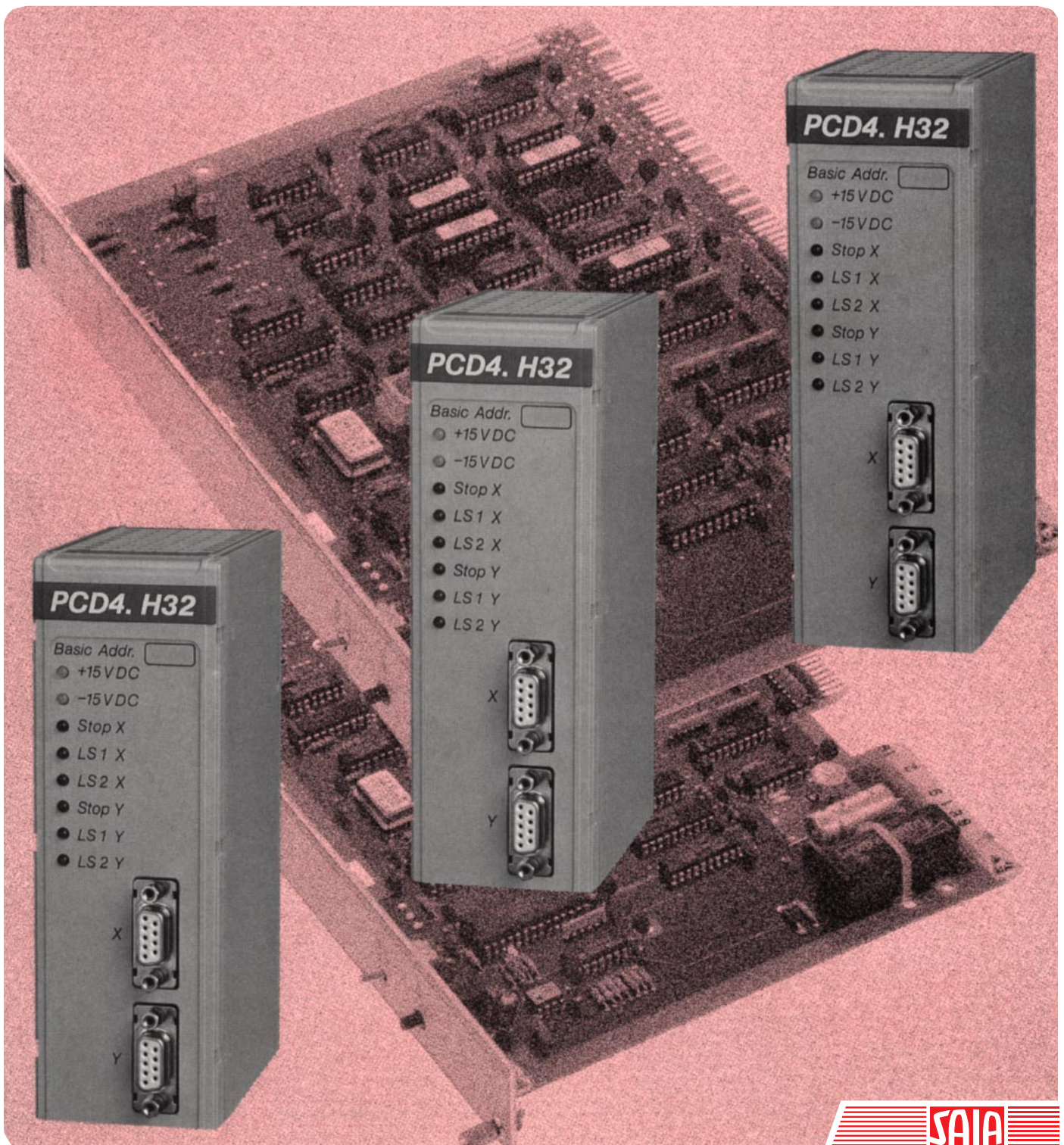


**SAIA®PCD**  
Process Control Devices

**Manuale**  
**Modulo di controllo per**  
**servomotori PCD4.H3..**



**SAIA® Process Control Devices**

**Manuale**

**Modulo di controllo per servomotori**

**PCD4.H3..**

**Edizione 26/729 I1 - 05.97**

SAIA-Burgess Electronics SA 1997 - Tutti i diritti riservati

Soggetto a modifiche senza preavviso

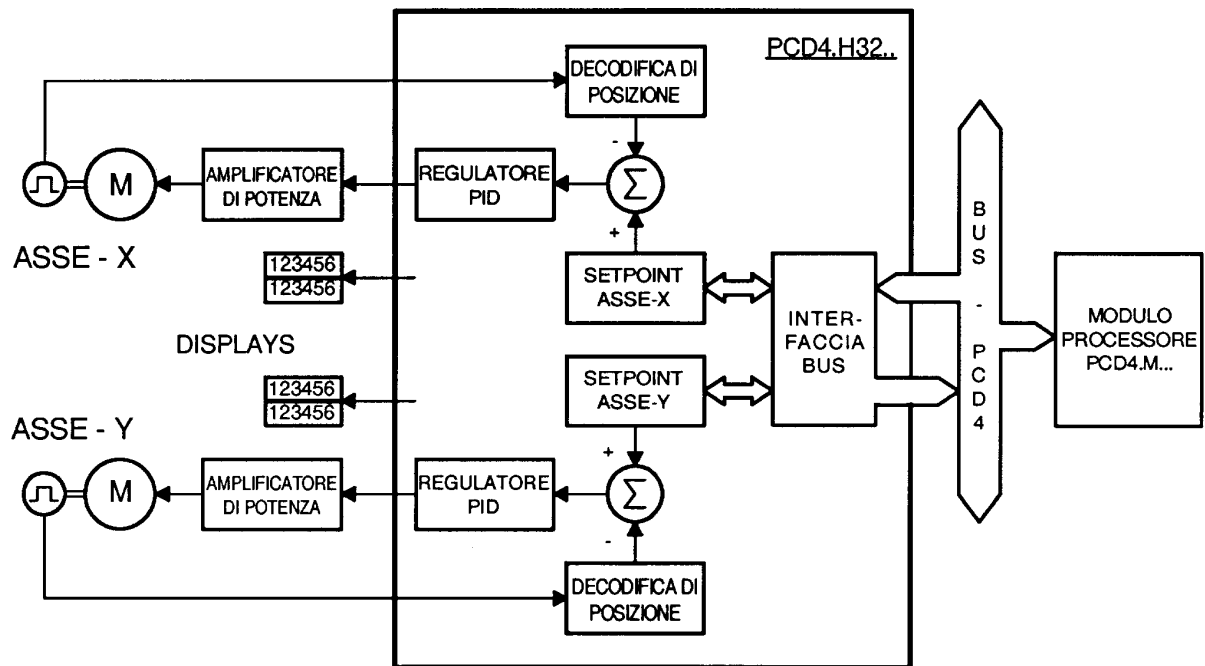
# Indice

---

<b>1.</b>	<b>Introduzione</b>		
<b>2.</b>	<b>Dati tecnici</b>		
<b>3.</b>	<b>Presentazione</b>		
3.1	Circuito stampato	pag.	3-1
3.2	Pannello frontale	pag.	3-3
<b>4.</b>	<b>Schema logico</b>		
<b>5.</b>	<b>Connettori e indirizzamento</b>		
5.1	Connettori	pag.	5-1
5.2	Indirizzamento	pag.	5-3
<b>6.</b>	<b>Funzionamento</b>		
6.1	Modi di Funzionamento	pag.	6-1
6.2	Generatore del profilo di velocità	pag.	6-2
6.3	Regolatore PID	pag.	6-4
6.4	Decodifica di posizione e circuito di ingresso	pag.	6-5
6.5	Convertitore D/A (uscita analogica)	pag.	6-11
6.6	Generatore PWM	pag.	6-12
<b>7.</b>	<b>Scrittura dei programmi per il modulo H3</b>		
7.1	Installazione software	pag.	7-1
7.2	Blocchi funzione principali: "AxInit" e "AxHndlg"	pag.	7-12
7.3.	Elenco dei comandi	pag.	7-20
7.4	Descrizione dei comandi	pag.	7-23
<b>8.</b>	<b>Riconoscimento e Gestione errori</b>		
<b>9.</b>	<b>Esempi</b>		
9.1	Esempio 1	pag.	9-1
9.2	Esempio 2	pag.	9-10
9.3	Esempio 3	pag.	9-16
<b>10.</b>	<b>Riassunto dei comandi e dei simboli</b>		

# 1. Introduzione

Schema a blocchi :



## Funzione e Utilizzo:

Il modulo PCD4.H3.. può pilotare uno o due assi indipendenti che utilizzano motori a velocità variabile (servomotori). Questi ultimi sono motori a CC o a CA dotati di pilotaggio di potenza e di trasduttori incrementali della posizione dell'albero, che consentono di controllarne la posizione e la velocità.

Il modulo si connette al PCD4 tramite il bus PCD4 e utilizza 16 indirizzi. In teoria questo significa che ad un singolo sistema PCD4 si possono collegare fino a 16 moduli di controllo movimento (32 assi).

Per ogni asse il movimento è controllato da un microprocessore in base ai parametri caricati (velocità, accelerazione e posizione di destinazione). Ciascun asse è controllato in modo indipendente, quindi non è possibile eseguire interpolazioni per ottenere percorsi curvilinei. Si può tuttavia programmare il collegamento di più assi (punto-punto) per ottenere un funzionamento quasi-sincrono.

## Applicazioni tipiche:

- Gestione di robot
- Macchine per il montaggio automatico di componenti
- Macchine per trasporto automatico
- Macchine per imballaggio
- Macchine per la lavorazione delle lamiere

### **Programmazione:**

Viene fornita una libreria software che contiene dei Blocchi Funzione per il controllo del modulo. Tale libreria software è resa disponibile in formato sorgente PCD e quindi facilmente interpretabile.

La programmazione del modulo di controllo movimento risulta in questo modo facilitata, non richiedendo l'utilizzo di istruzioni complicate.

### **Caratteristiche salienti:**

- Posizione e Velocità sono controllate in modo PID.
- Velocità, posizione di destinazione e parametri PID possono essere modificati durante il movimento
- Stadio di potenza dell'azionamento controllato da un segnale Analogico  $\pm 10V$  o in PWM (PWM = Pulse Width Modulated: *Modulazione di Ampiezza di Impulsi*).
- Ingressi digitali a 24 Volt per microinterruttori di fine corsa e di riferimento (funzionamento in logica positiva).
- Ingressi di segnale da encoder a 24V (funzionamento in logica positiva o negativa) oppure RS422 a 5V (line driver).
- Uscite digitali per la connessione ai moduli display PCA2.D14 con 2 x 6 cifre per asse.

**Elenco dei moduli:**

Tipo	Assi	Uscita Controller	Segnali dell'Encoder
PCD4.H310 <sup>1)</sup>	X	± 10V	24V
PCD4.H320 <sup>1)</sup>	X , Y	± 10V	24V
PCD4.H311 <sup>1)</sup>	X , Y	± 10V	5V (RS422)
PCD4.H321 <sup>1)</sup>	X	± 10V	5V (RS422)
PCD4.H316 <sup>2)</sup>	X	PWM	24V
PCD4.H326 <sup>2)</sup>	X , Y	PWM	24V
PCD4.H317 <sup>2)</sup>	X	PWM	5V (RS422)
PCD4.H327 <sup>2)</sup>	X , Y	PWM	5V (RS422)

- 1) Moduli standard
- 2) Forniti su richiesta

## 2. Dati tecnici

---

<b>Controllo spostamento</b>	Incrementale : 2 segnali in quadratura e indice (e corrispondenti segnali negati)
Ingressi 24V	
Livello segnale	Livello Basso = 0..4V Livello Alto = 19..32V
Corrente di ingresso 24V	10mA
Modo di funzionamento	Logica positiva o negativa
Ingresso a 5V	Differenziale 5V ingressi RS422
Isolati	No
Frequenza massima	100kHz
<b>Ingressi digitali</b>	Per ogni asse: 2 fine corsa e 1 segnale di riferimento a 24V CC
Livello segnale	Livello Basso = 0..4V Livello Alto = 19..32V
Corrente di ingresso 24V	10mA
Modo di funzionamento	Logica positiva
Filtro di ingresso	100kHz
<b>Uscite digitali</b>	Per pilotaggio del modulo display PCA2.D14 PCA2.D14 (2*6 cifre per asse)
2 uscite	Dati e clock (condivisi dai due assi)
1 uscita per asse	Abilitazione (logica negativa: attivo basso)
Corrente di uscita Ia	1..100mA (senza protezione contro i corto circuiti) Carico resistivo minimo 240 Ohm a 24V CC

<b>Uscita del controller</b>	Per pilotare il Servo-Amplificatore
Uscita analogica	±10V (risoluzione 12 bit più bit di segno) Carico resistivo minimo 3kOhm Protezione contro i corto circuiti
Uscita PWM	<b>Puls-Width-Modulated</b> (risoluzione 8 bit)
	Segnali : SIGN e MAGNITUDE regolati per il controllo diretto di un driver a chopper
	Uscita a collettore aperto : I <sub>max.</sub> = 500mA U <sub>max.</sub> = 50V

**Modi di funzionamento**                      Controllo di posizione o di velocità

**Elementi che determinano la posizione**

Posizione	Unità di misura : mm Range : $[-2^{30} \dots + (2^{30} - 1)] / k * 10^{-3}$ k = f{risoluzione encoder e rapporto di trasmissione}
Velocità	Unità di misura : mm/s Range : $[-2^{30} \dots + (2^{30} - 1)] / k * 22348 * 10^{-6}$ k = f{risoluzione encoder e rapporto di trasmissione}
Accelerazione	Unità di misura : mm/s <sup>2</sup> Range : $[-2^{30} \dots + (2^{30} - 1)] / k * 76206 * 10^{-9}$ k = f{risoluzione encoder e rapporto di trasmissione}
Regolatore PID	Tempo di scansione : 341ms I fattori Proporzionale, Integrale e Derivativo sono programmabili. (Il tempo di lettura per il termine derivativo è programmabile separatamente).
Programmazione	Con Blocchi Funzione forniti in codice sorgente PCD



**Alimentazione**

Esterna (utente) + 24V CC ( 19V .. 32V ) ripple filtrato al 10%

Corrente di alimentazione

Esterna 24V :

per H310, 320, 316 e 326  $I_{\max} = 150\text{mA/asse} + \text{alimentazione encoder}$

per H311, 321, 317 e 327  $I_{\max} = 300\text{mA/asse}$   
(Corrente massima per alimentazione encoder a 5V: 300mA/asse)

