è stato comandato opponendosi ad eventuali carichi esterni.

| Byte | Comando | Risposta | |
|------|-------------------|-------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdRegWait (0x18) | CmdRegWait (0x18) | |
| 05 | 0x00 | 0x00 | |
| 06 | 0x00 | 0x00 | |
| 07 | 0x00 | 0x00 | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni

Il comando ha effetto solo se il motore è fuori regolazione.

Note

Quando il motore viene portato in regolazione per la prima volta dopo l'accensione, la risposta a questo comando viene inviata con un ritardo di circa 0.1 sec. Questo ritardo è determinato dall'azzeramento elettrico del rotore; tale azzeramento deve avvenire prima che il motore sia portato nello stato di regolazione.

Per evitare questa attesa è necessario utilizzare il comando descritto nella Sezione 3.4 (codice 0xcc).

5 Comandi obsoleti

5.2 Stop con attesa

Il comando ferma il motore eseguendo una opportuna rampa di decelerazione ed interrompendo l'eventuale movimento in corso.

| Byte | Comando | Risposta | |
|------|--------------------|--------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdHoldWait (0x50) | CmdHoldWait (0x50) | |
| 05 | 0x00 | 0x00 | |
| 06 | 0x00 | 0x00 | |
| 07 | 0x00 | 0x00 | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni

Il comando ha effetto solo se il motore è in movimento.

Note

La risposta al comando giunge al master solo dopo che la velocità teorica del motore è 0, quindi il ritardo è funzione sia della velocità del motore che dell'accelerazione di Hold impostata nei parametri. Per evitare questa attesa è necessario utilizzare il comando Stop (codice 0xbc) descritto nella Sezione 3.5.

5.3 Movimento manuale con assegnamento della velocità (Jog)

Il comando consente di attivare il movimento manuale del motore assegnandone la velocità con un valore prestabilito v . Tale procedura è spesso indicata con il termine "Jog". Il valore di velocità v è un numero a 16 bit con segno, indicato come segue:

| bit 15-8 | bit 7-0 |
|----------|---------|
| VH | VL |

Il segno di v indica il verso di rotazione del motore:

- il segno "+" indica una rotazione verso quote crescenti
- il segno "-" indica una rotazione verso quote decrescenti.

L'unità di misura della velocità è espressa in notazione [Q15].

| Byte | Comando | Risposta | |
|------|---------------|---------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdJog (0x40) | CmdJog (0x40) | |
| 05 | 0x00 | 0x00 | |
| 06 | VH | VH | |
| 07 | VL | VL | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni

Il comando ha effetto solo se il motore è nello stato di regolazione oppure sta effettuando un movimento manuale con assegnamento della velocità (Jog).

Note

La velocità da inserire nei byte 06 e 07 del comando deve essere espressa in notazione [Q15].

Esempio: calcolo dei valori VH e VL per attivare un movimento del motore con una velocità di -185 mm/min, essendo il passo meccanico del motore pari a 0.32 mm/giro. La velocità ha come grandezza base 8000 rpm.

$$-185[\text{mm}/\text{min}] \Leftrightarrow \frac{-185}{0,32 \times 8000} \times 2^{15} = -2368[\text{Q15}] = 0xF6C0[\text{Q15}]$$

| VH | VL |
|------|------|
| 0xF6 | 0xC0 |

Se si vuole inviare la velocità in rpm è necessario utilizzare il comando descritto nella Sezione 3.8.

5.4 Modifica di un parametro dell'azionamento

Mediante questo comando è possibile modificare i parametri dell'azionamento. Questo comando è obsoleto; è preferibile utilizzare il comando descritto nella Sezione 4.4.

La codifica dei parametri modificabili è riportata in Sezione 6.

| Byte | Comando | Risposta | |
|------|------------------------------|------------------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | CmdChgPar (0x04) | 0x0 | BitStato |
| 04 | Parte alta codice parametro | CmdJog (0x40) | |
| 05 | Parte bassa codice parametro | Parte bassa codice parametro | |
| 06 | Parte alta valore parametro | Parte alta valore parametro | |
| 07 | Parte bassa valore parametro | Parte bassa valore parametro | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni

Il comando ha effetto solo se il motore è in allarme, fuori regolazione o in regolazione. Se si vuole effettuare un cambio parametri in unità standard utilizzare il comando di cambio parametri (codice 0xb8) descritto nella Sezione 4.4.

5.5 Azzeramento elettrico con attesa

Azzerare la posizione elettrica del rotore. Questa operazione viene effettuata automaticamente dal motore la prima volta che viene portato in regolazione.

5 Comandi obsoleti

Vi sono diversi tipi di azzeramento: si consiglia di utilizzare sempre il tipo 0.

| Byte | Comando | Risposta | |
|------|-------------------------|-------------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdAzzElWait (0x30) | CmdAzzElWait (0x30) | |
| 05 | 0x00 | 0x00 | |
| 06 | 0x00 | 0x00 | |
| 07 | Tipo azzeramento (0x00) | Tipo azzeramento (0x00) | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni

Il comando ha effetto solo se il motore è fuori regolazione.

Note

- la risposta al comando è data solo al termine dell'azzeramento elettrico (ritardo di circa 0.1 sec).
- se si vuole evitare questo ritardo utilizzare il comando di azzeramento elettrico (codice 0xc4) descritto nella Sezione 4.8.

5.6 Campionamento di variabili

Il comando abilita il campionamento di due variabili interne al firmware.

| Byte | Comando | Risposta | |
|------|--------------------|--------------------|----------|
| 01 | STX (0x02) | STX (0x02) | |
| 02 | Nodo | Nodo | |
| 03 | 0x08 | 0x0 | BitStato |
| 04 | CmdSample (0x88) | CmdSample (0x88) | |
| 05 | 0x00 | 0x00 | |
| 06 | Codice variabile 1 | Codice variabile 1 | |
| 07 | Codice variabile 2 | Codice variabile 2 | |
| 08 | CRC | CRC | |
| 09 | ETX (0x03) | ETX (0x03) | |

Condizioni

Il comando può essere inviato in qualsiasi condizione.

6 Parametri

I parametri, riportati nella seguente tabella, possono essere inviati mediante il comando **CMDCHGPARN (0xb8)** descritto nella Sezione 4.4.

| Codice | Descrizione | Nome | Taglia | Unità di misura | Valori di default | Valori Min/Max |
|--------|--|-----------|--------|-----------------|-------------------|----------------|
| | | | | | SM137 | SM137 |
| | | | | | SM140 | SM140 |
| 0x0100 | Guadagno Proporzionale dell'anello di corrente | KP_I | 16bit | x0.01 | 30 | 0 / 32767 |
| | | | | | 40 | 0 / 32767 |
| 0x0101 | Guadagno Integrativo dell'anello di corrente | KI_I | 16bit | x0.01 | 12 | 0 / 32767 |
| | | | | | 10 | 0 / 32767 |
| 0x0103 | Valore minimo dell'uscita del regolatore Proporzionale Integrativo di corrente | PIMIN_I | 16bit | Volt x0.1 | -194 | -32767 / 0 |
| | | | | | -195 | -32767 / 0 |
| 0x0104 | Valore massimo dell'uscita del regolatore Proporzionale Integrativo di corrente | PIMAX_I | 16bit | Volt x0.1 | 194 | 0 / 32767 |
| | | | | | 195 | 0 / 32767 |
| 0x0105 | Guadagno Proporzionale dell'anello di velocità | KP_VEL | 16bit | x 0.01 | 150 | 0 / 32767 |
| | | | | | 600 | 0 / 32767 |
| 0x0106 | Guadagno Integrativo dell'anello di velocità | KI_VEL | 16bit | x0.01 | 10 | 0 / 32767 |
| | | | | | 50 | 0 / 32767 |
| 0x0107 | Percentuale di feedforward per il regolatore in velocità | KFF_VEL | 16bit | % | 100 | 0 / 100 |
| | | | | | 100 | 0 / 100 |
| 0x0108 | Valore minimo dell'uscita del regolatore Proporzionale Integrativo di velocità. (Questo valore limita la massima coppia fornita dal motore) | PIMIN_VEL | 16bit | Ax0.01 | -500 | -900 / 0 |
| | | | | | -1800 | -3500 / 0 |
| 0x0109 | Valore massimo dell'uscita del regolatore Proporzionale Integrativo di velocità. L'unità di misura è in ampere [Q15](3). (Questo valore limita la massima coppia fornita dal motore) | PIMAX_VEL | 16bit | Ax0.01 | 500 | 0 / 900 |
| | | | | | 1800 | 0 / 3500 |
| 0x010A | Guadagno Proporzionale dell'anello di posizione | KP_POS | 16bit | x0.01 | 500 | 0 / 32767 |
| | | | | | 500 | 0 / 32767 |
| 0x010B | Guadagno Integrativo dell'anello di posizione | KI_POS | 16bit | x0.01 | 0 | 0 / 32767 |
| | | | | | 0 | 0 / 32767 |