Interna dal bus PCD4 +5V / 120mA per asse  
-15V / 5mA per asse (solo per H310, 311, 320 e 321)

**Condizioni operative**

Temperatura ambiente 0°C .. +50°C senza ventilazione

Resistenza ai disturbi 1kV , accoppiamento capacitivo secondo le specifiche IEC 801-4

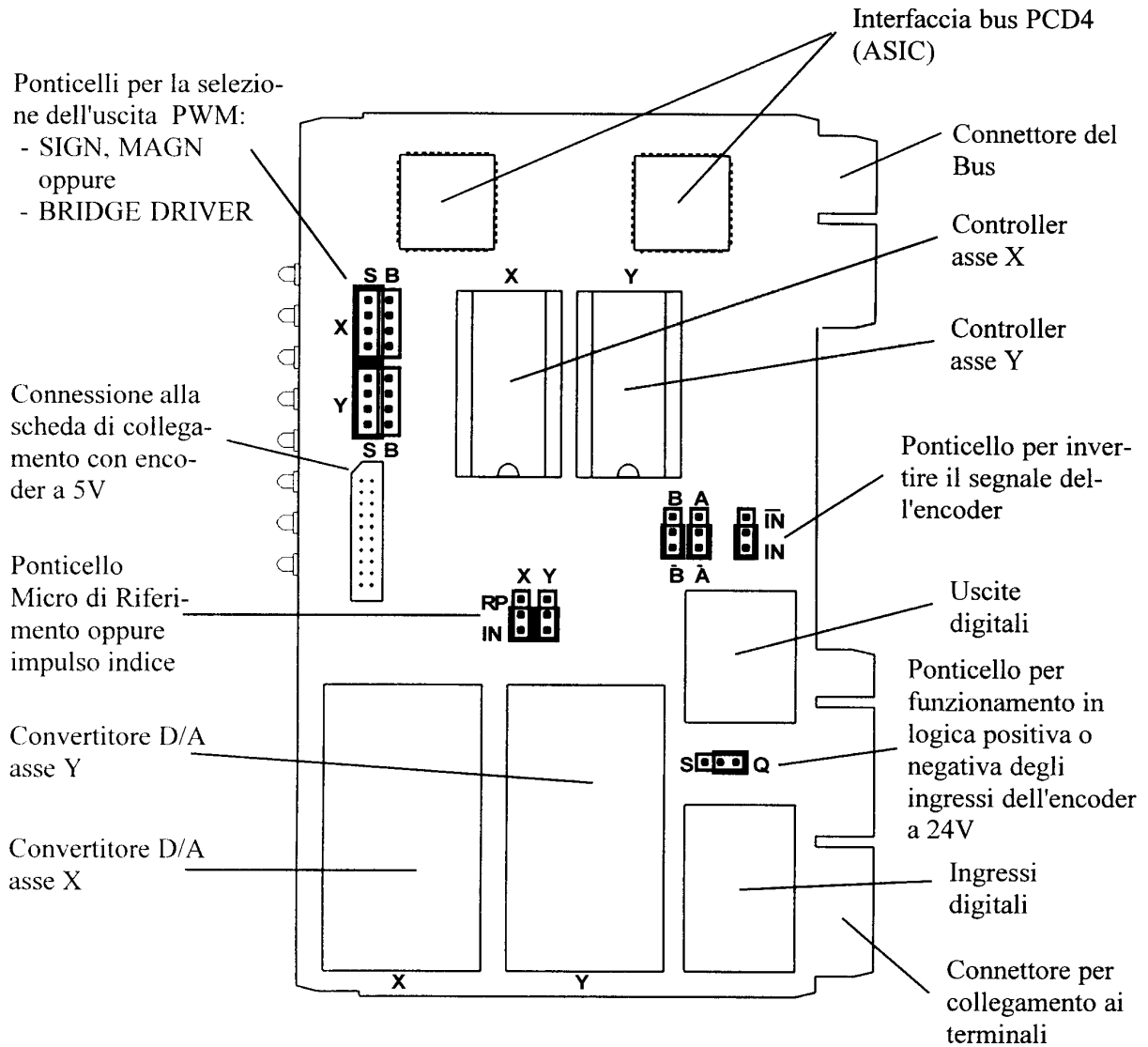
Resistenza meccanica Secondo le specifiche IEC 65A

Condizioni di Immagazzinamento Temperatura: -20°C.. +85°C  
Umidità: 0 .. 95%

## 3. Presentazione

### 3.1 Circuito stampato

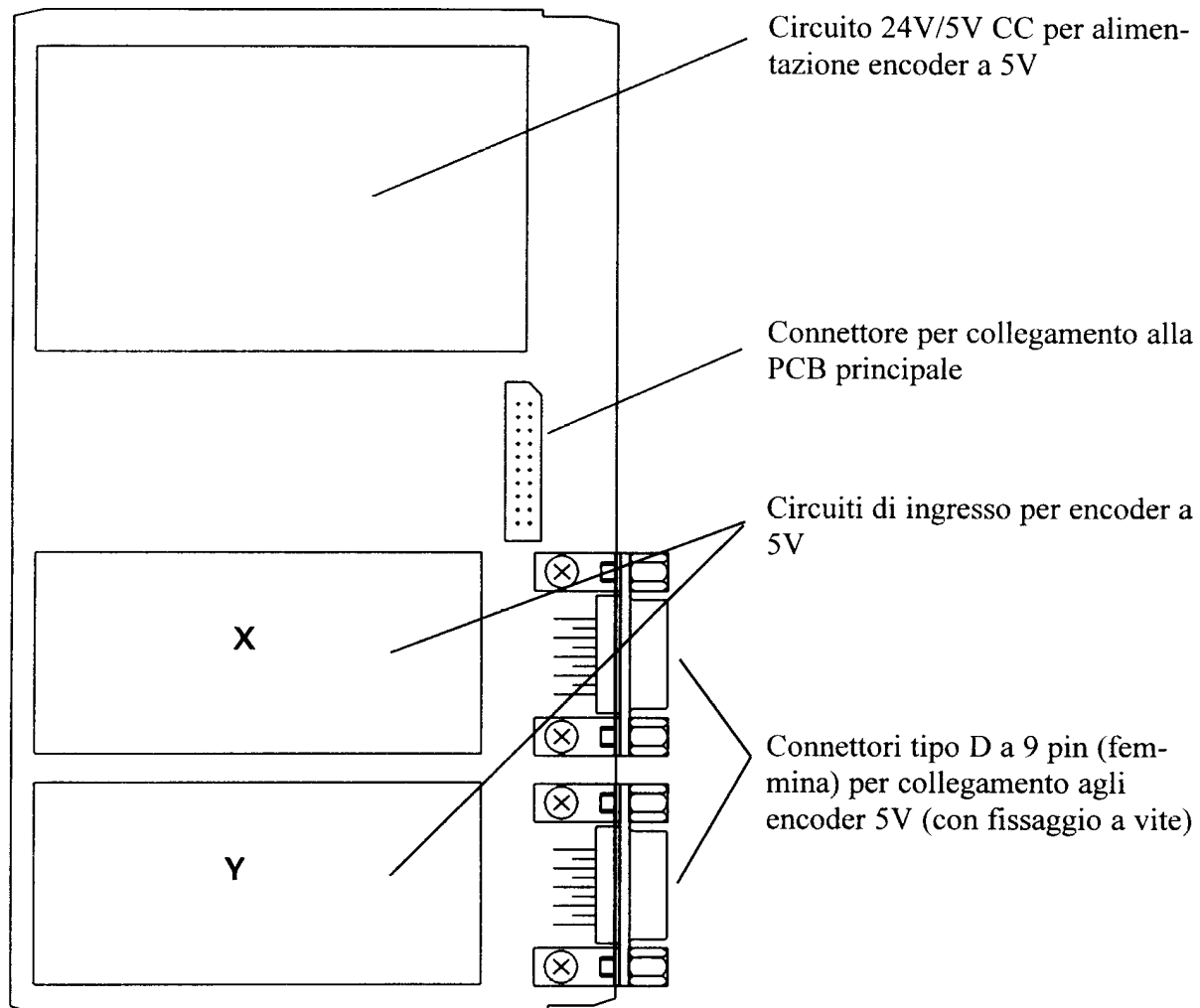
#### 3.1.1 PCB principale (2 assi)



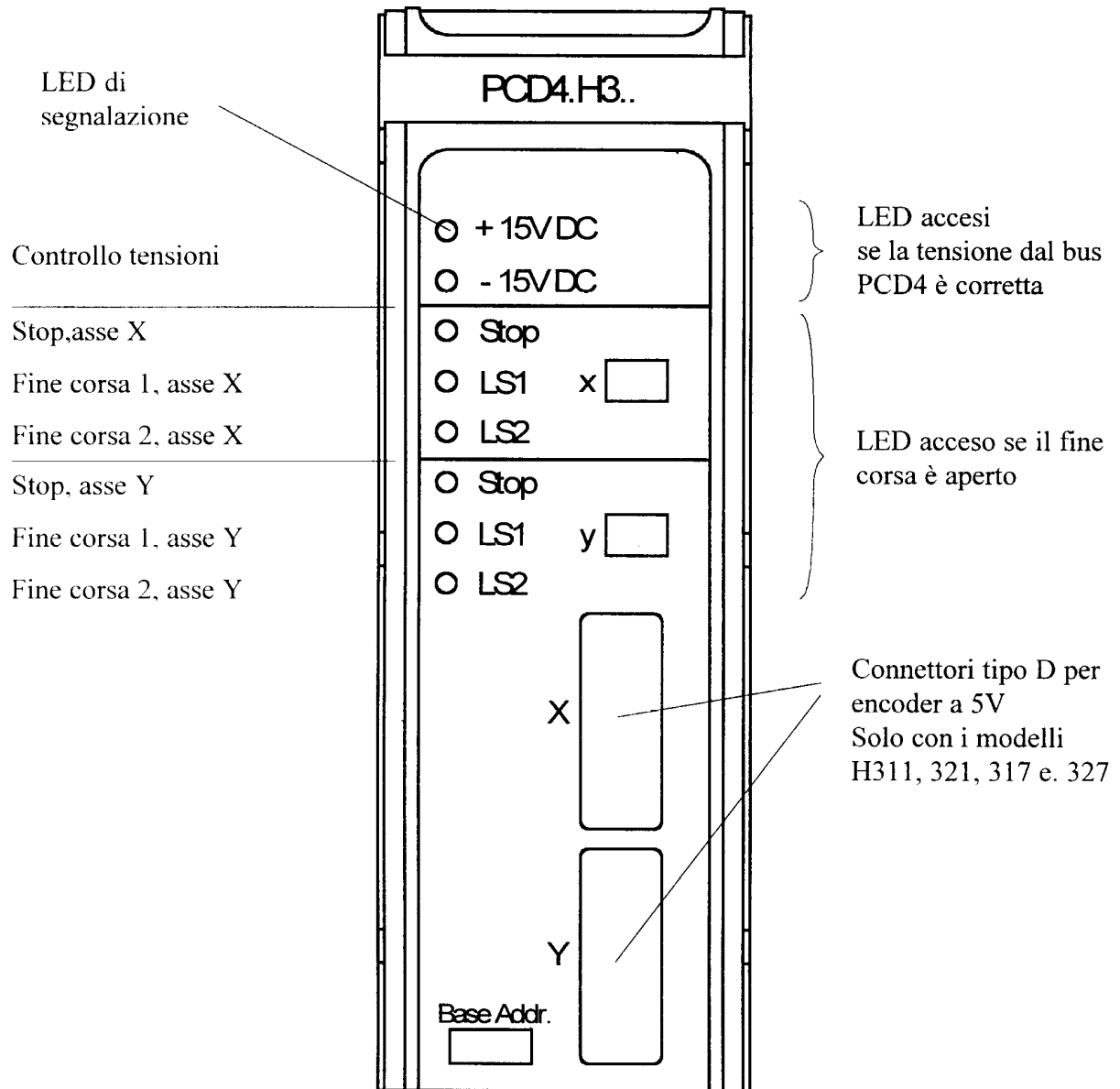
- I moduli del convertitore D/A sono presenti solo nei modelli H310, 320, 311 e 321

- I ponticelli per la selezione dell'uscita PWM sono presenti solo nei modelli H316, 317, 326 e 327

### 3.1.2 PCB aggiuntiva per encoder a 5V (2 assi)

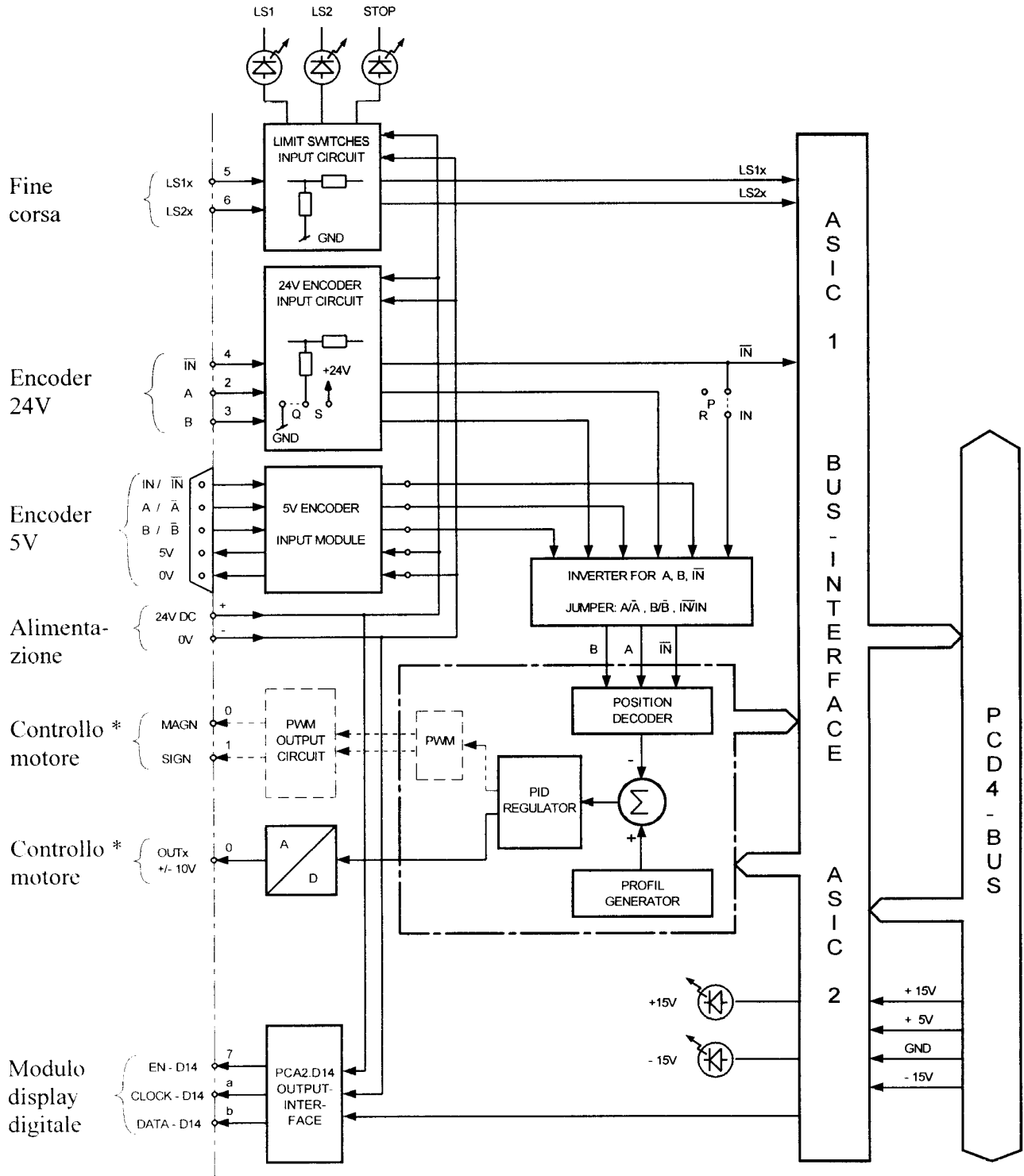


### 3.2 Pannello frontale



# 4. Schema Logico

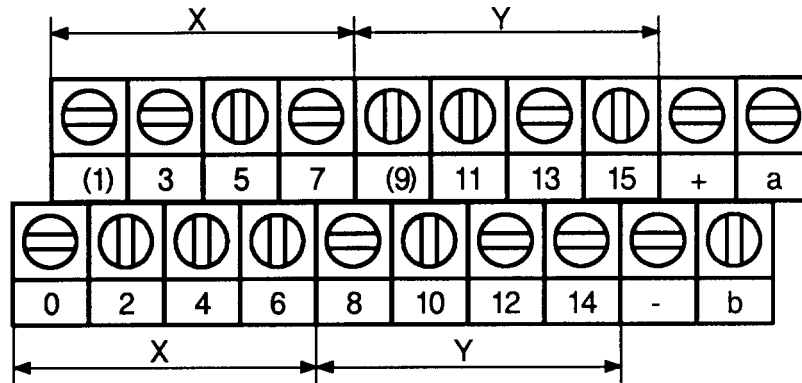
Nello schema è rappresentato solo l'asse X



\*) Uscita di controllo motore sia PWM che analogica  $\pm 10V$  (vedere l'elenco dei tipi di modelli).

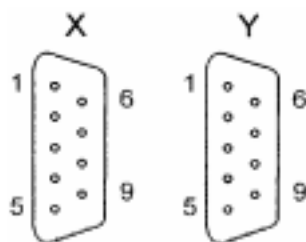
## 5. Connettori e indirizzamento

### 5.1 Connettori



Terminali della morsettiera:

0	Out-x	(PWM-MAG <sub>x</sub> )	8	Out-y	(PWM-MAG <sub>y</sub> )
1		(PWM-SIGN <sub>x</sub> )	9		(PWM-SIGN <sub>y</sub> )
2	Fase-Ax		10	Fase-Ay	
3	Fase-Bx		11	Fase-By	
4	/IN <sub>x</sub> (Rif. x)		12	/IN <sub>y</sub> (Rif. y)	
5	LS1 <sub>x</sub>		13	LS1 <sub>y</sub>	
6	LS2 <sub>x</sub>		14	LS2 <sub>y</sub>	
7	EN-D14 <sub>x</sub>		15	EN-D14 <sub>y</sub>	
-	GND		+	+24V	
a	CLK-D14		b	DATA-D14	



Connettori tipo D :  
(femmina)

1	PGND	PGND
2	Ax	Ay
3	/Ax	/Ay
4	Bx	By
5	GND	GND
6	/Bx	/By
7	IN <sub>x</sub>	IN <sub>y</sub>
8	/IN <sub>x</sub>	/IN <sub>y</sub>
9	+5V	+5V

Legenda:

Out-x, y	Uscita del controller, rispettivamente $\pm 10V$ e PWM-MAGN
PWM-SIGN	Uscita
Fase-A	Ingresso encoder per la Fase A (24V)
Fase-B	Ingresso encoder per la Fase B (24V)
/IN (Rif.)	Ingresso encoder per segnale indice o micro di riferimento
LS1	Ingresso fine corsa (24V)
LS2	" " " "
EN-D14	Uscita ENABLE per modulo display PCA2.D14
CLK-D14	Uscita CLOCK " " " "
DATA-D14	Uscita DATA " " " "
+24V	Alimentazione +24V CC
+5V	Uscita 5V per alimentazione encoder RS422
GND	Collegamento di massa per alimentazione 24VCC e 5VDC
PGND	Connessione alla terra di protezione per encoder a 5V. Non deve essere collegato, dal momento che la calza del cavo è collegata alla massa attraverso la struttura metallica del connettore tipo D e la vite di fissaggio. Vedere anche sezione 6.4.

## 5.2 Indirizzamento

Il modulo occupa 16 indirizzi del bus PCD. Dal momento che il modulo ha due interfacce verso il bus (ASIC), ciascuno dei 16 indirizzi ha due significati.

Significato dei 16 indirizzi:

	DATA IN :		DATA OUT:
	0 Data bus (LSB) <sup>(1)</sup>		0 Data bus (LSB) <sup>(1)</sup>
	1 " "		1 " "
	2 " "		2 " "
	3 " "		3 " "
	4 " "		4 " "
	5 " "		5 " "
	6 " "		6 " "
	7 Data bus (MSB) <sup>(2)</sup>		7 Data bus (MSB) <sup>(2)</sup>
<b>X</b>	8 -----	8	Write (WR)
	9* Fine corsa (LS1x)	9	Read (RD)
	10* Fine corsa (LS2x)	10	Port select (PS)
	11* Micro In/Ref. Asse X	11*	Chip select (1/0=Asse X/Y)
<b>Y</b>	12 -----	12	Clock (PCA2.D14)
	13* Fine corsa (LS1y)	13	Data ( " )
	14* Fine corsa (LS2y)	14*	Enable asse X (PCA2.D14)
	15* Micro In/Ref. Asse Y	15*	Enable asse Y (PCA2.D14)

Gli indirizzi indicati sono relativi all'indirizzo base.

Indirizzo assoluto = indirizzo base del modulo + indirizzo relativo.

Solo gli indirizzi contrassegnati con (\*) sono di interesse dell'utente.

Tutti gli altri indirizzi sono utilizzati nell'ambito dei Blocchi Funzione.

I segnali "Enable" per il display PCA2.D14 sono attivi a livello basso; nel modulo il livello di queste uscite viene invertito.

<sup>(1)</sup> LSB = Least Significant Bit (Bit meno significativo)

<sup>(2)</sup> MSB = Most Significant Bit (Bit più significativo)



## 6. Funzionamento

### 6.1 Modi di Funzionamento

---

Esistono due modi di funzionamento:

- CONTROLLO DI POSIZIONE
- CONTROLLO DI VELOCITÀ

#### Controllo di posizione

Il posizionamento richiede la seguente sequenza di comandi:

1. Introduzione della posizione e dei parametri per il profilo della velocità
2. Inizio posizionamento
3. Attesa del segnale "posizione di destinazione raggiunta"

Dopo l'introduzione dei parametri (fattori PID, velocità, accelerazione ecc.) l'operazione di posizionamento comporta un avvicinamento controllato alla posizione di destinazione; velocità, fattori PID e posizione di destinazione possono essere modificate durante il movimento.

#### Controllo di velocità

Sequenza di comandi:

1. Introduzione dei parametri per il profilo della velocità
2. Inizio movimento
3. Interruzione movimento introducendo un comando di stop.

Nel funzionamento in controllo di velocità, la velocità cresce in base all'accelerazione definita, fino al raggiungimento della velocità nominale. Il funzionamento viene quindi controllato a tale velocità fino alla ricezione di un comando di stop. La velocità nominale può essere modificata durante il movimento.

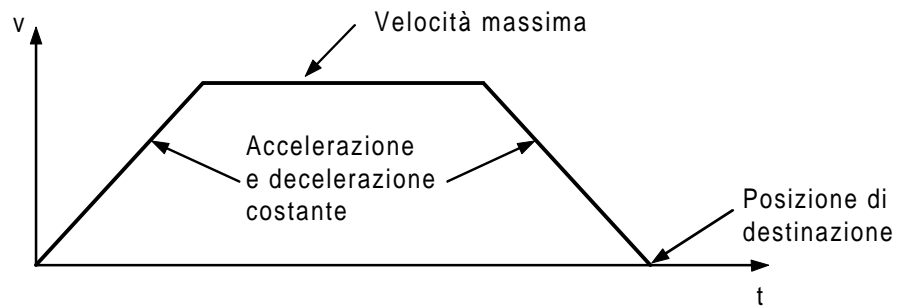
#### Unità funzionali

Come si può osservare nello schema logico, il modulo è costituito dalle seguenti unità funzionali:

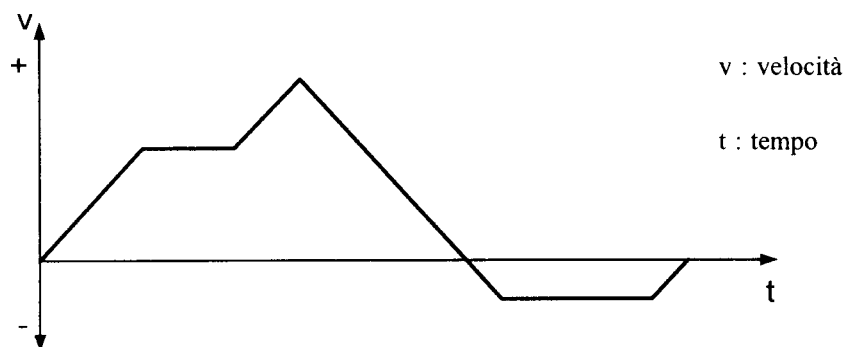
- Generatore del profilo della velocità
- Regolatore PID
- Decodifica posizione e circuito di ingresso
- Interfaccia bus (ASIC) verso il bus PCD
- Convertitore D/A per l'uscita analogica o generatore di impulsi (PWM)

## 6.2 Generatore del profilo di velocità

In base alla accelerazione e alla velocità indicate, il generatore di profilo calcola la velocità impostata come una funzione del tempo sia in modo posizionamento che in modo velocità. Quando lavora in modo posizionamento, la differenza tra la posizione impostata e la posizione reale è inviata al regolatore PID. In questo modo si ottiene un posizionamento preciso del motore.



Profilo di velocità standard



Profilo di velocità in caso di modifica dei parametri (di velocità e posizione) durante il movimento.

La velocità e la posizione di destinazione possono essere modificate in qualsiasi punto durante il funzionamento e il regolatore provvederà ad accelerare o decelerare in base all'accelerazione definita.

Nel funzionamento in controllo di velocità, il regolatore accelera fino a raggiungere la velocità definita dall'utente, quindi continua a velocità costante finché non riceve un comando di arresto.

Principio di funzionamento del controllo di velocità:

La posizione impostata viene incrementata in continuazione (in base alla velocità desiderata). La differenza tra la posizione impostata e quella reale (fornita dall'encoder) viene ripetutamente passata al regolatore PID. Quest'ultimo provvede a compensare la fluttuazione di velocità causata da eventuali interferenze, aumentando o diminuendo automaticamente l'uscita del regolatore.

Se il motore non raggiunge la velocità impostata (ad esempio a causa di un bloccaggio del rotore), la differenza tra la posizione impostata e la posizione reale è elevata.

Ciò produce un messaggio di errore di posizione che può generare un allarme o arrestare automaticamente il motore. L'errore di posizione massimo accettabile è regolabile.

### 6.3 Regolatore PID

Il regolatore PID abilita il motore ad avvicinarsi con precisione alla posizione di destinazione e a mantenere tale posizione, quando è attivo, finché non viene ricevuto un comando di arresto.

Il regolatore utilizza il seguente algoritmo:

$$U(n) = k_p * e(n) + k_i * \sum_{N=0}^n e(n) + k_d * [e(n') - e(n'-1)]$$

Dove :  $U(n)$  --> Uscita di controllo per il motore

$e(n)$  --> Errore di posizione alla n<sup>sima</sup> scansione

$n$  --> Scansione per la parte integrale

$n'$  --> Scansione per la parte derivativa

$k_p$  --> Fattore proporzionale

$k_i$  --> Fattore integrale

$k_d$  --> Fattore derivativo

Parametri definibili dall'utente:

- Fattori di controllo  $k_p$ ,  $k_i$ ,  $k_d$
- Tempo di scansione termine derivativo
- Limite di integrazione (IL) per la parte integrale

È possibile modificare i **fattori  $k_p$ ,  $k_i$  e  $k_d$**  durante un movimento.

Il **tempo di scansione** per le **parti proporzionale e integrale** è di 341µs. Ciò significa che l'uscita di controllo viene aggiornata ad intervalli di 341µs.

Il **tempo di scansione per la parte derivativa** può essere definita in multipli di 341µs (massimo 256\*341µs). Per operazioni a bassa velocità dovrebbero essere selezionati tempi lunghi.