| Codice | Descrizione | Nome | Taglia | Unità di misura | Valori di default | Valori Min/Max |
|--------|---|---------------|--------|-----------------|-------------------|--------------------------|
| | | | | | SM137 | SM137 |
| | | | | | SM140 | SM140 |
| 0x010C | Percentuale di feedforward per il regolatore di posizione | KFF_POS | 16bit | % | 70 | 0 / 100 |
| | | | | | 70 | 0 / 100 |
| 0x010D | Valore minimo dell'uscita del regolatore Proporzionale Integrativo di posizione. (Questo valore limita la massima velocità fornita dal motore) | PIMIN_POS | 16bit | rpm | -4500 | -5000 / 0 |
| | | | | | -4500 | -5000 / 0 |
| 0x010E | Valore massimo dell'uscita del regolatore Proporzionale Integrativo di posizione. (Questo valore limita la massima velocità fornita dal motore) | PIMAX_POS | 16bit | rpm | 4500 | 0 / 5000 |
| | | | | | 4500 | 0 / 5000 |
| 0x010F | Massimo Errore di inseguimento in posizione | MAXERRORP | 32bit | Cnt | 0 | 0 / (2 ³¹ -1) |
| | | | | | 0 | 0 / (2 ³¹ -1) |
| 0x0110 | Persistenza dell'errore di inseguimento in posizione oltre il quale il motore va in allarme | TIM_MAXERRORP | 16bit | msec | 0 | 0 / 32000 |
| | | | | | 0 | 0 / 32000 |
| 0x0111 | Massimo Errore di inseguimento in velocità | MAXERRORV | 16bit | rpm | 0 | 0 / 8000 |
| | | | | | 0 | 0 / 8000 |
| 0x0112 | Persistenza dell'errore di inseguimento in velocità oltre il quale il motore va in allarme | TIM_MAXERRORV | 16bit | msec | 0 | 0 / 32000 |
| | | | | | 0 | 0 / 32000 |
| 0x0113 | Tolleranza per dichiarare movimento quotato concluso | INPOS | 16bit | cnt | 0 | 0 / 32000 |
| | | | | | 0 | 0 / 32000 |
| 0x0114 | Tempo in cui la posizione rientra nella tolleranza per dichiarare il movimento concluso | TIM_INPOS | 16bit | msec | 0 | 0 / 32000 |
| | | | | | 0 | 0 / 32000 |
| 0x0115 | Valore di coppia che deve essere sempre superiore a MAXTORQ affinché il motore vada in allarme | MAXTORQ | 16bit | Ax0.01 | 250 | 0 / 32767 |
| | | | | | 900 | 0 / 32767 |
| 0x0116 | Tempo durante il quale la coppia deve essere sempre superiore a AXTORQ affinché il motore vada in allarme | TIM_MAXTORQ | 16bit | msec | 1000 | 0 / 32000 |
| | | | | | 1000 | 0 / 32000 |
| 0x0117 | Velocità massima per movimenti quotati | VMAXPOS | 16bit | rpm | 4000 | 0 / 32767 |
| | | | | | 4000 | 0 / 32767 |

| Codice | Descrizione | Nome | Taglia | Unità di misura | Valori di default | Valori Min/Max |
|--------|--|----------------|--------|----------------------------|---------------------|----------------------------|
| | | | | | SM137 | SM137 |
| | | | | | SM140 | SM140 |
| 0x0118 | Velocità di azzeramento automatico | VMAXAZZ | 16bit | rpm | 0 | -32767 / 32767 |
| | | | | | 500 | -32767 / 32767 |
| 0x0119 | Accelerazione utilizzata durante i movimenti non quotati e le fermate comandate in HOLD | AMAX | 16bit | 10000 cnt/sec ² | 48 | 0 / 32767 |
| | | | | | 40 | 0 / 32767 |
| 0x011A | Accelerazione utilizzata durante i movimenti quotati | AMAXPOS | 16bit | 10000 cnt/sec ² | 32 | 0 / 32767 |
| | | | | | 20 | 0 / 32767 |
| 0x011B | Quota assegnata durante l'azzeramento automatico | ORIG_AZZ | 32bit | cnt | 0 | $-(2^{31})-1/(2^{31}-1)$ |
| | | | | | 0 | $-(2^{31})-1/(2^{31}-1)$ |
| 0x011C | Limite software inferiore | LOW_SLIM | 32bit | cnt | - | $-(2^{31})-1 / (2^{31})-1$ |
| | | | | | $32767*2^{16}$ | $-(2^{31})-1 / (2^{31})-1$ |
| 0x011D | Limite software superiore | HIGH_SLIM | 32bit | cnt | $32767*2^{16}$ | $-(2^{31})-1 / (2^{31})-1$ |
| | | | | | - | $-(2^{31})-1 / (2^{31})-1$ |
| 0x011E | Ha un significato bit a bit descritto nella Tabella 1 | BIT_A | 16bit | | 0 | |
| 0x011F | Parametro riservato NON MODIFICARE | ANGELETTRTACCA | 16bit | | 676 | 0 / 799 |
| | | | | | 676 | 0 / 1999 |
| 0x0120 | Primo componente feedforward di velocità | RESERVED1 | | | 600 | |
| | | | | | 300 | |
| 0x0121 | Secondo componente feedforward di velocità | RESERVED2 | | | 1400 | |
| | | | | | 600 | |
| 0x0122 | Terzo componente feedforward di velocità | RESERVED3 | | | 2600 | |
| | | | | | 2200 | |
| 0x0123 | Quarto componente feedforward di velocità | RESERVED4 | | | 6 | 16 |
| | | | | | | |
| 0x0124 | Parametro riservato | RESERVED5 | | | | |
| 0x0125 | Parametro riservato | RESERVED6 | | | | |
| 0x0126 | Parametro riservato | RESERVED7 | | | | |
| 0x0127 | Parametro riservato | RESERVED8 | | | | |
| 0x0128 | Parametro riservato | RESERVED9 | | | | |
| 0x012A | Parametro riservato | RESERVED11 | | | | |
| 0x012B | Parametro riservato | RESERVED11 | | | | |
| 0x012C | Su Enet-X: tempo di attuazione del bus. Deve essere un numero multiplo di 4. Su 485: ritardo minimo nella risposta del motore ad un comando | TIMFB | 16bit | msec | 8 EnetX 50 RS485 | 1/100 |
| | | | | | 8 EnetX 50 RS485 | 1/100 |

| Codice | Descrizione | Nome | Taglia | Unità di misura | Valori di default | Valori Min/Max |
|--------|---|------------|--------|-----------------|---------------------|----------------|
| | | | | | SM137 | SM137 |
| | | | | | SM140 | SM140 |
| 0x012D | Tempo trascorso dalla ricezione del messaggio oltre il quale il motore si pone nello stato di ALLARME | TIMEOUTFB | 16bit | | 8 EnetX 50 RS485 | 1/100 |
| | | | | | 8 EnetX 50 RS485 | 1/100 |
| 0x012E | Solo per SM140 Bit a bit indica il livello logico degli ingressi digitali. | INVDIN | 16bit | | 0 | 0 / 7 |
| | | | | | | |
| 0x012F | Solo per SM140 Tipo azzeramento automatico. Si veda il comando CmdAzz. | TIPOAZZ | 16bit | | 0 | 0 / 2 |
| | | | | | | |
| 0x0130 | Solo per SM140 Velocità di uscita dal microinterruttore di azzeramento. | VAZZOUTMIC | 16bit | rpm | -125 | -32767 / 32767 |
| | | | | | | |

Tabella 1 : Descrizione del significato dei bit del parametro **BIT_A (0x011E)**

| Byte | Default | Significato |
|------|---------|--|
| 0 | 0 | Se posto a 1 abilita il controllo dei limiti di finecorsa software |
| 1 | 0 | Riservato. Lasciare a 0 |
| 2 | 0 | Riservato. Lasciare a 0 |
| 3 | 0 | Se posto a 1 inverte il verso di rotazione standard del motore. Il verso positivo di rotazione standard è antiorario guardando l'albero dal lato della flangia |
| 4 | 0 | Riservato. Lasciare a 0 |
| 5 | 0 | Riservato. Lasciare a 0 |
| 6 | 0 | Solo per SM140 e SM137 rev. C Se posto a 1 disabilita CMDNOREG per uscire dallo stato di allarme. In questo caso l'unico comando per uscire dallo stato di allarme è CMDRESET |
| 7 | 0 | Riservato. Lasciare a 0 |
| 8 | 0 | Solo per SM140 Se posto a 1 abilita la camma di extracorsa negativa. |
| 9 | 0 | Solo per SM140 Se posto a 1 abilita la camma di extracorsa positiva. |
| 10 | 0 | Solo per SM140 Se posto a 1 impone che il motore si ponga nello stato di allarme qualora incontri una delle due camme di extracorsa. Se posto a 0 impone che il motore si ponga nello stato AXSTOP , con una opportuna rampa di decelerazione, qualora incontri una delle due camme di extracorsa. |
| 11 | 0 | Versione 119 del firmware e successive Riservato a EnetX: se posto a 1 non effettua l'aggancio tra bus e regolazione |

| | | |
|----|---|---|
| 12 | 0 | <p>Solo per SM140 versione 119 del firmware e successive</p> <p>Se posto a 0 impone che il motore effettui sempre una frenata controllata prima di uscire dalla regolazione. Con questo bit a 0 dopo ogni condizione di allarme (eccetto ALOVERCURR e ALOVERPOWER per le quali si rischierebbe di danneggiare l'azionamento) o dopo un comando CMDGOEMERG o CMDNOREG il motore verifica che la propria velocità di rotazione sia 0. Se così non è, effettua una frenata controllata utilizzando un valore di decelerazione pari al valore del parametro AMAX. Durante questa fase se l'inerzia è particolarmente elevata è necessario applicare al motore una resistenza di frenatura (modulo P144 fornito da CNI).</p> |
|----|---|---|

I parametri sono inviabili attraverso il comando **CMDCHGPARN (0xb8)**.

6.1 Taratura dei parametri

6.1.1 Struttura del regolatore

L'azionamento è costituito da tre regolatori innestati uno sull'altro:

- il regolatore di corrente che riceve il riferimento dal regolatore di velocità (riferimento in coppia/corrente) e pilota direttamente il PWM
- il regolatore di velocità che riceve il riferimento dal regolatore di posizione (riferimento in velocità) e fornisce il riferimento in coppia al regolatore di corrente
- il regolatore di posizione che riceve il riferimento dal generatore di traiettoria e fornisce il riferimento al regolatore di velocità.

Un regolatore generalmente è composto da 2 parti:

- una parte in "retroazione" che lavora sull'errore di inseguimento
- una parte in "feedforward" che lavora solo sul riferimento

La parte in retroazione è fondamentale: è quella che in effetti permette al motore di raggiungere il suo riferimento e nell'azionamento è composta da una parte proporzionale ed una integrativa. La parte in retroazione riceve in ingresso l'errore di inseguimento ed in base ad esso aumenta o diminuisce l'uscita del regolatore.

Considerando, ad esempio, solo la componente proporzionale del regolatore di posizione, se l'errore di inseguimento aumenta (il motore non riesce a raggiungere la posizione comandata), la componente proporzionale della retroazione aumenta "proporzionalmente" l'uscita del regolatore di posizione.

Tale uscita costituisce il riferimento di velocità per il regolatore di velocità.

Quindi: se il motore non riesce a raggiungere la posizione comandata, il regolatore di posizione aumenta il riferimento di velocità imponendo al regolatore di velocità un aumento della velocità del motore in modo che il motore possa diminuire l'errore di posizione.

Si capisce che se l'errore di posizione è 0, il regolatore di posizione richiede una velocità 0 al regolatore di velocità e quindi il motore si ferma generando un errore di posizione. In altre parole: la parte in retroazione del regolatore, per lavorare "correttamente", ha sempre bisogno di un errore di inseguimento diverso da 0 e quindi non consente di raggiungere stabilmente il riferimento.

Inoltre per ottenere un errore basso è necessario aumentare i guadagni della parte di retroazione ma questo avvicina il sistema controllato all'instabilità innescando vibrazioni sempre più elevate all'aumentare dei guadagni.

Soprattutto per questi motivi, è necessaria la parte in feedforward: questa parte, in base al riferimento, fornisce in uscita al regolatore un valore tale che teoricamente l'errore di inseguimento sia 0.

Sempre considerando il regolatore di posizione, supponendo che la parte di feedforward sia al 100% e che la parte di retroazione sia esclusa (guadagno proporzionale ed integrativo a 0) se il riferimento di posizione è dato con una velocità di 2000 rpm, allora il regolatore di posizione fornisce al regolatore di velocità un riferimento di 2000 rpm, anche in assenza di errore di inseguimento.

Se la parte di feedforward è al 50%, questa fornirebbe un riferimento al regolatore di velocità di 1000rpm.

Questo ci consente di ottenere bassi errori di inseguimento anche con bassi guadagni nella parte di retroazione. Contemporaneamente, abbassando i guadagni ci allontaniamo dal punto in cui il sistema diventa instabile e quindi riusciamo a controllare assi anche con rapporti di inerzia molto sbilanciati.

Però la parte in feedforward ha il difetto di imporre all'asse un controllo piuttosto rigido e quindi, se portata ai limiti, rischia di sollecitare troppo la meccanica dell'asse stesso.

La taratura dei parametri è proprio questo: trovare il giusto compromesso fra tutte le componenti.

6.1.2 Taratura della parte in feedforward del regolatore

Come principio generale occorre sempre tarare per primi i regolatori più interni.

Il regolatore di corrente risente quasi esclusivamente delle caratteristiche elettriche del motore e quindi va sempre lasciato con i suoi valori di default.

Occorre iniziare dal regolatore di velocità ed in particolare dalla sua componente di feedforward.

Parametri da tarare

I parametri che devono essere tarati sono:

- RESERVED1 (0x0120)
- RESERVED2 (0x0121)
- RESERVED3 (0x0122)
- RESERVED4 (0x0123)

Preparazione

- durante le prove è necessario che l'asse possa muoversi in sicurezza per la corsa più lunga possibile.
- vanno escluse le componenti in retroazione del regolatore di posizione e di quello di velocità: occorre mettere a zero i guadagni proporzionale ed integrativo di questi due regolatori
- occorre far risaltare al massimo le componenti di feedforward: è necessario mettere al 100% i guadagni di feedforward dei regolatori di posizione e velocità.

Passo 1: taratura dei parametri RESERVED1 e RESERVED2

In questo passo tariamo la componente di feedforward necessaria a vincere gli attriti della macchina.

- poniamo a 0 sia RESERVED3 che RESERVED4
- RESERVED1 è la velocità al di sopra della quale interviene il feedforward che si oppone agli attriti.
È espresso in notazione [Q15]. Di norma un valore tipico è 200 (su SM137 ed SM140 questo significa che l'intervento avverrà per velocità superiori a:

$$\frac{200}{32768} \times 8000 = 48,8 \text{ rpm}$$

- **RESERVED2** è la quantità di coppia necessaria a superare gli attriti e va tarata per tentativi. Si effettua un JOG con l'asse e si aumenta **RESERVED2** fino a quando non si percepisce che spingendo a mano l'asse nello stesso verso del JOG l'asse viene spostato con poca fatica. È importante verificare che l'asse si fermi dopo che non lo si spinge più con la mano, altrimenti è necessario ridurre **RESERVED2**.

Passo 2: taratura del parametro RESERVED3

Ora tariamo la componente di feedforward che risulta proporzionale alla velocità di riferimento.

- utilizzando i valori di **RESERVED1** e **RESERVED2** trovati in precedenza, muoviamo in JOG l'asse alla stessa velocità a cui normalmente lavorerà
- aumentiamo lentamente (incrementi di 100) **RESERVED3** fino a quando la velocità del motore a regime (dopo la rampa di accelerazione) sia uguale o molto vicina a quella impostata nel JOG.

Passo 3: taratura del parametro RESERVED4

A questo punto tariamo la componente che interviene durante le accelerazioni e le decelerazioni.

- utilizzando i valori di **RESERVED1**, **RESERVED2** e **RESERVED3** trovati in precedenza, muoviamo in JOG l'asse alla stessa velocità ed accelerazione a cui normalmente lavorerà.
- osservando l'andamento della velocità reale e di quella teorica lungo le rampe, aumentiamo **RESERVED4** molto lentamente (incrementi di 2) fino a quando le due velocità non coincidono.

Passo 4: Verifica

Effettuando dei movimenti quotati a velocità di lavoro si dovrebbe osservare che nonostante le componenti in retroazione siano annullate, l'asse raggiunge la quota obiettivo con una discreta precisione.

6.1.3 Taratura della parte in retroazione del regolatore di velocità

Se la taratura della parte di feedforward è avvenuta con successo, i valori di default dei guadagni dovrebbero già essere soddisfacenti.

Si può, per tentativi, aumentarli o diminuirli uno per volta per vedere se le prestazioni migliorano o peggiorano.

In generale, conviene lasciare il feedforward al 100% e tenere un guadagno integrativo diverso da 0 ma, a parte questo, è la sensibilità di chi effettua la taratura a capire come occorre modificare i guadagni.

Le prove vanno effettuate muovendo il motore con il comando di JOG.

6.1.4 Taratura della parte in feedforward del regolatore di posizione

Per il regolatore di posizione occorre tarare esclusivamente la percentuale di feedforward tenendo conto anche delle sollecitazioni innescate sulla meccanica.

Le prove vanno effettuate con la parte in retroazione annullata e utilizzando dei movimenti quotati. Generalmente il valore varia dal 50% al 100%

6.1.5 Taratura della parte in feedforward del regolatore di posizione

Per il regolatore di posizione occorre tarare esclusivamente la percentuale di feedforward tenendo conto anche delle sollecitazioni innescate sulla meccanica.

Le prove vanno effettuate con la parte in retroazione annullata e utilizzando dei movimenti quotati. Generalmente il valore varia dal 50% al 100% .

A Parametri di basso livello

ATTENZIONE: i parametri descritti in seguito non possono essere salvati nella flash-memory. I parametri di basso livello sono mantenuti per compatibilità software ma è consigliato non utilizzarli.

I parametri di basso livello possono essere inviati mediante il comando **CMDCHGPAR (0x04)**.

| Codice | Descrizione | Nome | Valori di default | Valori Min/Max |
|--------|--|------------|-------------------|----------------|
| | | | SM137 | SM137 |
| | | | SM140 | SM140 |
| 0x1000 | Mantissa del guadagno proporzionale-integrativo dell'anello di corrente, in unità di misura [ME15] | KIQDPIMANT | 13762 | 0 / 32767 |
| | | | 16383 | 0 / 32767 |
| 0x1001 | Esponente del guadagno proporzionale-integrativo dell'anello di corrente in unità di misura [ME15] | KIQDPIEXP | 0 | 0 / 6 |
| | | | 0 | 0 / 6 |
| 0x0002 | Guadagno del fattore di correzione dell'anello di corrente | KIQDKCOR | 9362 | 0 / 32767 |
| | | | 6552 | 0 / 32767 |
| 0x0003 | Non utilizzato | KIQDKFF | -- | -- |
| 0x0004 | Valore minimo dell'uscita del regolatore Proporzionale Integrativo di corrente. L'unità di misura è espressa in Volt; il valore deve essere dato in notazione [Q15]. | KIQDLIMIN | -32767 | -32767/0 |
| | | | | |
| 0x0005 | Valore massimo dell'uscita del regolatore Proporzionale Integrativo di corrente. L'unità di misura è espressa in Volt; il valore deve essere dato in notazione [Q15]. | KIQDLIMAX | 32767 | 0 / 32767 |
| | | | 32767 | 0 / 32767 |
| 0x2006 | Mantissa del guadagno proporzionale integrativo dell'anello di velocità in unità di misura [ME15] | KVELPIMANT | 26214 | 0 / 32767 |
| | | | 26624 | 0 / 32767 |
| 0x2007 | Esponente del guadagno proporzionale integrativo dell'anello di velocità in unità di misura [ME15] | KVELPIEXP | 1 | 0 / 6 |
| | | | 3 | 0 / 6 |
| 0x0008 | Guadagno del fattore di correzione dell'anello di velocità | KVELKCOR | 2047 | 0 / 32767 |
| | | | 2520 | 0 / 32767 |
| 0x0009 | Percentuale di feedforward per il regolatore in velocità | KVELKFF | 32767 | 0 / 32767 |
| | | | 32767 | 0 / 32767 |
| 0x000A | Valore minimo dell'uscita del regolatore Proporzionale Integrativo di velocità. L'unità di misura è espressa in A; il valore deve essere dato in notazione [Q15]. In pratica, questo parametro limita la coppia massima richiesta al motore nel verso negativo di rotazione | KVELLIMIN | -14486 | -26067 / 0 |
| | | | -13034 | -25342 / 0 |