Il **limite di integrazione IL**, limita la quantità che risulta dall'espressione:

$$k_i * \sum_{N=0}^n e(n)$$

## 6.4 Decodifica di posizione e circuito di ingresso

### Registrazione della posizione e della velocità

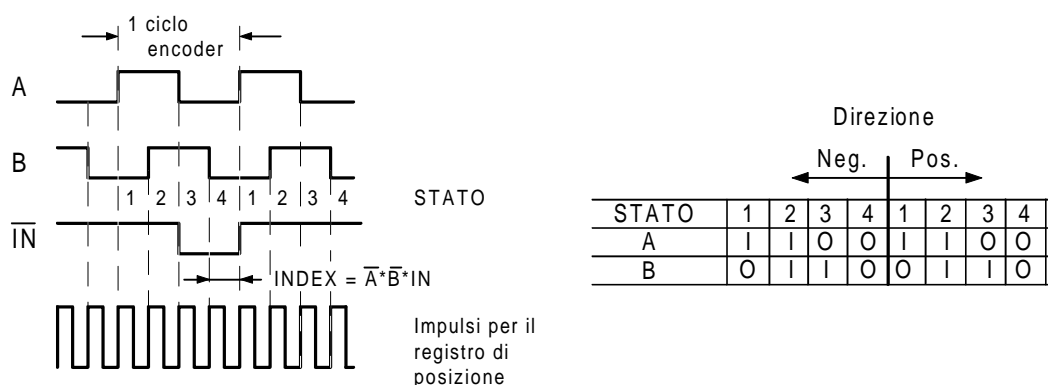
La posizione e la velocità del motore sono lette da un encoder incrementale. Si possono utilizzare i seguenti segnali dell'encoder:

PCD4 .H310 }  
 " .H320 } A,B,IN e  $\overline{A},\overline{B},\overline{IN}$  (morsetti)  
 " .H316 } segnali a 24V in logica positiva o negativa.

PCD4 .H311 }  
 " .H321 } A,A; B,B; IN,IN; (connettore frontale tipo D)  
 " .H317 } ingressi 5V RS 422 (line driver)  
 " .H327 }

### Ingressi A, B, $\overline{IN}$ :

Forme d'onda dei segnali A,B, $\overline{IN}$  sul decodificatore di posizione :



### Ingressi A, B :

Ad ogni transizione (0 -> 1 e 1 -> 0) dei segnali A e B, il registro di posizione interno viene incrementato o decrementato di 1. In questo modo si moltiplica per quattro la risoluzione dell'encoder. La posizione di riferimento deve essere quindi anch'essa moltiplicata per quattro. Per il decodificatore di posizione, i segnali devono avere esattamente la sequenza illustrata nella figura precedente. Se l'encoder fornisce segnali diversi, si deve utilizzare un ponticello per invertirli (vedere le pagine seguenti).

Ingresso  $\overline{\text{IN}}$ :

Nel caso di moduli per encoder a 24V (tipi H310, 320, 316 e 326), l'ingresso  $\overline{\text{IN}}$  può essere utilizzato come ingresso per l'impulso indice (segnale di zero proveniente dall'encoder) o come punto di riferimento.

- Utilizzo come ingresso per impulso indice:
- (ponticello IN/RP in posizione IN)
  
- Ogni volta che i tre segnali dell'encoder sono tutti a zero e dopo che è stato chiamato il blocco funzione “SetIP” (set index position = imposta la posizione dell'indice), la posizione assoluta del motore viene scritta nel registro della posizione indice.
  
- Utilizzo come ingresso per punto di riferimento:  
(ponticello IN/RP in posizione RP)

Si può collegare un microinterruttore di riferimento, ad esempio per definire la posizione zero. Tuttavia per fare ciò è necessario che il ponticello IN/RP sia in posizione RP, disattivando così l'ingresso per il decodificatore di posizione.

Quando si utilizzano i moduli H310, 320, 316 o 326 con l'ingresso  $\overline{\text{IN}}$  come segnale indice, il micro di riferimento deve essere collegato ad un ingresso di un modulo di ingresso digitale (es. E 100).

### **Moduli per il collegamento dei segnali su encoder a 5V**

Questi moduli vengono montati su una piastra addizionale che genera direttamente l'alimentazione a 5V e che si collega all'encoder attraverso un connettore tipo D. Quando si utilizzano i moduli H311, 321, 317 o 327, si può collegare un micro di riferimento (sui terminali) oltre al segnale indice (sul connettore tipo D).

Come si può desumere dallo schema seguente, per collegare un encoder a 5V si deve utilizzare un cavo schermato:

- lunghezza massima del cavo : 20m
- sezione minima del conduttore : 0.25mm<sup>2</sup>

Per assicurare un elevato grado di immunità ai disturbi, si deve utilizzare un connettore tipo D con rivestimento metallico, collegato direttamente alla massa di protezione (PGND) del PCD4.

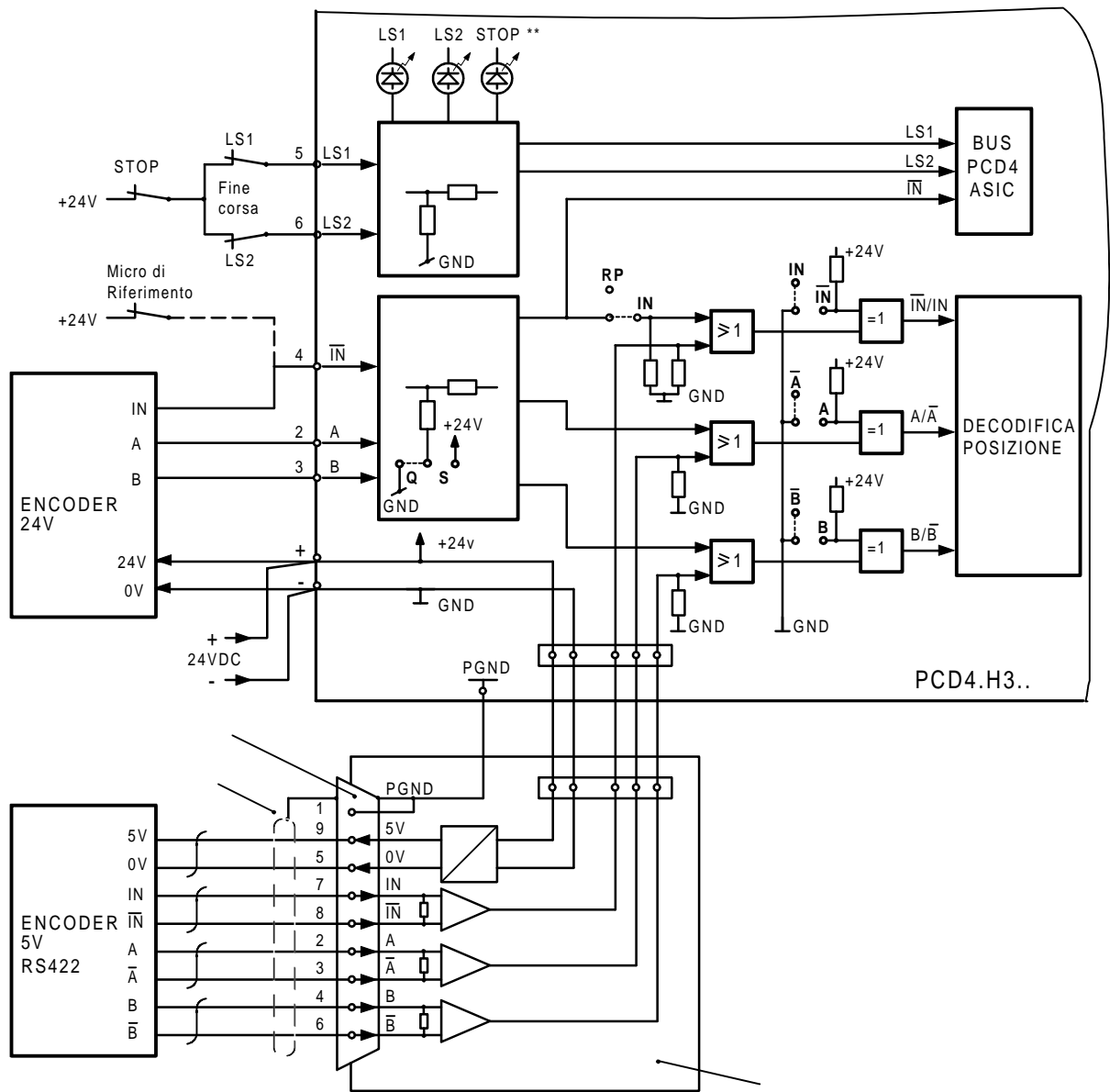
**Micro di fine corsa e di riferimento**

I microinterruttori di fine corsa (ingressi LS1 e LS2) e di riferimento (ingresso IN) richiedono una alimentazione a 24V CC. I segnali provenienti da questi microinterruttori possono essere letti attraverso il bus del PCD4. Ciò significa che devono essere tenuti sotto controllo dal programma utente, in modo da poter eseguire qualsiasi procedura necessaria.

I fine corsa (LS1/2) e il micro di arresto (opzionale) collegato a questi, non dovrebbero mai essere utilizzati come precauzioni di sicurezza. A questo scopo dovrebbero essere destinati dei micro di sicurezza e di emergenza aggiuntivi che intervengano direttamente sui circuiti di pilotaggio del motore principale.

**Schema degli ingressi e dei connettori :**

(è rappresentato un solo asse)



\*) Per quanto riguarda le misure di sicurezza, fare riferimento al paragrafo precedente: "Micro di fine corsa e di riferimento"

\*\*\*) Tabella di stato dei LED (LS1, LS2 e STOP) :

INGRESSI		LED		
LS1	LS2	LS1	LS2	STOP
24V	24V	SPENTO	SPENTO	SPENTO
APERTO	24V	ACCESO	SPENTO	SPENTO
24V	APERTO	SPENTO	ACCESO	SPENTO
APERTO	APERTO	SPENTO	SPENTO	ACCESO



Selezione dei ponticelli (**fare riferimento anche al circuito stampato, cap. 3.1**) :

Ponticello	Funzione	Predisposizione di fabbrica
Q/S	Ingressi A, B, $\overline{IN}$ per funzionamento in logica positiva o negativa	Funzionamento in logica positiva (Q)
IN/RP	Ingresso per Impulso indice o punto di riferimento	Impulso indice (IN)
A/ $\overline{A}$	Inversione di fase segnale A	$\overline{A}$
B/B	Inversione di fase segnale B	$\overline{B}$
IN/ $\overline{IN}$	Inversione di impulso indice $\overline{IN}$	IN

} solo encoder a 24V (per Q/S e IN/RP)  
} sia encoder a 24V (per A/ $\overline{A}$ , B/B e IN/ $\overline{IN}$ )

- 1) La commutazione tra funzionamento in logica positiva o negativa, per i segnali degli encoder a 24V, viene realizzata con un solo ponticello per entrambi gli assi.
- 2) Per encoder a 24V è possibile collegare attraverso i terminali, 4 (per l'asse X) o 12 (per l'asse Y):
  - segnali indice per l'encoder
  - oppure
  - microinterruttori di riferimento

Per encoder a 5V i segnali indice vengono forniti attraverso il connettore tipo D, quindi i terminali 4 e 12 vengono lasciati liberi per il micro di riferimento (ponticello in posizione RP). Il ponticello "IN/RP" può essere selezionato separatamente per ciascun asse sulla piastra a circuito stampato.

- 3) L'inversione del segnale dell'encoder (sia 24 che 5V) richiede inoltre 1 ponticello per ciascuno dei 3 segnali: A, B e IN; anche questi ponticelli sono condivisi da entrambi gli assi.

### Apertura del modulo contenitore per modificare i ponticelli

Per poter modificare la posizione dei ponticelli, è necessario rimuovere la piastra a circuito stampato dal modulo contenitore. Per fare ciò si deve premere sulle chiusure a molla poste sui due lati del pannello frontale. A questo punto svitare la vite posta in alto a sinistra sul modulo che contiene la piastra a circuito stampato, in modo che la stessa possa essere rimossa dal contenitore.

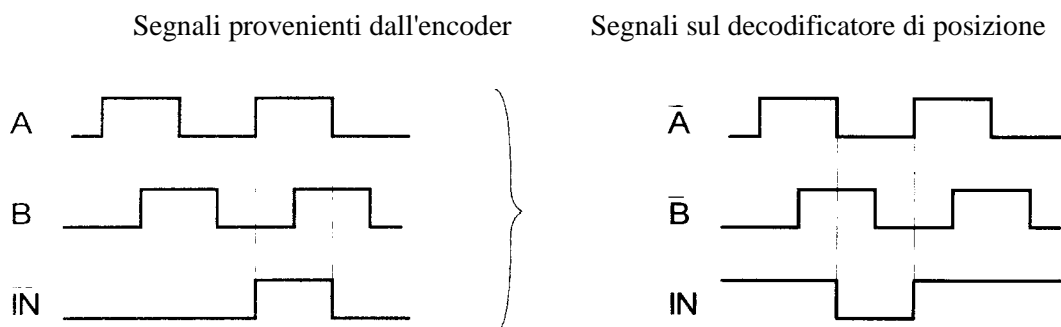
Quando il ponticello è stato inserito nel modo corretto, fissare nuovamente la piastra e richiudere il contenitore.



**Nota:** La piastra a circuito stampato principale, così come i moduli analogici, contiene dei componenti sensibili alle cariche elettrostatiche.

### Esempio di riferimento per la selezione dei ponticelli

Un encoder fornisce i seguenti segnali a 24V in logica positiva:



Per produrre la sequenza di segnali desiderata in ingresso al decodificatore di posizione, i tre segnali devono essere tutti invertiti.

<p>I ponticelli: <math>\overline{\text{A}}/\overline{\text{A}}</math> --&gt; <math>\overline{\text{A}}</math></p> <p><math>\overline{\text{B}}/\overline{\text{B}}</math> --&gt; <math>\overline{\text{B}}</math></p> <p><math>\overline{\text{IN}}/\overline{\text{IN}}</math> --&gt; <math>\overline{\text{IN}}</math></p>	}	<p>corrispondono alle predisposizioni di fabbrica</p>
--	---	---

### Importante :

Se si stanno utilizzando **encoder a 24V con microinterruttori di riferimento** (ponticello "IN/RP" in posizione RP) il ponticello IN,IN deve essere posto in posizione IN.

Motivo: Il segnale indice  $\overline{\text{IN}}$  in ingresso al decodificatore di posizione, non deve essere lasciato permanentemente a zero in quanto, ad elevate velocità ciò può portare ad un errore nel regolatore.