| Codice | Descrizione | Nome | Valori di default | Valori Min/Max |
|--------|--|------------|-------------------|----------------|
| | | | SM137 | SM137 |
| | | | SM140 | SM140 |
| 0x000B | Valore massimo dell'uscita del regolatore Proporzionale Integrativo di velocità. L'unità di misura è espressa in A; il valore deve essere dato in notazione [Q15]. In pratica, questo parametro limita la coppia massima richiesta al motore nel verso positivo di rotazione | KVELLIMAX | 14486 | 0 / 26067 |
| | | | 13034 | 0 / 25342 |
| 0x300C | Mantissa del guadagno proporzionale integrativo in dell'anello di posizione in unità di misura [ME15] | KPOSPIMANT | 20480 | 0 / 32767 |
| | | | 20480 | 0 / 32767 |
| 0x300D | Esponente del guadagno proporzionale integrativo dell'anello di posizione in unità di misura [ME15] | KPOSPIEXP | 3 | 0 / 6 |
| | | | 3 | 0 / 6 |
| 0x000E | Guadagno del fattore di correzione dell'anello di posizione | KPOSKCOR | 0 | 0 / 32767 |
| | | | 0 | 0 / 32767 |
| 0x000F | Percentuale di feedforward per il regolatore in posizione | KPOSKFF | 22937 | 0 / 32767 |
| | | | 22937 | 0 / 32767 |
| 0x0010 | Valore minimo dell'uscita del regolatore Proporzionale Integrativo di posizione. L'unità di misura è espressa in rpm; il valore deve essere dato in notazione [Q15]. In pratica, questo parametro limita la velocità massima richiesta al motore nel verso negativo di rotazione | KPOSLIMMIN | -18432 | -32767 / 0 |
| | | | -18432 | -32767 / 0 |
| 0x0011 | Valore massimo dell'uscita del regolatore Proporzionale Integrativo di posizione. L'unità di misura è espressa in rpm; il valore deve essere dato in notazione [Q15]. In pratica, questo parametro limita la velocità massima richiesta al motore nel verso positivo di rotazione | KPOSLIMAX | 18432 | 0 / 32767 |
| | | | 18432 | 0 / 32767 |
| 0x4012 | 16 bit meno significativi del massimo errore di inseguimento in posizione, espresso in conteggi (1) | MAXERRORPL | 0 | -32768 / 32767 |
| | | | 0 | -32768 / 32767 |
| 0x4013 | 16 bit più significativi del massimo errore di inseguimento in posizione, espresso in conteggi (1) | MAXERRORPH | 0 | 0 / 32767 |
| | | | 0 | 0 / 32767 |
| 0x0014 | Accelerazione utilizzata durante i movimenti non quotati e le fermate comandate in HOLD. L'unità di misura è espressa in giri/sec ² ; il valore deve essere dato in notazione [Q15] | AMAXHOLD | 2089 | 1 / 32767 |
| | | | 696 | 1 / 32767 |
| 0x0015 | Accelerazione utilizzata durante i movimenti quotati. L'unità di misura è espressa in giri/sec ² ; il valore deve essere dato in notazione [Q15] | AMAXTRAJ | 1392 | 1 / 32767 |
| | | | 348 | 1 / 32767 |
| 0x0016 | Velocità massima dei movimenti quotati. L'unità di misura è espressa in rpm; il valore deve essere dato in notazione [Q15] (1) | VMAXTRAJ | 16384 | 0 / 32767 |
| | | | 16384 | 0 / 32767 |

| Codice | Descrizione | Nome | Valori di default | |
|--------|---|----------------|-------------------|----------------|
| | | | SM137 | SM137 |
| | | | SM140 | SM140 |
| 0x5017 | 16 bit meno significativi della quota assegnata durante l'azzeramento automatico. L'unità di misura è espressa in conteggi | ORIGINEL | 0 | -32768 / 32767 |
| | | | 0 | -32768 / 32767 |
| 0x5018 | 16 bit più significativi della quota assegnata durante l'azzeramento automatico. L'unità di misura è espressa in conteggi | ORIGINEH | 0 | -32768 / 32767 |
| | | | 0 | -32768 / 32767 |
| 0x0019 | Ha un significato bit a bit: bit0=1: abilita il controllo dei limiti software bit1: riservato bit2: riservato bit3: la variazione del valore inverte il verso di rotazione del motore Il significato dei bit è identico a quello descritto nella tabella 1 di pagina 48. | TESTSLIM | 0 | |
| 0x601A | 16 bit meno significativi del limite software inferiore. L'unità di misura è espressa in conteggi | LOWSLIML | 0 | -32768 / 32767 |
| | | | 0 | -32768 / 32767 |
| 0x601B | 16 bit più significativi del dato di limite software inferiore espresso in conteggi | LOWSLIMH | 0 | -32768 / 32767 |
| | | | 0 | -32768 / 32767 |
| 0x601C | 16 bit meno significativi del dato di limite software superiore espresso in conteggi | HIGHSLIML | 0 | -32768 / 32767 |
| | | | 0 | -32768 / 32767 |
| 0x601D | 16 bit più significativi del dato di limite software superiore espresso in conteggi | HIGHSLIMH | 0 | -32768 / 32767 |
| | | | 0 | -32768 / 32767 |
| 0x001E | Massimo errore di inseguimento in velocità. L'unità di misura è espressa in rpm; il valore deve essere dato in notazione [Q15] | MAXERRORV | 0 | 0 / 32767 |
| | | | 0 | 0 / 32767 |
| 0x001F | Massima coppia richiedibile al motore per più di TIMEOUTMAXTORQUE msec. L'unità di misura è espressa in Ampere; il valore deve essere dato in notazione [Q15] | MAXTORQUE | 7243 | 0 / 32767 |
| | | | 6517 | 0 / 32767 |
| 0x0020 | Parametro riservato. | ANGELETTRTACCA | 676 | -- |
| | | | 676 | -- |
| 0x0021 | Parametro riservato | RESERVED1 | 600 | 0 / 32767 |
| | | | 300 | 0 / 32767 |
| | | | | |
| | | | | |
| 0x0022 | Parametro riservato | RESERVED2 | 1400 | 0 / 32767 |
| | | | 600 | 0 / 32767 |

A Parametri di basso livello

| Codice | Descrizione | Nome | Valori di default | Valori Min/Max |
|--------|---|------------------|-------------------|----------------|
| | | | SM137 | SM137 |
| | | | SM140 | SM140 |
| 0x0023 | Parametro riservato | RESERVED3 | 2600 | 0 / 32767 |
| | | | 2200 | 0 / 32767 |
| 0x0024 | Parametro riservato | RESERVED4 | 6 | 0 / 32767 |
| | | | 16 | 0 / 32767 |
| 0x0025 | Parametro riservato | RESERVED5 | | |
| 0x0026 | Parametro riservato | RESERVED6 | | |
| 0x0027 | Parametro riservato | RESERVED7 | | |
| 0x0028 | Parametro riservato | RESERVED8 | | |
| 0x0029 | Parametro riservato | RESERVED9 | | |
| 0x002A | Parametro riservato | RESERVED10 | | |
| 0x002B | Parametro riservato | RESERVED11 | | |
| 0x002C | Parametro riservato | RESERVED12 | 0 | |
| 0x002D | Tempo durante il quale la coppia superare il valore del parametro MAXTORQUE affinché il motore vada in allarme. In [msec] | TIMEOUTMAXTORQUE | 1000 | 0 / 32767 |
| | | | 1000 | 0 / 32767 |

Legenda:

- (1) Un valore nullo di questo parametro disabilita il controllo relativo.

Rappresentazione di un numero in notazione [Q15].

La rappresentazione di un numero in notazione [Q15] prevede la moltiplicazione di un numero reale per il massimo valore esprimibile con 15 bit cioè per 2^{15} ed effettuare un troncamento sul numero risultante.

In questo modo si ha una corrispondenza lineare tra valori decimali e valori interi.

La notazione [Q15] è spesso utilizzata per l'assegnazione del valore dei parametri del motore. Data una grandezza f ed il suo valore massimo, detto anche "valore base" e indicato dalla sigla f_{base} , la grandezza f è rappresentata in notazione [Q15] attraverso la formula seguente:

$$f_{[Q15]} = \frac{f}{f_{base}} \times 2^{15} \quad \text{con} \quad -32767 \leq f_{[Q15]} \leq 32767$$

Ad esempio: se sul motore SM137 la corrente base è pari a 11.313 A e la corrente assorbita è pari a $I=5^\circ$, il valore di corrente assorbita in notazione [Q15] sarà pari a:

$$5[A] \Leftrightarrow \frac{5}{11,313} \times 2^{15} = 14482[Q15]$$

Notazione mantissa-esponente [ME15].

Per grandezze che non hanno una grandezza base di riferimento può essere utilizzata la notazione [ME15] che prevede una mantissa ed un esponente:

$$f[M15] = (f_{mant}; f_{esp}) \Leftrightarrow f = \frac{f_{mant}}{2^{15}} \times 2^{f_{esp}}$$

Si noti che la notazione [ME15] non è univoca, cioè: uno stesso numero può essere rappresentato in tale notazione in modi diversi.

Ad esempio, il numero 15.5 può essere rappresentato nei seguenti modi:

$$f_1[M15] = (31744;4) \Leftrightarrow f_1 = \frac{31744}{2^{15}} \times 2^4 = 15,5$$

$$f_2[M15] = (15872;5) \Leftrightarrow f_2 = \frac{15872}{2^{15}} \times 2^5 = 15,5$$

Guadagni dei regolatori Proporzionali Integrativi di corrente, velocità e posizione.

I guadagni impostabili nei regolatori Proporzionali Integrativi presenti nel motore si possono suddividere in:

- guadagni che lavorano in retroazione;
- guadagni che lavorano in feedforward.

I guadagni che lavorano in retroazione si basano su una struttura proporzionale-integrativa con limitazione della grandezza in uscita e correzione della componente integrale.

Questi guadagni sono 3:

- guadagno proporzionale-integrativo parte mantissa (KxxxPIMANT);
- guadagno proporzionale-integrativo parte esponente (KxxPIEXP);
- guadagno fattore di correzione;

Le relazioni con i classici guadagni proporzionale (K_p) ed integrativo (K_i) sono le seguenti:

$$\left\{ \begin{array}{l} K_{pi} = K_p + K_i \\ K_{cor} = \frac{K_i}{K_p + K_i} \end{array} \right.$$

Inoltre:

$$\left\{ \begin{array}{l} K_{xxxPIMANT_M}; K_{xxxPIEXP_M} = K_{pi}[ME15] \\ K_{xxxKCOR_M} = K_{cor}[Q15] \end{array} \right.$$

Per Kcor la grandezza base è 1.

A Parametri di basso livello

Il guadagno che lavora in feedforward è espresso intermini percentuale ed ha una notazione di tipo [Q15] considerando come valore base 100.

Esempio:

$$\left\{ \begin{array}{l} K_p = 0,5 \\ K_i = 0,08 \\ K_{FF} = 15 \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{l} K_{pi} = 0,5 + 0,08 = 0,58 \\ K_{cor} = \frac{0,08}{0,58} = 0,1379 \\ K_{FF} = 15 \end{array} \right\} \left\{ \begin{array}{l} K_{xxxPIMANT_M} = 0,58 \times 2^{15} = 19005 \\ K_{xxxPIEXP_M} = 0 \\ K_{xxxKCOR_M} = 0,1379 \times 2^{15} = 4520 \\ K_{xxxKFF_M} = \frac{15}{100} \times 2^{15} = 4915 \end{array} \right.$$

L'uscita del regolatore Proporzionale Integrativo viene poi limitata da un valore massimo ed un valore minimo che sono rispettivamente: KxxxLIMMIN e KxxxLIMMAX. Entrambi sono rappresentati in notazione [Q15].

Grandezze base per la notazione [Q15]

| Unità di misura grandezze | Grandezza base SM137 | Grandezza base SM140 |
|-------------------------------------|----------------------|----------------------|
| Tensione in volt [V] | 19.4 | 19.6 |
| Corrente o Coppia in ampere [A] | 11.3137 | 45.2548 |
| Velocità in giri al minuto [rpm] | 8000 | 8000 |
| Accelerazione giri/sec ² | 9411 | 9411 |

Quantizzazione SM137

La quantizzazione del parametro **AMAXHOLD** è di 4.096 [giri/sec²] mentre quella del parametro **AMAXTRAJ** è di:

- 78.125 [giri/sec²] per la revisione B
- 19.53125 [giri/sec²] per la revisione C

Tale quantizzazione implica che se, per esempio, si ha un passo di 0.32 mm/giro e si impone un'accelerazione **AMAXTRAJ** di 430 [mm/sec²] si ottiene un'accelerazione di 17 78.125 0.32=425 [mm/sec²].

B Stati interni

All'accensione, lo stato del motore è di tipo **AXALARM**.

Lo stato corrente del motore può essere letto attraverso il comando **CmdGetSmStat** (Sez. 4.18).

La seguente tabella riporta tutti gli stati interni del motore.

| Codice | Significato | Nome |
|--------|---|------------|
| 0x0000 | Il motore si trova fuori regolazione; se l'albero viene spostato dalla posizione corrente il motore non oppone resistenza | AXNOREG |
| 0x0001 | L'asse è in allarme. Dal punto di vista meccanico il motore è fuori regolazione. A differenza di quanto avviene per lo stato AXNOREG , non può andare direttamente in regolazione | AXALARM |
| 0x0002 | Il motore è in regolazione. In questo stato il motore cerca di mantenere la posizione corrente del rotore | AXSTOP |
| 0x0003 | Il motore sta tarando gli offset dei sensori di corrente. | AXADCOFF |
| 0x0004 | L'asse sta azzerando la posizione elettrica del rotore. | AXAZZEL |
| 0x0006 | Il motore sta effettuando un azzeramento automatico ed è in fase di ricerca della tacca di zero dell'encoder | AXAZZAUTO |
| 0x0007 | L'asse si sta fermando con una decelerazione pari a quella impostata nel parametro AMAXHOLD | AXHOLD |
| 0x0008 | Il motore si sta muovendo a velocità costante. | AXFREERUN |
| 0x0009 | L'asse sta eseguendo un movimento quotato con accelerazione pari al valore impostato nel parametro AMAXTRAJ e velocità massima pari al valore impostato nel parametro VMAXTRAJ | AXEXEC |
| 0x000A | Non implementato | AXTORQUE |
| 0x000B | Il motore è in fase di interpolazione (solo per versioni del firmware >=110) | AXINTERP |
| 0x000D | Il motore sta effettuando il latch della quota sulla tacca di zero (solo per versioni del firmware >=110) | AXLATCH |
| 0x000E | Il motore sta cercando di effettuare una frenata controllata in modo da fermarsi con l'accelerazione descritta nel parametro 0x119 . Appena il motore sarà fermo il suo nuovo stato diventerà AXALARM . Questo stato è presente su SM140 a partire dalla versione 119 | PREAXALARM |
| 0x000F | Il motore sta cercando di effettuare una frenata controllata in modo da fermarsi con l'accelerazione descritta nel parametro 0x119 . Appena il motore sarà fermo il suo nuovo stato diventerà AXNOREG . Questo stato è presente su SM140 a partire dalla versione 119 | PREAXNOREG |

Nelle tabelle seguenti sono illustrate tutte le possibili combinazioni tra stato dell'azionamento, comando inviato ed effetto generato da ciascun comando.

Legenda dei simboli utilizzati:

- “=” lo stato rimane invariato
- “errore” il comando non viene accettato e il motore risponde con il comando **CMDNACK**.
- “Stato 1 ⇒ Stato 2”: alla ricezione del comando il motore si porta nello stato “Stato 1”. Al termine delle operazioni svolte nello “Stato 1”, lo stato del motore diviene lo “Stato 2”
- “*” comandi e stati attivi del motore SM140 e a partire dalla revisione C del motore SM137

N.B.: lo stato iniziale del motore è AXALARM

| Comando | Stato | | | | | | |
|-----------------|----------------|--|----------|----------------|--|----------|--|
| | PREAXALARM (6) | AXALARM | AXADCOFF | PREAXNOREG (6) | AXNOREG | AXAZZEL | AXSTOP |
| CMDEMERG | == errore | AXALARM | AXALARM | == errore | AXALARM | AXALARM | AXALARM |
| CMDADCOFF | == errore | AXADCOFF ⇒AXALARM | = errore | == errore | = errore | = errore | = errore |
| CMDNOREG | == errore | AXNOREG | = errore | == errore | = | | AXNOREG |
| CMDAZZEL | == errore | = errore | = errore | == errore | AXAZZEL⇒ AXNOREG | = errore | = errore |
| CMDREG | == errore | = errore | = errore | == errore | (1)AXAZZEL⇒ AXSTOP | = errore | = |
| CMDAZZ | == errore | = errore | = errore | == errore | = errore | = errore | = errore oppure AXAZZAUTO⇒ AXHOLD⇒ AXSTOP |
| CMDMAZZ | == errore | = Effettua l'azzeramento manuale | = errore | == errore | = Effettua l'azzeramento manuale | = errore | = Effettua l'azzeramento manuale |
| CMDJOG | == errore | = errore | = errore | == errore | = errore | = errore | AXFREERUN |
| CMDTRAJ | == errore | = errore | = errore | == errore | = errore | = errore | AXEXEC ⇒ AXSTOP (3) |
| CMDHOLD | == errore | = errore | = errore | == errore | = errore | = errore | = |
| CMDCHGPAR | == errore | = | = errore | == errore | = | = errore | = |
| CMDGETPAR | == | = | = | == | = | = | = |
| CMDGETALARM | == | = | = | == | = | = | = |
| CMDGETPOS | == | = | = | == | = | = | = |
| CMDGETPOST | == | = | = | == | = | = | = |
| CMDGETVEL | == | = | = | == | = | = | = |
| CMDGETTOR | == | = | = | == | = | = | = |
| CMDGETVER | == | = | = | == | = | = | = |
| CMDGETSMSTAT | == | = | = | == | = | = | = |
| CMDSAMPLE | == | = | = | == | = | = | = |
| CMDGETSAMP | == | = | = | == | = | = | = |
| CMDCHGPARN * | == errore | = | = errore | == errore | = | = errore | = |
| CMDGETPARN * | == | = | = | == | = | = | = |
| CMDTRAJVEL * | == errore | = errore | = errore | == errore | = errore | = errore | AXEXEC⇒ AXSTOP(3) |
| CMDSETOVERR * | == | = | = | == | = | = | = |
| CMDGETOVERR * | == | = | = | == | = | = | = |
| CMDJOGN * | == errore | = errore | = errore | == errore | = errore | = errore | AXFREERUN |
| CMDSAVEPARFL * | == errore | | | == errore | | = errore | = errore |
| CMDERASEFIR * | == errore | | | == errore | | = errore | = errore |
| CMDCHGBOOT * | == errore | | | == errore | | = errore | = errore |
| CMDGOTOBOOT * | == errore | | | == errore | | = errore | = errore |
| CMDGOINTERP * | == errore | = errore | = errore | == errore | = errore | = errore | AXINTERP |
| CMDSTOPINTERP * | == errore | = | = | == errore | = | = | = |
| CMDLATCHINTERP* | == errore | = errore | = errore | == errore | = errore | = errore | = errore |