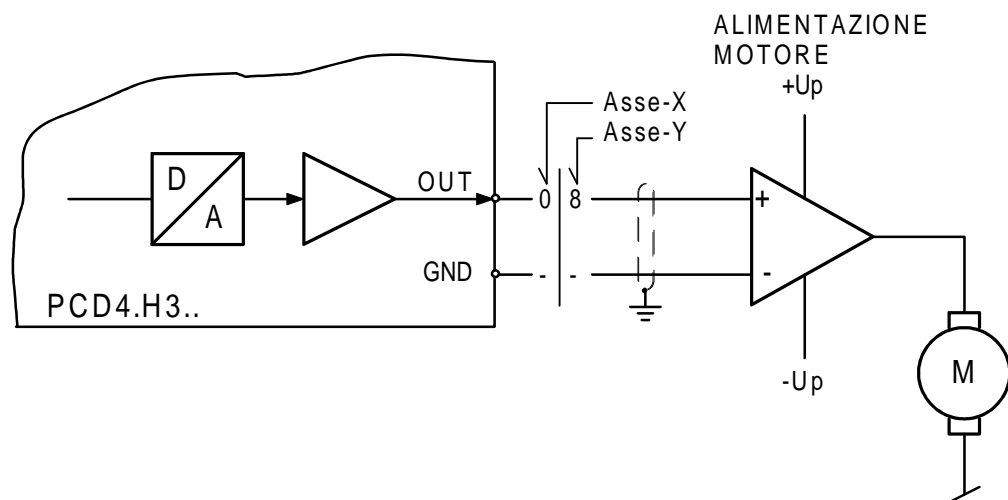
## 6.5 Convertitore D/A (uscita di controllo analogica)

I moduli: PCD4.H310  
 " .H320  
 " .H311  
 " .H321 } hanno una uscita analogica per il controllo del motore

Per ogni asse è previsto un convertitore D/A a 12-bit.

### Collegamento all'uscita analogica:

(è rappresentato un solo asse)



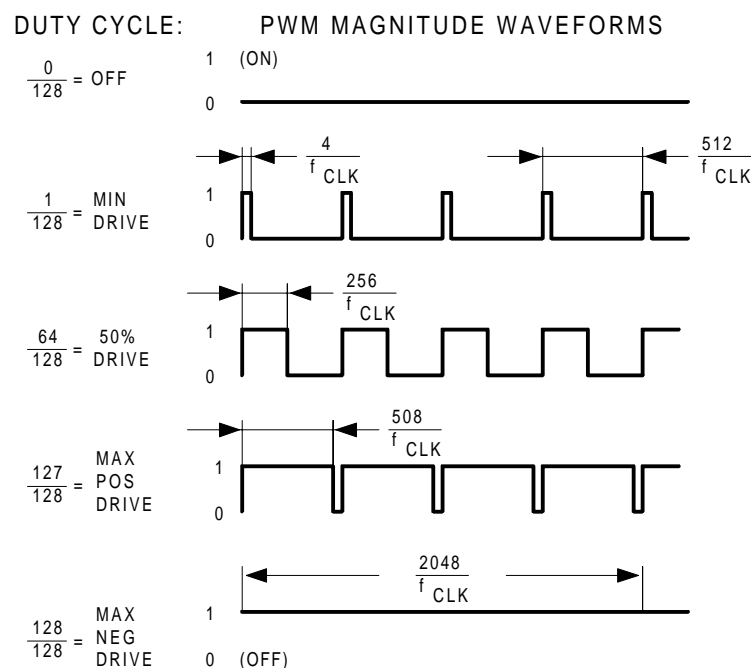
## 6.6 Generatore PWM

I moduli: PCD4 .H316 } hanno una uscita PWM con segnale  
 " .H326 } SIGN e MAGNITUDE oppure con la  
 " .H317 } logica di uscita per pilotare direttamente  
 " .H327 } un chopper.

In alcuni amplificatori di potenza la logica citata in precedenza è integrata. Quindi è possibile emettere i segnali SIGN e MAGNITUDE sia direttamente che attraverso un circuito logico utilizzando un ponticello.

### Segnale PWM sull'uscita del generatore PWM: (senza segnale SIGN)

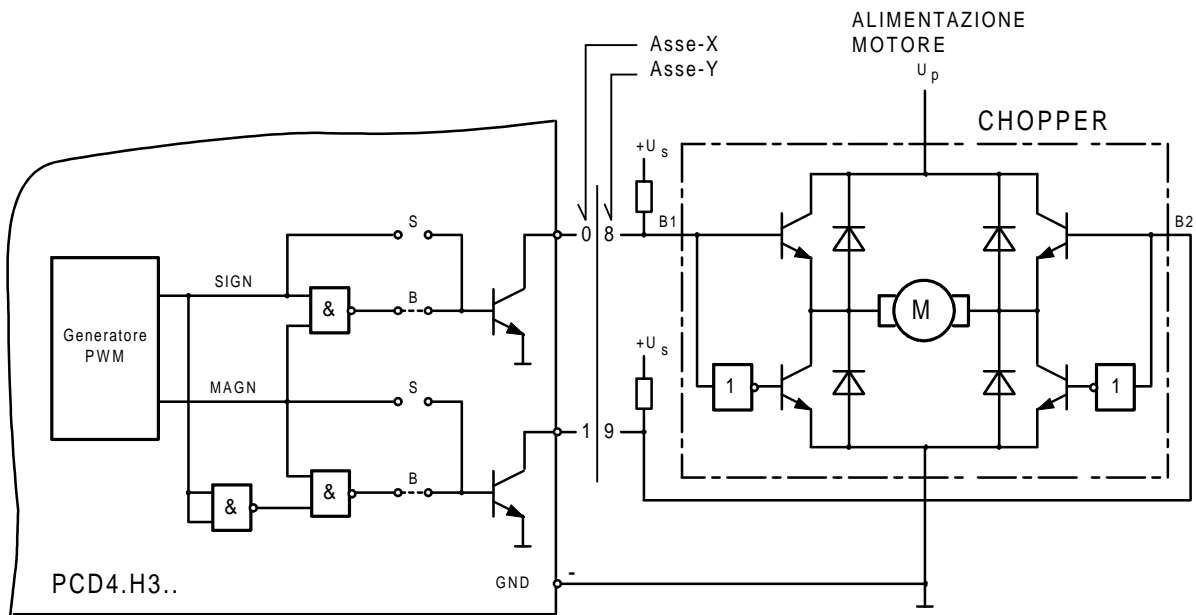
Il segnale ha una risoluzione a 8 bit.  
 Range: -128 .... +127



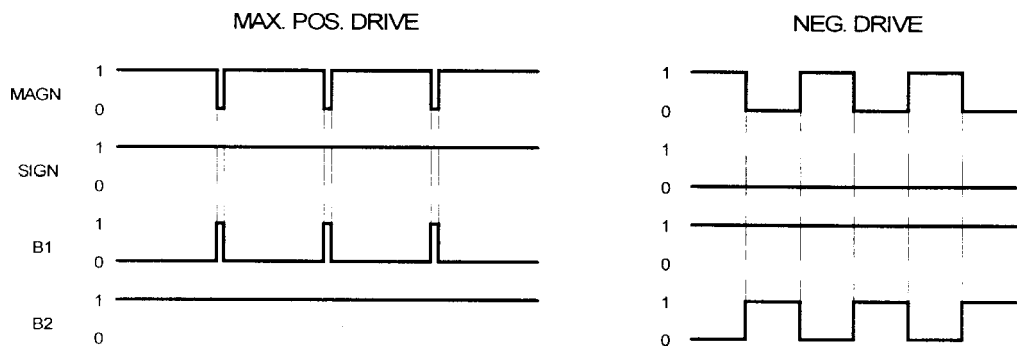
$$f_{\text{CLK}} = 6\text{MHz}$$

**Schema logico dell'uscita PWM:**

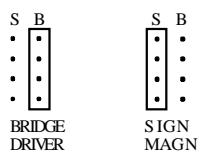
(è rappresentato un solo asse)



La figura illustra la connessione tra i segnali SIGN/MAGN e l'uscita per il pilotaggio diretto del ponte.



Selezione ponticello:



Selezione separata per le uscite degli assi X e Y.

Predisposizione di fabbrica: SIGN, MAGN (Pos. S)

## 7. Scrittura dei programmi per il modulo H3

### 7.1 Installazione software

---

#### 7.1.1 Il pacchetto software PCD9.H3E1

(PCD9.H3E1 dischi da 5.25", PCD9.H3E6 dischi da 3.5")

Il pacchetto software PCD9.H3 contiene i blocchi funzione scritti in linguaggio PCD che possono essere richiamati da un programma utente per il controllo del modulo H3.

Il pacchetto contiene i seguenti due file:

- |                  |   |
|------------------|---|
| <b>H3DEF.SRC</b> | Questo file contiene la dichiarazione di tutti i simboli utilizzati dal programma. Contiene inoltre la configurazione di installazione del modulo H3. |
| <b>H3FB.SRC</b>  | Questo file contiene i blocchi funzione per il controllo del modulo.  |

I requisiti del pacchetto sono:

- numero di linee di programma  $\leq$  1250
- livelli di nidificazione degli FB 6

#### 7.1.2 Assemblaggio e link del file

Esistono due modi per assemblare ed eseguire il link di file H3. Si possono utilizzare definizioni di simboli esterni (globali), oppure utilizzare simboli definiti localmente nel file H3DEF.SRC che, in questo caso deve essere incluso nel programma utente con la direttiva assembler \$INCLUDE. Il programma H3 può supportare entrambi i metodi.

Il metodo di default è quello di non utilizzare definizioni di simboli esterni.

### Utilizzo di definizioni di simboli locali

Il file di definizione simboli H3DEF.SRC deve essere incluso all'inizio del programma utente con la direttiva \$INCLUDE e i file devono essere editati in modo tale che vengano definite le seguenti EQUates

- H3DEF.SRC : PUBLSYM EQU 0
- H3FB.SRC : EXTNSYM EQU 0

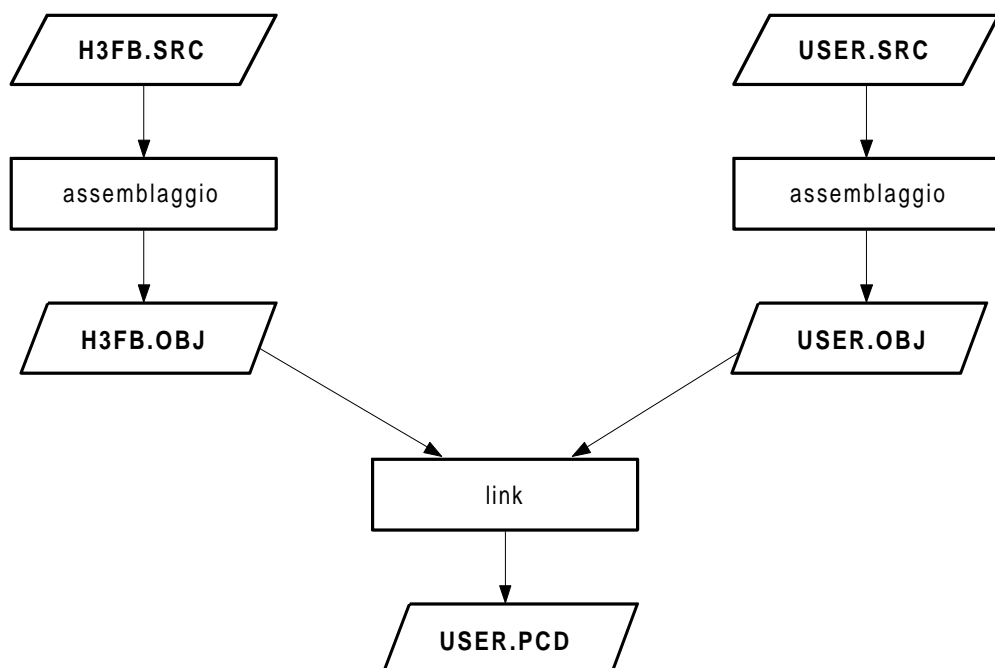
Inclusione del file H3 nel programma utente:

File del programma utente (es. USER.SRC)

```
<inizio file> (titolo, intestazione progr. ecc.)  
  
$INCLUDE H3DEF.SRC  
$INCLUDE <altri file da includere>  
.  
"  
"  
<progr. utente>  
"  
"  
<fine file>
```

Il file di definizione dei simboli H3 deve essere incluso prima di qualsiasi altro file include (che può far uso di simboli definiti in H3DEF.SRC).

Il flusso seguente illustra le fasi di assemblaggio e di link dei file.



## Utilizzo di simboli esterni

Tutti i file vengono assemblati individualmente e quindi linkati. I file devono essere prima editati in modo tale che vengano definite le seguenti EQUates:

```
- H3DEF.SRC : PUBLSYM    EQU    1
- H3FB.SRC  : EXTNSYM    EQU    1
```

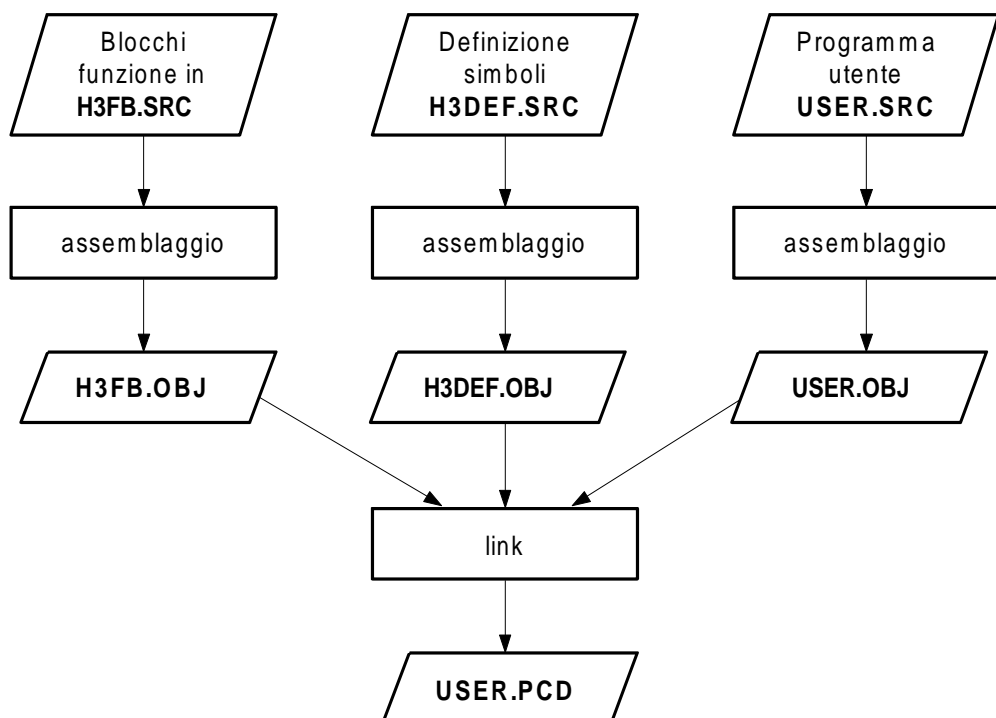
Se si sceglie questo metodo, tutti i simboli usati (dal file H3DEF.SRC) devono essere definiti come EXTERnal (Esterni) nel programma utente. Gli indirizzi dei registri e dei flag ("RA1", "FA1" ecc.) devono essere definiti nel file, dal momento che l'assembler non può sommare due simboli esterni (es. SET F FSart+FA1).