| Comando | Stato | | | | | |
|----------------|-------------------|---|-------------------|----------|-------------|------------|
| | AXAZZAUTO | AXFREERUN | AXEXEC | AXHOLD | AXINTERP(*) | AXLATCH(*) |
| CMDEMERG | AXALARM | AXALARM | AXALARM | AXALARM | = errore | = errore |
| CMDADCOFF | = errore | = errore | = errore | = errore | = errore | = errore |
| CMDNOREG | AXNOREG | AXNOREG | AXNOREG | AXNOREG | = errore | = errore |
| CMDAZZEL | = errore | = errore | = errore | = errore | = errore | = errore |
| CMDREG | = errore | = errore | = errore | = errore | = errore | = errore |
| CMDAZZ | = | AXAZZAUTO⇒ AXHOLD⇒AXSTOP oppure = errore | = errore | = errore | = errore | = errore |
| CMDMAZZ | = errore | = errore | = errore | = errore | = errore | = errore |
| CMDJOG | = errore | =(2) | = errore | = errore | = errore | = errore |
| CMDTRAJ | = errore | = errore | = errore | = errore | = errore | = errore |
| CMDHOLD | AXHOLD⇒ AXSTOP | AXHOLD⇒ AXSTOP | AXHOLD⇒ AXSTOP | = | = errore | = errore |
| CMDCHGPAR | = errore | = errore | = errore | = errore | = errore | = errore |
| CMDGETPAR | = | = | = | = | = | = errore |
| CMDGETALARM | = | = | = | = | = | = errore |
| CMDGETPOS | = | = | = | = | = | = errore |
| CMDGETPOST | = | = | = | = | = | = errore |
| CMDGETVEL | = | = | = | = | = | = errore |
| CMDGETTOR | = | = | = | = | = | = errore |
| CMDGETVER | = | = | = | = | = | = errore |
| CMDGETSMSTAT | = | = | = | = | = | = errore |
| CMDSAMPLE | = | = | = | = | = | = errore |
| CMDGETSAMP | = | = | = | = | = | = errore |
| CMDCHGPARN * | = errore | = errore | = errore | = errore | = errore | = errore |
| CMDGETPARN * | = | = | = | = | = errore | = errore |
| CMDTRAJVEL * | = errore | = errore | = (5) | = errore | = errore | = errore |
| CMDSETOVERR * | = | = | = | = | = errore | = errore |
| CMDGETOVERR * | = | = | = | = | = errore | = errore |
| CMDJOGN * | = errore | = (2) | = errore | = errore | = errore | = errore |
| CMDSAVEPARFL * | = errore | = errore | = errore | = errore | = errore | = errore |
| CMDERASEFIR * | = errore | = errore | = errore | = errore | = errore | = errore |
| CMDCHGBOOT * | = errore | = errore | = errore | = errore | = errore | = errore |
| CMDGOTOBOOT * | = errore | = errore | = errore | = errore | = errore | = errore |
| CMDGOINTERP * | = errore | = errore | = errore | = errore | = errore | = errore |
| CMDSTOPINTERP* | = errore | = errore | = errore | = errore | AXSTOP | AXSTOP |
| CMDLATCHINTER* | = errore | = errore | = errore | = errore | AXLATCH | = errore |

Note:

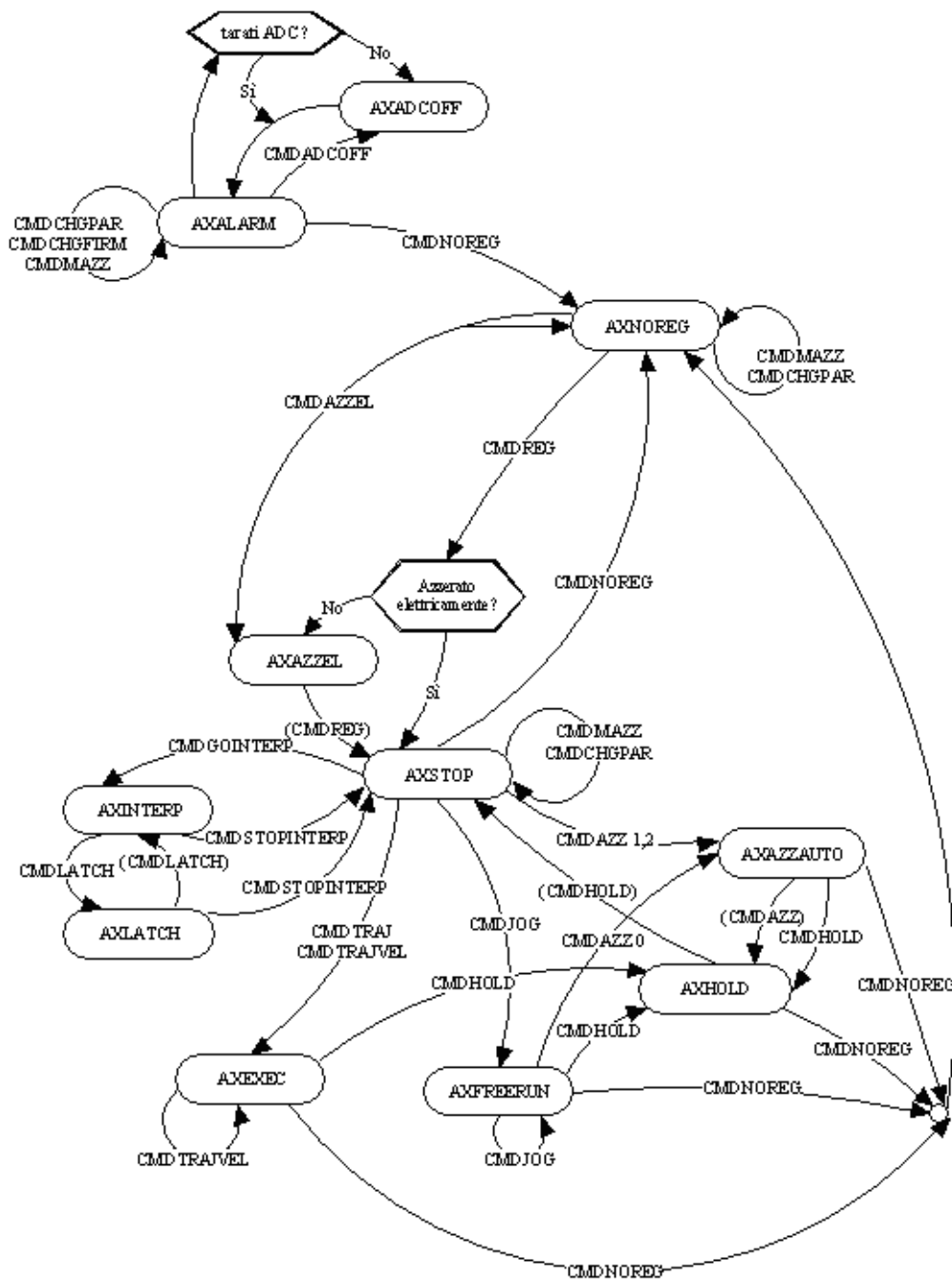
- (1) Se il motore non ha ancora effettuato un azzeramento elettrico, prima di portarsi nello stato **AXSTOP** si pone nello stato **AXAZZEL** il tempo necessario a concludere l'azzeramento elettrico.
- (2) Se necessario, viene modificata la velocità di JOG con delle rampe tra una velocità e l'altra
- (3) Lo stato diventa **AXEXEC** solo se il motore è stato azzerato in modo manuale o in modo automatico.

B Stati interni

- (4) Questi comandi sono legati rigidamente allo stato della comunicazione. Contattare il fornitore per maggiori chiarimenti
- (5) Se necessario, viene modificata la velocità di esecuzione del movimento mediante delle rampe tra una velocità e l'altra. Inoltre, se necessario, viene modificata la quota obiettivo del movimento
- (6) Questi stati sono presenti sull'SM140 a partire dalla versione 119

Se la modifica alla velocità o alla quota non è effettuabile il motore invia una "risposta a comando non valido" **CmdNACK** (Appendice F).

Transizioni di stato:



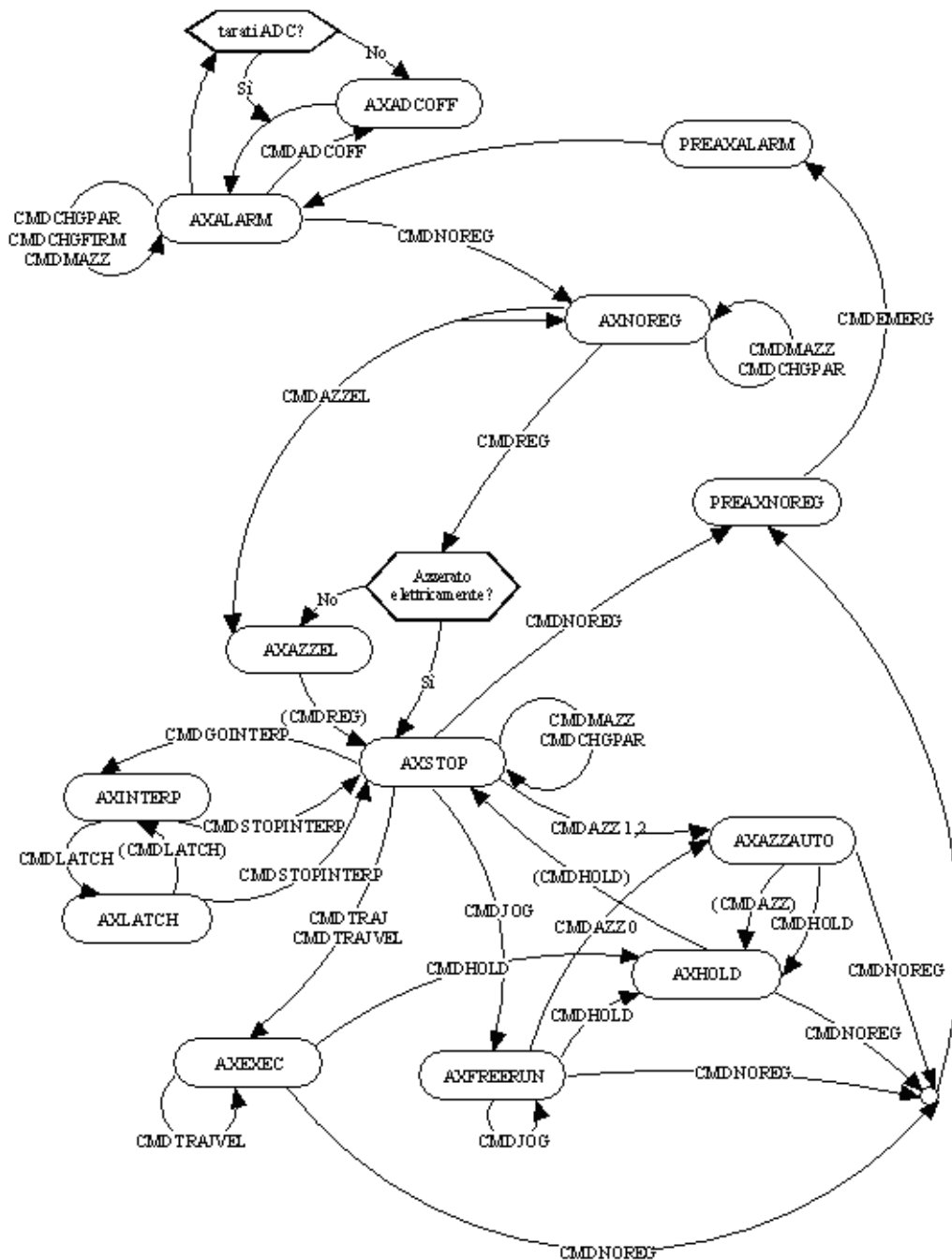
Note:

A fianco del comando **CMDAZZ** può essere presente un numero; tale numero rappresenta il valore del parametro **TIPOAZZ** che permette la transizione di stato disegnata.

Le transizioni di stato legate alle camme di extracorsa non sono rappresentate.

Per uscire dallo stato di allarme è consigliabile l'uso del comando **CMDRESET** (Sezione 3.2).

Se nel motore SM140 non è stata disabilitata la possibilità di frenare in modo controllato, ogni qual volta il motore esce dalla regolazione, l'automa a stati diventa quello riportato di seguito:



Note:

Si può notare la presenza di 2 nuovi stati che hanno il compito di portare il motore a fermarsi effettuando una rampa di decelerazione pari a quella di Jog.

Quando il motore è fermo si porta nello stato **AXNOREG** o **AXALARM** a seconda dei casi.

Per uscire dallo stato di allarme è consigliabile l'uso del comando **CMDRESET** (Sezione 3.2).

C Stato dell'azzeramento

Mediante il comando `CmdGetStatAzz` è possibile richiedere all'azionamento lo stato dell'azzeramento in corso.

Nella seguente tabella sono riportati gli stati possibili ritornati che dipendono dal tipo di azzeramento automatico in corso, impostato mediante il parametro `TIPOAZZ (0x012F)`:

| Codice | Nome | Significato | Tipo (TIPOAZZ) |
|--------|----------------|---|----------------|
| 0x0000 | NOAZZ | Il motore non è stato azzerato oppure l'ultimo azzeramento effettuato non è stato concluso | 0 - 1 - 2 |
| 0x0001 | AZZMAN | Il motore è stato azzerato in modo manuale | 0 - 1 - 2 |
| 0x0002 | SEARCHINGMICRO | Il motore sta effettuando l'azzeramento automatico ed è nella fase di ricerca del microinterruttore di azzeramento | 1 - 2 |
| 0x0003 | AZZAUTO | Il motore è stato azzerato in modo automatic | 0 - 1 - 2 |
| 0x0004 | LEAVINGMICRO | Il motore sta effettuando l'azzeramento automatico, ed è in fase di rilascio del microinterruttore di azzeramento. | 1 - 2 |
| 0x0006 | SARCHINGTACCA | Il motore è in fase di ricerca della tacca di zero dell'encoder per concludere la procedura di azzeramento automatico | 0 - 1 |

D Messaggi

L'azionamento può produrre 2 tipi di messaggi:

1. ALLARMI: sono messaggi relativi a condizioni di errore gravi che portano il motore nello stato di EMERGENZA
2. WARNING: sono messaggi relativi a condizioni di errore non gravi che non portano il motore nello stato di EMERGENZA

Tali messaggi possono essere recuperati utilizzando il comando di `CmdGetAlarm` (Sezione 4.15).

D.1 Messaggi di ALLARME

Nella seguente tabella sono riportati i messaggi di ALLARME e gli eventuali interventi di ripristino delle condizioni corrette di lavoro del motore.

| Codice | Nome | Significato | Interventi |
|--------|--------------|---|---|
| 0x0000 | NOALARM | Il motore sta funzionando regolarmente | Nessuno |
| 0x0001 | ALOVERHEATED | L'azionamento ha superato la temperatura di 70°C | Verificare l'effettivo carico meccanico del motore e/o il tempo di riposo rispetto a quello di lavoro |
| 0x0002 | ALOVERCURR | La parte di potenza dell'azionamento è stata percorsa da una corrente eccessiva | Possibile corto circuito sulla parte di potenza: contattare il fornitore |
| 0x0003 | ALOVERLOAD | La coppia necessaria al motore per muoversi è stata superiore alla coppia massima per un tempo troppo alto | Aumentare il valore del parametro <code>MAXTORQUE</code> e corrispondentemente anche dei parametri <code>KVELLIMAX</code> e <code>KVELLIMIN</code> |
| 0x0004 | ALGENTRAIETT | Errore legato al generatore di traiettoria interno al motore | Contattare il fornitore |
| 0x0005 | ALOVERLIM | Il motore si trova oltre uno dei limiti di finecorsa software. Questo errore può occorrere quando il controllo dei limiti di finecorsa software è abilitato e l'asse è stato azzerato | Verificare se la causa è determinata da: <ul style="list-style-type: none"> ■ sovralongazioni a fine movimento ■ Jog oltre i limiti ■ carico esterno al motore |
| 0x0006 | ALMAXERRORP | Il motore ha avuto un errore di inseguimento in posizione, superiore a quello impostato nei parametri | Verificare la taratura dell'asse o l'eventuale presenza di ostacoli meccanici. Aumentare il valore di soglia dell'errore di inseguimento in posizione |
| 0x0007 | ALMAXERRORV | Il motore ha avuto un errore di inseguimento in velocità, superiore a quello impostato nei parametri | Verificare la taratura dell'asse o l'eventuale presenza di ostacoli meccanici. Aumentare il valore di soglia dell'errore di inseguimento in velocità |

| Codice | Nome | Significato | Interventi |
|--------|------------------------------|---|--|
| 0x0008 | ALCOMERROR | Errore di comunicazione tra il motore e il master (1) | Verificare i cablaggi. Se il parametro TIMEOUTFB (0x012D) è diverso da 0, aumentarlo. Verificare che il master non sia stato spento mentre il motore era alimentato |
| 0x0009 | ALNOPOWER | L'alimentazione di potenza è scesa sotto il valore di 16 V mentre il motore era in regolazione. | Verificare che la macchina non sia andata in emergenza mentre il motore era in regolazione. Verificare la potenza dell'alimentatore |
| 0x000A | ALNOPARAMINFL (*) | I parametri salvati in flash memory non sono corretti | Riinvia i parametri al motore e salvarli nella flash memory per non far comparire di nuovo il messaggio all'accensione |
| 0x000B | ALNOPRGINFLASH (*) | Il software applicativo salvato in flash memory non è corretto. | Riinvia al motore il software applicativo |
| 0x000C | ALERASINGFL (*) | Durante l'invio al motore del software applicativo si sono verificati dei problemi nella cancellazione della flash memory | Riinvia al motore il software applicativo |
| 0x000D | ALPRGMINGFL (*) | Durante l'invio al motore del software applicativo si sono verificati dei problemi nella scrittura della flash memory | Riinvia al motore il software applicativo |
| 0x000E | ALWRONGDATA2 INTERP (*) | Durante l'interpolazione il motore ha ricevuto un pacchetto dati con codifica errata | Verificare la presenza di errori di comunicazione. Verificare il software di comunicazione del master che pilota il motore e/o il cablaggio del cavo |
| 0x000F | ALWROGSETP (*) | Durante l'interpolazione è stato fornito un setpoint di posizione che richiede una velocità troppo elevata al motore | Verificare il software di comunicazione del master che pilota il motore |
| 0x0010 | ALNOSETP2 INTERP (*) | Durante l'interpolazione non è arrivato al motore il setpoint di posizione nel tempo previsto | <ul style="list-style-type: none"> ■ Verificare che il parametro che dichiara la frequenza del bus sia corretto ■ Verificare la presenza di errori di comunicazione ■ Verificare il software di comunicazione del master che pilota il motore e/o il cablaggio del cavo |
| 0x0011 | ALWRONGFREQ (*) | La frequenza del bus è diversa da quella dichiarata nel parametro del motore | Verificare che il parametro che dichiara la frequenza del bus sia corretto |
| 0x0012 | ALNOTIPICPARAM | All'accensione non sono stati rilevati alcuni parametri necessari al corretto funzionamento del motore | Contattare la CNI |

| Codice | Nome | Significato | Interventi |
|--------|------------------|--|---|
| 0x0013 | ALTIMEOUTCOMUNIC | Errore di comunicazione nel protocollo di colloquio con il motore | Verificare la connessione e l'integrità del cavo di collegamento ed eventualmente aumentare il parametro 0x012 (D-TIMEOUTFB). |
| 0x0014 | ALOVERPOWER | La tensione dell'alimentazione di potenza è salita al di sopra di 38 V | Diminuire l'inerzia riportata al rotore o il dato di accelerazione massima oppure utilizzare un freno elettrico (modulo P144 fornito da CNI). |
| 0x0020 | ALDSPOVERLOAD | Errore software interno all'azionamento | Contattare il fornitore |
| 0x0021 | ALWRONGIRQ | Errore software interno all'azionamento | Contattare il fornitore |

Legenda:

- (1) L'errore di comunicazione nel caso di protocollo seriale 485 può essere causato da:
 - Timeout: il master non ha più inviato dati
 - Arrivato un pacchetto di caratteri troppo lungo (o privo del carattere ETX)
 - Arrivati 2 caratteri STX senza ETX
 - CRC sbagliato
 - Arrivato carattere ESC seguito da carattere sbagliato (ETX, STX o ESC)
- (*) Errori attivi sul motore SM140 e solo a partire dalla revisione C del motore SM137

D.2 Messaggi di WARNING

Nella seguente tabella sono riportati i messaggi di WARNING e gli eventuali interventi di ripristino delle condizioni corrette di lavoro del motore.