La definizione di "RAi" e "FAi" nel programma utente (questi sono descritti nel capitolo 7.1.3) è la seguente:

```
RA1    EQU    0*NoRfeA
RA2    EQU    1*NoRfeA
|
FA1    EQU    0*NoFfeA
FA2    EQU    1*NoFfeA
|
```

Per la stessa ragione, devono essere definiti i simboli "NoRfeA" e "NoFfeA". Tali definizioni si possono prendere dal file H3DEF.SRC.

Il flusso seguente illustra le fasi di assemblaggio e di link.





### 7.1.3 Configurazione dell'installazione H3 nel file H3DEF.SRC

L'installazione di H3 deve essere configurata in H3DEF.SRC prima di iniziare la programmazione.

I dettagli della configurazione comprendono: l'indirizzo base del primo modulo H3, il numero di assi, gli indirizzi base degli elementi utilizzati ecc. Tutte le definizioni sono impostate con i valori di default e possono all'occorrenza essere modificate. Di seguito vengono riportate le informazioni riguardanti la configurazione che si trovano all'inizio del file H3DEF.SRC.

Simbolo	Valore di default	Commento
FMAH3	EQU 0	; Indirizzo del primo modulo H3 ; Definisce l'indirizzo base del primo modulo H3. ; NOTA: Tutti i moduli H3 devono essere inseriti uno dopo l'altro ; senza lasciare spazi vuoti nel bus PCD4
IMode	EQU 6	; Modalità di inizializzazione ; Definita in base all'uscita ; utilizzata per il "set point" (analogico/PMW). ; Moduli H310,311,320,321: IMode=6 ; Moduli H316,317,326,327: IMode=5
MNA	EQU 2	; Massimo numero di assi ; Definisce il numero di assi utilizzato. ; Il numero di registri e di Flag da riservare dipendono da queste informazioni.
BAF	EQU 2000	; Indirizzo base dei Flag
BAR	EQU 2000	; Indirizzo base dei Registri
BAC	EQU 1000	; Indirizzo base dei Contatori
BAFB	EQU 900	; Indirizzo base dei Blocchi Funzione

Simbolo	Valore di default	Commento
RA1	EQU 0*NoRfeA	; Indirizzo del blocco di Registri per ; l'Asse 1. ; Questa costante indica il primo ; registro nell'ambito del blocco dei ; registri occupati dall'asse n° 1. ; La costante "NoRfeA" definisce il ; numero di registri occupati ; per ogni asse. ; "NoRfeA" non deve essere modificato
RA2	EQU 1*NoRfeA	; Indirizzo del blocco di Registri ; per l'Asse 2.
FA1	EQU 0*NoFfeA	; Indirizzo del blocco di Flag per ; l'Asse 1. ; Questa costante indica il primo ; flag nell'ambito del blocco di flag ; occupati dall'asse n° 1. ; La costante "NoFfeA" definisce ; il numero di flag occupati ; per ogni asse. ; "NoFfeA" non deve essere modificato
FA2	EQU 0*NoFfeA	; Indirizzo del blocco di Flag per ; l'Asse 2.

Se vengono utilizzati più di due assi, devono essere definiti ulteriori blocchi di registri e di flag (RA1, RA2, FA1, FA2 ...).

Ad esempio:

Asse i	RAi	FAi
1	0*NoRfeA	0*NoFfeA
2	1*NoRfeA	1*NoFfeA
3	2*NoRfeA	2*NoFfeA
4	3*NoRfeA	3*NoFfeA
5	4*NoRfeA	4*NoFfeA

ecc.

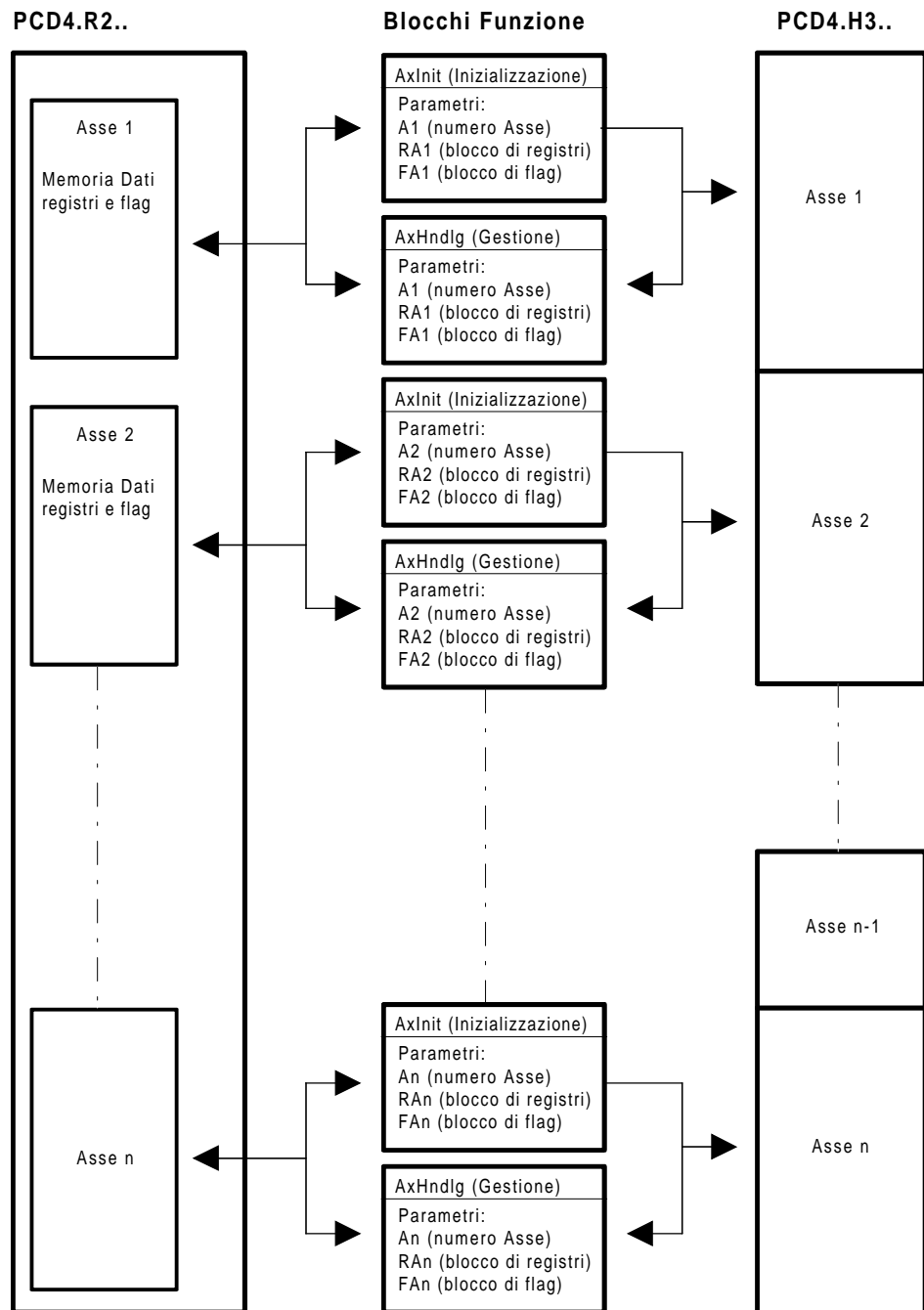
I dati della configurazione sono seguiti da ulteriori dichiarazioni di simboli che NON devono essere modificate.

Dopo aver eseguito la configurazione può essere utile sapere quanti elementi vengono utilizzati dal software H3. Al termine del file H3DEF.SRC si trova una tabella di simboli che contiene i valori assegnati dall'assembler. Dal listing del file H3DEF.LST, si può determinare quanti elementi sono utilizzati.

Elementi utilizzati dal software H3:

Simbolo	Valore	Commento
TFI	EQU F NBF-BAF	; Totale dei Flag utilizzati
TCo	EQU C NBC-BAC	; Totale dei Contatori utilizzati
TRe	EQU R NBR-BAR	; Totale dei Registri
TFB	EQU NFB-BAFB	; Totale dei Blocchi funzione ; utilizzati
NFF	EQU FNBF	; Primo Flag libero
NFC	EQU C NBC	; Primo Contatore libero
NFR	EQU R NBR	; Primo Registro libero
NFFB	EQU NFB	; Primo Blocco funzione libero

## 7.1.4 Trasferimento dati CPU &lt;--&gt; Modulo H3



Il modulo H3 è controllato da due blocchi funzione "AxInit" e "AxHndlg" e i dati relativi agli assi vengono memorizzati in aree di memoria specifiche per ciascun asse. I blocchi di registri e di flag definiti per ciascun asse, memorizzano i parametri operativi che vengono letti e scritti durante lo scambio dati con il modulo H3.

L'asse e i blocchi dati vengono associati correttamente specificando il numero di Asse "Ai", l'indirizzo del blocco di registri "RAi" e l'indirizzo del blocco di flag "FAi" all'atto della chiamata del blocco funzione.

### 7.1.5 Organizzazione e Accesso alla memoria dati

Indirizzo del blocco registri	Allocazione registri		
	Indirizzo	Simbolo	Funzione
Indirizzo del registro base		Asse 1	
	BAR+RA1+0	KProp	Fattore proporzionale
	BAR+RA1+1	KInt	Fattore integrale
	BAR+RA1+2	KDer	Fattore derivativo
	BAR+RA1+3	IntL	Limite di integrazione
	BAR+RA1+4	SampI	Intervallo di campionamento
	BAR+RA1+5	MCFac	Fattore di controllo movimento
	BAR+RA1+6	PosEr	Errore di posizione
	BAR+RA1+7	DestP	Posizione di destinazione
	BAR+RA1+8	BrkP	Posizione break point
	BAR+RA1+9	Veloc	Velocità
	BAR+RA1+10	Accel	Accelerazione
	BAR+RA1+11	StaFRR	Registro reset flag di stato
	BAR+RA1+12	MCW	Parola di controllo movimento
	BAR+RA1+13	RActP	Posizione effettiva
	BAR+RA1+14	RSetP	Posizione impostata
	BAR+RA1+15	RActV	Velocità effettiva
	BAR+RA1+16	RSetV	Velocità impostata
	BAR+RA1+17	RIndP	Posizione indice
	BAR+RA1+18	RIntTS	Somma integrale
	BAR+RA1+19	RSigB	Registro segnali
	BAR+RA2+0	Asse 2	
	BAR+RA2+19		
	BAR+RA3+0	Asse 3	
	BAR+RA3+19		
	BAR+RA <sub>n</sub> +0	Asse n	
	BAR+RA <sub>n</sub> +19		
		Registri condivisi	Utilizzati dai blocchi funzione come area di lavoro in memoria

**Allocazione dei Flag**

Indirizzo del blocco di flag	Indirizzo	Simbolo	Funzione	
Indirizzo di base	Asse 1			
	BAF+FA1+0	OnDest	Posizione di destinaz. raggiunta	
	BAF+FA1+1	IPuls	Impulso indice	
	BAF+FA1+2	WrapOc	Overflow registro di posizione	
	BAF+FA1+3	ExcPEr	Errore di posizione	
	BAF+FA1+4	BrkPos	Posizione break point raggiunta	
	BAF+FA1+5	DplM	Modo display PCA2.D14	
	-----			Flag di comando:
	BAF+FA1+6	FLdDR	Carica destinazione relativa	
	BAF+FA1+7	FLdDA	Carica destinazione assoluta	
	BAF+FA1+8	FLdVR	Carica velocità relativa	
	BAF+FA1+9	FLdVA	Carica velocità assoluta	
	BAF+FA1+34	FBackw	Ritorno alla velocità definita	
	BAF+FA2+0	Asse 2		
BAF+FA2+34				
BAF+FA3+0	Asse 3			
BAF+FA3+34				
BAF+FA <sub>n</sub> +0	Asse n			
BAF+FA <sub>n</sub> +34				
	Flag condivisi		Utilizzati dai blocchi funzione come area di lavoro in memoria.	

### Allocazione del contatore

É utilizzato un solo contatore condiviso da tutti gli assi.

### Accesso del programma utente ai registri e ai flag

I parametri degli assi sono individuati attraverso i nomi dei simboli (definiti in H3DEF.SRC). Per ciascun asse i parametri condividono lo stesso nome simbolo. Per riferirsi ad un parametro effettivo, al nome del simbolo viene aggiunto l'indirizzo di inizio del blocco di parametri relativo all'asse richiesto. Se si utilizza solo il nome del simbolo si fa riferimento ai parametri dell'asse 1. Gli indirizzi di inizio dei blocchi di parametri sono definiti dalle costanti "RAi" e "FAi" (blocchi di registri e di flag per l'asse i), come risulta dall'esempio illustrato più avanti. Per facilitarne l'interpretazione, i nomi dei simboli sono stati scritti a livello software utilizzando un misto di lettere maiuscole e minuscole. Tuttavia, dal momento che l'assembler PCD ignora la notazione maiuscola o minuscola, i nomi dei simboli all'interno del programma utente possono essere scritti indifferentemente in maiuscolo o minuscolo. Ad esempio "RA1" e "ra1" sono equivalenti.

Prima di poter caricare un parametro nel modulo H3, occorre prima caricare nel registro corrispondente il valore desiderato.

Esempio: Per caricare il registro "DestP" (posizione di destinazione per un determinato asse):

```
- Asse 1 —> LD R   DestP+RA1
                Valore

- Asse 3 —> LD R   DestP+RA3
                Valore

- Asse i —> LD R   DestP+RAi
                Valore
```

Per caricare la posizione di destinazione nel modulo H3, occorre eseguire il comando "Load Destination Position" (Carica Posizione di Destinazione). Impostando il flag "FLdDA" (Carica Destinazione Assoluta) il registro "DestP" verrà letto dal blocco funzione "AxHndlg" e caricato nel modulo H3.

Esempio: Per caricare i dati di destinazione "destinazione assoluta" nel modulo H3 per un dato asse:

- Asse 1 —> SET F FLdDA+FA1
- Asse 3 —> SET F FLdDA+FA3
- Asse n —> SET F FLdDA+FA<sub>n</sub>

Una volta iniziato il movimento, è necessario aspettare finchè non viene raggiunta la posizione di destinazione. Per fare ciò occorre interrogare il flag di stato "OnDest".

Esempio: Per verificare il flag di stato "OnDest" per un dato asse:

- Asse 1 —> STH F OnDest+FA1
- Asse 3 —> STH F OnDest+FA3
- Asse n —> STH F OnDest+FA<sub>n</sub>



## 7.2 Blocchi funzione principali: "AxInit" e "AxHndlg"

La comunicazione con i moduli H3 avviene esclusivamente chiamando i due blocchi funzione (FB) "AxInit" (Axis Initialisation) e "AxHndlg" (Axis Handling).

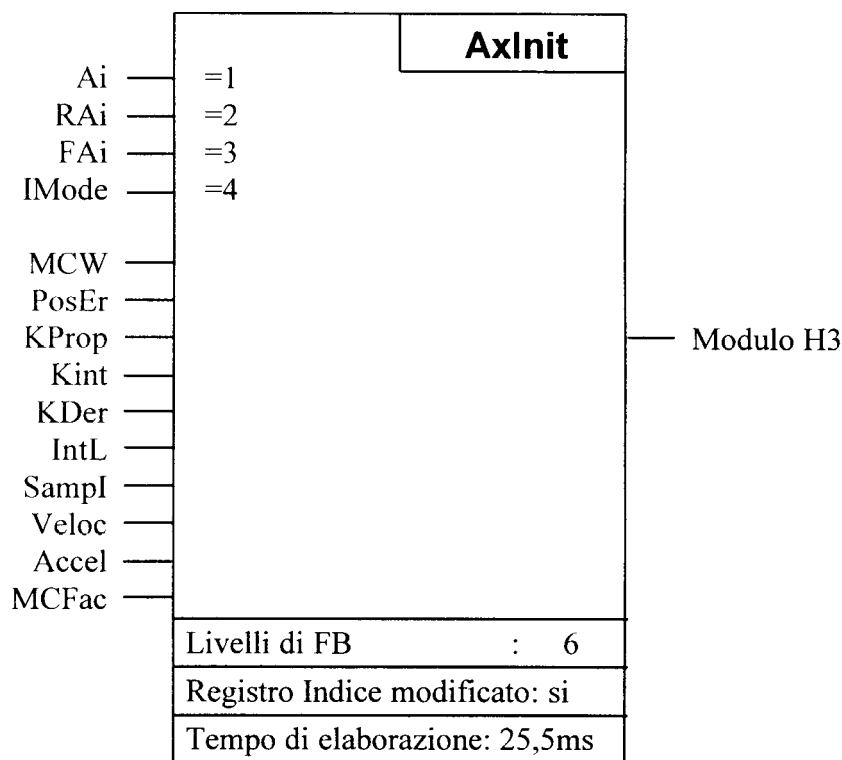
Richiamando questi FB, vengono forniti il numero di asse, l'indirizzo del blocco di registri e l'indirizzo del blocco di flag in modo tale da indirizzare i dati corrispondenti. I FB devono essere chiamati separatamente per ciascun asse.

### AxInit

Blocco Funzione: - Inizializzazione Asse

### AxInit

Package Software: PCD9.H3E1



#### Descrizione funzionale:

Questo FB è utilizzato per inizializzare un asse del modulo H3. Deve essere chiamato prima di poter utilizzare un asse. È consigliabile chiamare questo blocco funzione all'avviamento tramite XOB 16; in questo modo viene chiamato una volta sola. Questo FB esegue diversi comandi alcuni dei quali possono anche essere eseguiti dal blocco funzione "AxHndlg" e non vengono qui descritti in dettaglio.

"AxInit" esegue le seguenti funzioni:

### 1. Reset del regolatore nel modulo H3

Questa funzione azzerata tutti i parametri di movimento (accelerazione, velocità, posizione di destinazione e posizione di break point) e tutti i parametri PID (inclusa l'uscita controllata). La posizione corrente coincide con la posizione zero.

### 2. Inizializza la porta di uscita del regolatore

La porta di uscita viene inizializzata in base ai parametri "IMode". Quest'ultima è una costante che deve essere definita nel file H3DEF.SRC in base al modulo H3 utilizzato.

**3. Seleziona la modalità operativa** —> vedere comando "FSelOM"

**4. Carica l'errore di posizione** —> vedere comando "FSetPer"

**5. Carica i parametri di regolazione** —> vedere comando "FLdRP"

**6. Aggiorna i parametri di regolazione** —> vedere comando "FUdRP"

**7. Carica il valore di accelerazione** —> vedere comando "FLdAA"

**8. Carica il valore di velocità** —> vedere comando "FLdVA"

**9. Ripristina tutti i flag di comando** —> vedere FB "AxHndlg"

Esempio: Per chiamare il blocco funzione per l'asse 3

CFB	AxInit	
	3	; Asse numero 3
	RA3	; Blocco di registri per l'asse 3
	FA3	; Blocco di flag per l'asse 3
	IMode	; Modalità di inizializzazione

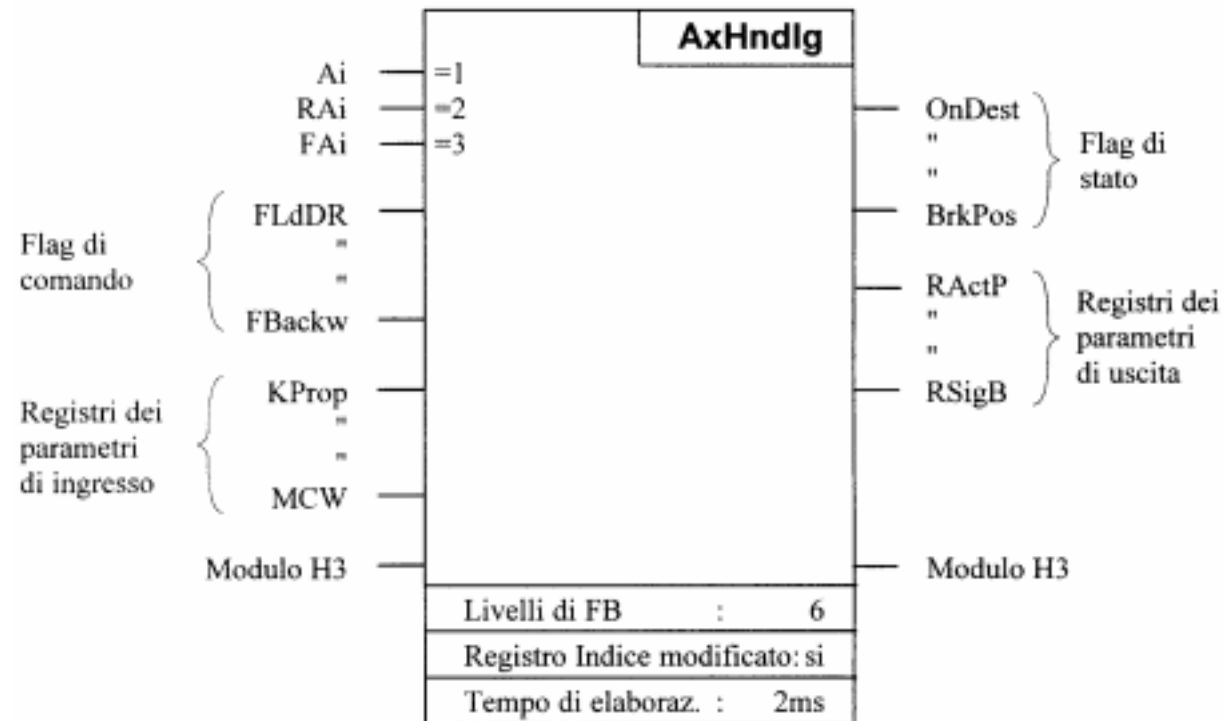
**Descrizione degli ingressi e delle uscite**

A meno che non venga specificato diversamente, tutti i nomi dei simboli utilizzati corrispondono a quelli definiti nel file H3DEF.SRC e non devono essere modificati.

Simbolo	Indicazione/Funzione	Para- metri	Dati		
			Tipo	Formato	Valore
Ai	Asse numero Ai non è definito come simbolo in H3DEF.SRC. Quando si chiama il FB viene usato un valore assoluto.	si	K	Intero	1...32
RAi	Indirizzo del blocco di registri per Asse i	si	K	Intero	(i-1) * 20
FAi	Indirizzo del blocco di flag per Asse i	si	K	Intero	(i-1) * 35
IMode	Modalità di inizializzazione	si	K	Intero	5H/6H
MCW	Parola di controllo movimento	no	R	Binario	-
PosEr	Errore di posizione(numero di impulsi)	no	R	Intero	0...32767
KProp	Fattore proporzionale	no	R	Intero	0...32767
KInt	Fattore integrale	no	R	Intero	0...32767
KDer	Fattore derivativo	no	R	Intero	0...32767
IntL	Limite di integrazione	no	R	Intero	0...32767
SampI	Intervallo di campionamento	no	R	Intero	0...255
Veloc	Velocità	no	R	Intero	vedere comando "FLdVA"
Accel	Accelerazione	no	R	Intero	vedere comando "FLdVA"
MCFac	Fattore di controllo movimento	no	R	virgola mobile	vedere comando "FLdVA"

**AxHndlg**Blocco funzione - **Gestione Asse****AxHndlg**

Package Software: PCD9.H3E1

**Descrizione funzionale:**

Dopo aver eseguito l'inizializzazione dell'asse con "AxInit", la comunicazione con il modulo H3 avviene esclusivamente attraverso questo blocco funzione. I parametri relativi all'asse letti da o inviati al modulo H3, inclusi i comandi di movimento vengono trasmessi al modulo attraverso questo FB.

Gli ingressi e le uscite del blocco funzione si suddividono nei seguenti gruppi:

**Ingressi**

- Parametri FB
- Flag di comando
- Registri contenenti i parametri di ingresso
- Dati del modulo H3

**Uscite**

- Flag di stato
- Registri contenenti i parametri di uscita
- Dati verso il modulo H3

## **Parametri FB**

La selezione dell'asse corretto e dei dati relativi (flag di comando, flag di stato e registri dei parametri) avviene fornendo il numero dell'asse "Ai", l'indirizzo del blocco di registri "RAi" e l'indirizzo del blocco di flag "FAi" all'atto della chiamata del blocco funzione.

## **Flag di comando**

Impostando un flag di comando da programma utente, si può eseguire un comando specifico (ad esempio: caricare la posizione di destinazione, avviare il movimento ecc). L'FB controlla i flag di comando e se uno di questi flag è impostato, viene eseguito il comando richiamandone la relativa sottofunzione (questa ha lo stesso nome del flag di comando, ma senza la lettera "F"). Dopo l'esecuzione del comando, il flag viene azzerato. Se non sono stati impostati flag di comando, l'FB non fa nulla.

Vi sono flag di comando distinti per ogni asse. Il capitolo 7.1.5 illustra come impostare i flag da programma utente.

Per garantire che il modulo H3 non esegua alcun comando all'avviamento, i flag di comando vengono azzerati dal Blocco Funzione di ini-zializzazione "AxInit". Per informazioni dettagliate su ogni singola funzione, fare riferimento al capitolo 7.4.

## **Registri dei parametri di ingresso**

Questi registri contengono tutti i parametri operativi. Il programma utente deve provvedere a caricare i registri opportuni, prima di chiamare il Blocco Funzione che provvederà a copiare i parametri dai registri al modulo H3.

## **Flag di stato**

Quando viene chiamato un FB, i flag di stato di quel particolare asse, vengono copiati in questi flag che possono essere interrogati dal programma utente. Per informazioni dettagliate su questi flag, fare riferimento al comando "FResSF".

## **Registri dei parametri di uscita**

I parametri letti dal modulo H3 vengono copiati in questi registri.

### Modulo H3

Si riferisce al controllore di un asse particolare all'interno del modulo H3.

#### Chiamata del blocco funzione:

Esempio: Per chiamare il FB "AxHndlg" per l'asse 2

CFB	AxHndlg	
	2	; Asse numero 2
	RA2	; Blocco di registri per l'asse 2
	FA2	; Blocco di flag per l'asse 2

### Struttura del programma e applicazione dei blocchi funzione "AxInit" e "AxHndlg"

Il modulo di programma utente H3 può essere suddiviso in tre parti:

- Inizializzazione
- Elaborazioni ripetitive
- Controllo del movimento

#### a) Inizializzazione

Il primo passo da eseguire all'interno di un Programma H3 è sempre l'inizializzazione degli assi. Ciascun asse deve essere inizializzato separatamente dagli altri utilizzando il Blocco Funzione "AxInit" richiamato dal modulo di partenza XOB 16. L'FB legge i valori precaricati nel blocco di registri. Per poter determinare il valore da caricare nei registri prima di chiamare "AxInit", occorre conoscere i dettagli del servomeccanismo; nel capitolo 9 "Esempi Applicativi" sono riportati tali dettagli. L'asse in questione potrà essere utilizzato solo dopo aver eseguito "AxInit".

#### b) Elaborazioni ripetitive

Tutte le attività eseguite regolarmente, fanno parte del ciclo di elaborazione. Tali attività vengono quindi programmate all'interno di un COB, che viene elaborato in modo ciclico. Il Blocco Funzione "AxHndlg" gestisce interamente lo scambio dati tra il programma e il controllore nel modulo H3. I flag di comando indicano all'FB quale attività deve essere eseguita, quindi l'FB interroga questi flag ogni volta in cui viene chiamato ed esegue i comandi indicati. È naturale quindi chiamare l'FB da un COB ciclico.

### c) Controllo Movimento

Poichè un controllo di movimento è sempre caratterizzato da una procedura sequenziale, il modo migliore per rappresentarlo è quello di utilizzare un programma strutturato in GRAFTEC.