| Codice | Nome | Significato | Interventi |
|--------|------------------|--|--|
| 0x4000 | ALNOAZZ | Si è cercato di eseguire un movimento quotato con asse scalibrato | Effettuare un azzeramento del asse prima del movimento quotato |
| 0x4001 | ALPARNONCORR | È stato inviato o letto un parametro inesistente | Verificare il software del master |
| 0x4002 | ALCMDLOOSED | Errore software interno all'azionamento | Contattare il fornitore |
| 0x4003 | ALWRONGCMD | È stato inviato un comando inesistente | Contattare il fornitore |
| 0x4004 | ALAXALREADYINPOS | Si è cercato di effettuare un movimento quotato alla quota in cui è posizionato il motore | Verificare il software del master |
| 0x4005 | ALREQPOSOVERLIM | È stato comandato un movimento quotato al di fuori dei limiti di finecorsa software | Verificare i limiti di finecorsa software o verificare il software del master |
| 0x4006 | ALNOTPOT (*) | Il motore è giunto al limite di finecorsa software superiore o inferiore | Verificare i limiti di finecorsa software impostati nei parametri o i comandi inviati al motore |
| 0x4007 | ALFLNOTERASED* | È stato richiesto di scrivere in flash memory su una zona non cancellata | Contattare il fornitore |
| 0x4008 | ALFLREADING* | Errore durante la scrittura su flash memory | Contattare il fornitore |
| 0x4009 | ALNOTHW_M(1) | Il motore ha incontrato la camma di extracorsa negativa | Muovere il motore nel verso delle quote crescenti oppure disabilitare il controllo della camma di extracorsa negativa |
| 0x400a | ALPOTHW_M(1) | Il motore ha incontrato la camma di extracorsa positiva | Muovere il motore nel verso delle quote decrescenti oppure disabilitare il controllo della camma di extracorsa positiva |
| 0x400b | ALMOVTOOLONG(2) | È stato comandato al motore un movimento quotato che implica uno spostamento troppo grande (maggiore di 67108863 conteggi encoder) | Spezzare il movimento comandato in più movimenti quotati senza fermata, ognuno di ampiezza inferiore all'ampiezza massima consentita |

Legenda:

- (*) Messaggio presente sul motore SM140 e solo a partire dalla revisione C del motore SM137 (versioni del firmware uguali o superiori alla 110)
- (1) Messaggio presente sull'SM140
- (2) A partire dalla versione 119 del firmware

E BitStato

Nel pacchetto di risposta ai comandi sono sempre presenti 4 bit denominati "BitStato" il cui significato è riportato nella tabella seguente.

| Bit | Nome | Risposta |
|-----|--------------------------------|--|
| 3 | <code>MASKBITALLARME_SM</code> | 1: il motore è nello stato di allarme |
| 2 | <code>MASKBITWARNING_SM</code> | 1: sull'azionamento è presente un messaggio per il master |
| 1 | <code>MASKBITCOMANDO_SM</code> | 0: l'ultimo comando inviato al motore deve essere ancora completato. Può accadere, per esempio, per il comando di azzeramento automatico e per il comando line |
| 0 | <code>MASKBITNOQUOTA_SM</code> | 1: la risposta al comando default è l'invio della quota teorica/velocità teorica/coppia teorica |

F Risposta a comando non valido

Quando il motore non può eseguire un comando la risposta fornita è il pacchetto di dati riportato nella seguente tabella:

| Byte | Risposta |
|------|-----------------------------|
| 01 | STX (0x02) |
| 02 | Nodo |
| 03 | 0x0 BitStato |
| 04 | CmdNACK (0xB0) |
| 05 | Byte 05 del comando inviato |
| 06 | Byte 06 del comando inviato |
| 07 | Byte 07 del comando inviato |
| 08 | CRC |
| 09 | ETX (0x03) |

G Ingressi digitali

G.1 Introduzione

Il motore SM140 è fornito di 3 ingressi digitali il cui livello logico può essere variato mediante il parametro **INVDIN** (0x012E) come riportato nelle seguenti tabelle.

| INVDIN | Ingresso digitale corrispondente | Descrizione |
|--------|----------------------------------|--|
| Bit 0 | 1 | Camma di extracorsa negativa |
| Bit 1 | 2 | Camma di extracorsa positiva |
| Bit 2 | 3 | Microinterruttore per l'azzeramento automatico |

| Valore Bit n di INVDIN | Tensione in ingresso (V) | Livello logico del segnale |
|------------------------|--------------------------|----------------------------|
| 0 | 1 | Basso |
| 0 | 24 | Alto |
| 1 | 0 | Alto |
| 1 | 24 | Basso |

Quindi: se il bit 2 di **INVDIN** vale 1 e la tensione in ingresso sul terzo ingresso digitale è 0 il livello logico del microinterruttore sarà Alto.

G.2 Camme di extracorsa

Sono gestite due camme di extracorsa: una positiva e una negativa.

La "camma di extracorsa positiva" è la camma che si incontra quando il motore aumenta la sua quota reale, la "camma di extracorsa negativa" è la camma che si incontra quando il motore diminuisce la sua quota reale, indipendentemente dal verso di rotazione del motore che può essere modificato mediante il bit 3 del parametro **BIT_A** (0x011E).

Se la gestione delle camme di extracorsa è stata abilitata (bit 8 e 9 del parametro **BIT_A**) e il motore durante il moto incontra una camma, in base al valore del bit 10 del parametro **BIT_A** si porta nello stato di allarme (**AXALARM**) oppure effettua una frenata tornando in regolazione (**AXSTOP**).

Oltre a questo, il motore fornisce un messaggio nel quale indica quale delle due camme è stata incontrata.

A questo punto il motore permette solo movimenti che consentono di uscire dalla camma incontrata.

La logica dei segnali di extracorsa di default è invertita per motivi di sicurezza; infatti, il motore vede la camma di extracorsa quando il relativo segnale logico è basso. Come già illustrato, è possibile invertire il livello logico del segnale attraverso il parametro **INVDIN** (0x012E) ma in questo caso il livello di sicurezza dell'extracorsa diminuisce a rischio e pericolo dell'utilizzatore.

G.3 Microinterruttore di azzeramento

Per poter effettuare l'azzeramento automatico in maniera indipendente (parametro **TIPOAZZ=1** o **TIPOAZZ=2**) il motore utilizza il terzo ingresso digitale che deve essere collegato al microinterruttore di azzeramento. Il motore è sul microinterruttore di azzeramento quando il relativo segnale logico è alto.

È possibile invertire il livello logico del segnale come già illustrato in precedenza.

Per una descrizione dei vari tipi di azzeramento e la modalità con cui viene utilizzato l'ingresso digitale in quelle condizioni, si veda il comando di azzeramento automatico **CmdAzz** (Sezione 3.7).

H Interpolazione

Attualmente l'interpolazione è disponibile solo su bus Enet-X.

Il byte contenente i bit significativi per il protocollo è descritto nelle tabelle successive.

Bit significativi per il protocollo da Master a Slave

| Significato | Bit 7 | Bit 6 | Bit 5 | Bit 4 | |
|-----------------------------------|-------|-------|-------|-------|----|
| Invio pacchetto vuoto | 0 | 0 | 0 | 0 | M0 |
| Invio pacchetto singolo | 1 | 1 | 1 | x | M1 |
| Inizia invio pacchetto multiplo | 0 | 1 | 1 | x | M2 |
| Continua invio pacchetto multiplo | 0 | 0 | 1 | x | M3 |
| Termina invio pacchetto multiplo | 1 | 0 | 1 | x | M4 |
| Interpolazione: invio setpoint | 1 | 0 | 0 | 0 | M5 |
| Interpolazione: richiesto latch | 1 | 1 | 0 | 0 | M6 |
| Non usato | 0 | 1 | 0 | x | |

Bit significativi per il protocollo da Slave a Master

| Significato | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 2 | Bit 1 | |
|---|-------|-------|-------|-------|-------|-------|----|
| Invio pacchetto di default | 1 | 1 | 0 | 0 | x | x | S0 |
| Invio pacchetto singolo | 1 | 1 | 1 | x | x | x | S1 |
| Inizia invio pacchetto multiplo | 0 | 1 | 1 | x | x | x | S2 |
| Continua invio pacchetto multiplo | 0 | 0 | 1 | x | x | x | S3 |
| Termina invio pacchetto multiplo | 1 | 0 | 1 | x | x | x | S4 |
| Slave in interpolazione (SM140: ingresso microinterruttore alto) | 1 | 0 | 0 | 0 | 1 | 0 | S5 |
| Slave in interpolazione (SM140: ingresso microinterruttore basso) | 1 | 0 | 0 | 0 | 0 | 1 | S5 |
| Interpolazione: quota latchata su latch | 1 | 0 | 0 | 0 | x | 1 | S6 |
| Non Usato | 0 | 0 | 0 | x | x | x | |
| Non Usato | 0 | 1 | 0 | x | x | x | |

Quando il master vuole inviare un comando di interpolazione deve porre il protocollo nello stato M5 iniziando immediatamente a inviare il primo setpoint. Lo slave confermerà lo stato di interpolazione rispondendo con lo stato S5 e inviando il dato di default richiesto (normalmente la quota reale).

Se il master desidera effettuare un latch, deve inviare una sola volta il frame M6. Lo slave si pone immediatamente alla ricerca della tacca di zero dell'encoder e la quota reale latchata è ritornata insieme al frame S6.

C ASSISTENZA**HSD** S.p.A.**TECHNOLOGICAL EQUIPMENT FOR AUTOMATION****REGISTERED OFFICE**

Via della Meccanica 16
61122 PESARO (ITALIA)
Loc. Chiusa di Ginestreto

FACTORY HEADQUARTERS

Piazzale Alfio De Simoni s.n.
61122 PESARO (ITALIA)
Phone no. (+39)0721.205.211
Fax (+39)0721.205.247
E-mail supporthsd@hsd.it
Web www.hsd.it

HSD Deutschland GmbH

Brückenstrasse 32
D-73037 Göppingen (DEUTSCHLAND)
Phone no. (+49)07161.956.660
Fax (+49)07161.956.6610
E-mail supporthsddeut@hsddeutschland.de
Web www.hsddeutschland.de

HSD USA Inc.

3764 SW, 30th Avenue
33312 Fort Lauderdale FL, USA
Phone no. (+1) 954.587.1991
Fax (+1) 954.587.8338
E-mail supporthsdusa@hsd.it
Web www.hsdusa.com

HSD Mechatronic Shanghai Co. Ltd.

D2, First Floor, 207 Taigu Road
Waigaoqiao Free Trade Zone
200131, Shanghai - China
Phone no. (+86) 215.866.1236
E-mail sales@hsd-china.cn
Web www.hsd-china.cn

HSD S.p.A.

Registered Office:
Via della Meccanica, 16
61122 Pesaro (PU) Italy

Factory Headquarters:
Piazzale Alfio De simoni, sn
61122 Pesaro (PU) Italy
Phone n.: (+39) 0721.205.211
Fax: (+39) 0721.205.247
C.F. Nr. 02196600965
P.I. Nr. IT 01376450415