In linea di principio, i movimenti sono sempre gli stessi e comprendono i seguenti passi:

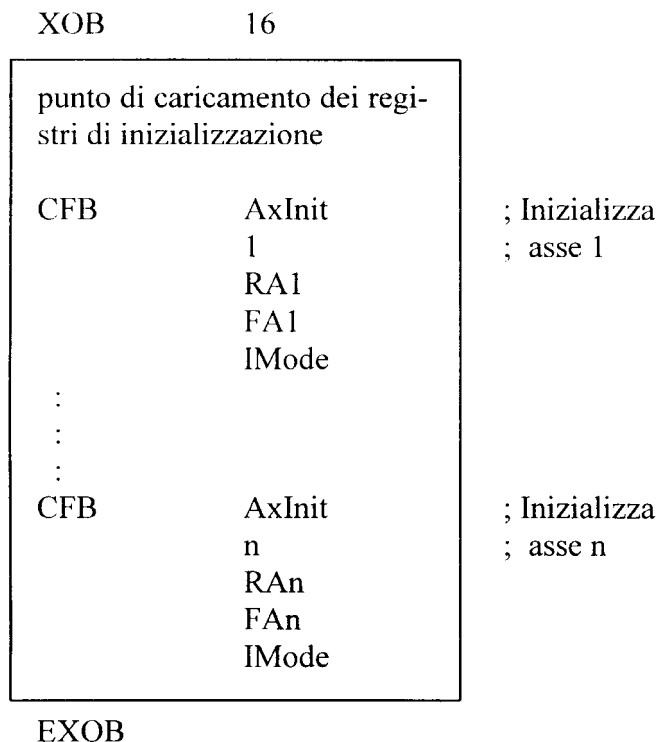
- inizio movimento asse (--> Step)
- attesa fino alla conclusione del movimento (--> Transizione)
- inizio prossimo movimento
- nuova attesa della conclusione
- e così via

Per ogni transizione non eseguita (in attesa che l'asse completi il proprio movimento) si esce dal programma GRAFTEC e si prosegue con l'esecuzione del programma ciclico.

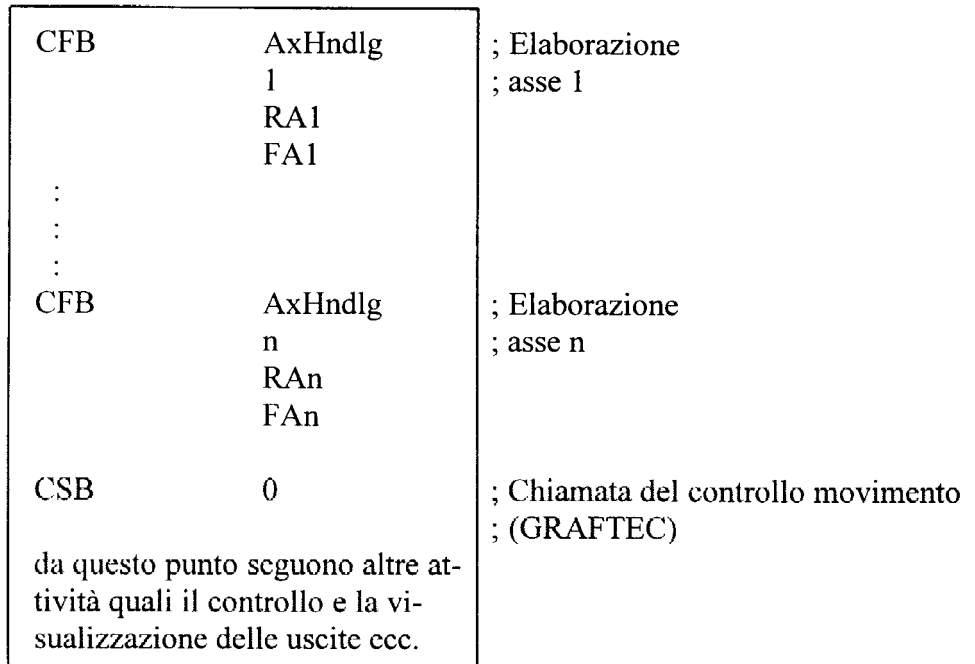
--> Chiamata dell' FB "AxHndlg" che esegue il lavoro definito nella struttura GRAFTEC.

(Caricamento dei parametri di movimento nel modulo H3 e avvio del movimento)

### Esempio di struttura di un programma H3:

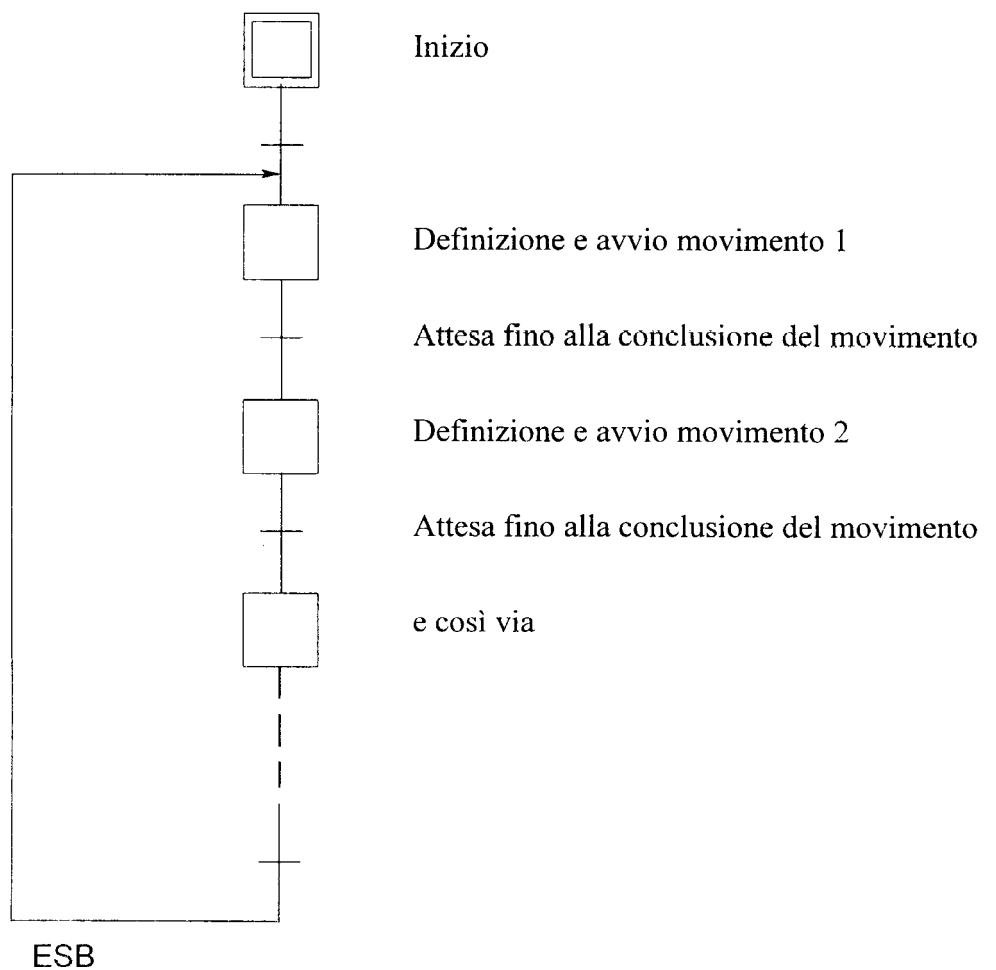


COB 0



ECOB

SB 0





### 7.3. Elenco dei comandi

---

I comandi vengono eseguiti dall'FB di gestione assi "AxHndlg". Il nome assegnato ai vari comandi, indica quale flag viene utilizzato per attivare il comando stesso.

#### Parametri operativi:

---

**FSelOM**    **Select Operation Mode** (Seleziona Modalità Operativa)  
**FSetPE**    **Set Position Error** (Imposta Errore di Posizione)

#### Parametri del profilo velocità

---

**FLdDA**    **Load Destination Absolute** (Carica pos. Assoluta di Dest.)  
**FLdDR**    **Load Destination Relative** (Carica pos. Relativa di Dest.)  
**FLdVA**    **Load Velocity Absolute** (Carica Velocità Assoluta)  
**FLdVR**    **Load Velocity Relative** (Carica Velocità Relativa)  
**FLdAA**    **Load Acceleration Absolute** (Carica Acceler. Assoluta)  
**FLdAR**    **Load Acceleration Relative** (Carica Acceler. Relativa)  
**FLdRP**    **Load Regulator Parameter** (Carica Parametri Regolatore)  
**FUpDRP**   **Up Date Regulator Parameter** (Aggiorna Param. Regol.)

#### Comandi di movimento

---

**FStart**    **Start motion** (Inizio movimento)  
**FStop**    **Stop motion** (Arresto movimento)  
**FMotOff**   **Motor off** (Disattivazione Motore)  
**FSSStepF**   **Single Step Forward** (Singolo Passo Avanti)  
**FSSStepB**   **Single Step Backwards** (Singolo Passo Indietro)  
**FForw**    **Forward with defined velocity** (Avanti a velocità definita)  
**FBackw**   **Backwards with defined velocity** (Indietro a vel. definita)

---

**Comandi di lettura dati**

---

<b>FRdAP</b>	<b>Read Actual Position</b> (Lettura Posizione Attuale)
<b>FRdSP</b>	<b>Read Setpoint Position</b> (Lettura Posizione Impostata)
<b>FRdAV</b>	<b>Read Actual Velocity</b> (Lettura Velocità Attuale)
<b>FRdSV</b>	<b>Read Setpoint Velocity</b> (Lettura Velocità Impostata)
<b>FRdITS</b>	<b>Read Integration Term Sum</b> (Lett. Somma di Integraz.)
<b>FRdIP</b>	<b>Read Index Position</b> (Lettura Posizione Indice)
<b>FRdSR</b>	<b>Read Signal Register</b> (Lettura Registro Segnali)

**Comandi vari**

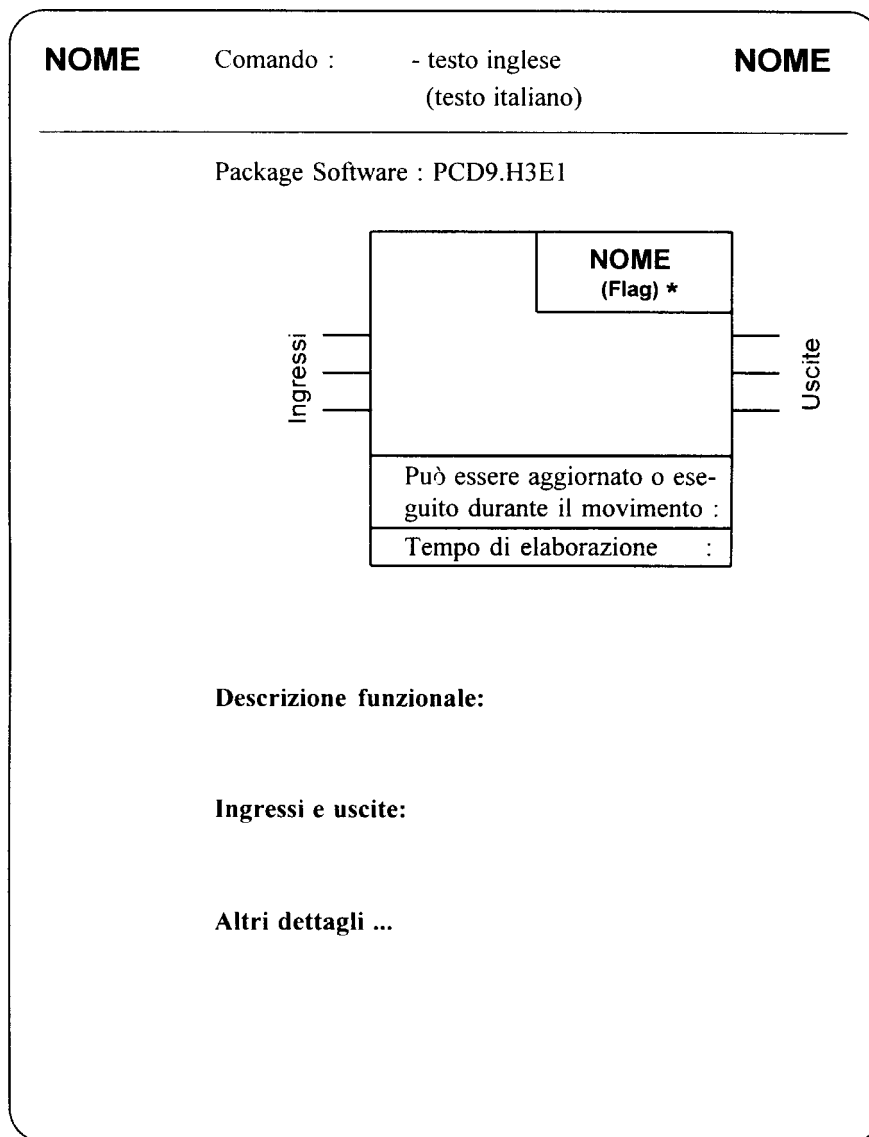
---

<b>FResSF</b>	<b>Reset Status Flag</b> (Azzera Flag di Stato)
<b>FLdBPA</b>	<b>Load Break Position Absolute</b> (Carica Breakpoint Assoluto)
<b>FLdBPR</b>	<b>Load Break Position Relative</b> (Carica Breakpoint Relativo)
<b>FSetIP</b>	<b>Set Index Position</b> (Imposta Posizione Indice)
<b>FsetZP</b>	<b>Set Zero Position</b> (Imposta Posizione Zero)

---

## 7.4 Descrizione dei comandi

Per semplicità, tutti i comandi vengono descritti utilizzando il medesimo formato:



I comandi descritti nelle pagine seguenti, rispettano l'ordine riportato nell'elenco dei comandi al capitolo 7.3.

Al termine del manuale viene riportato l'elenco di tutti i nomi e simboli in ordine alfabetico.

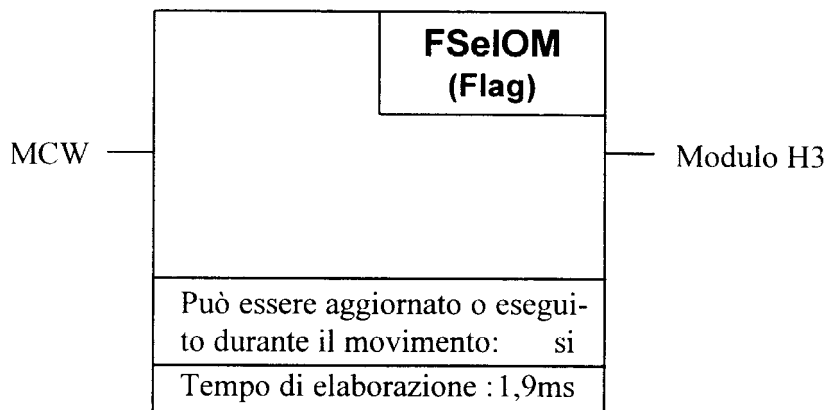
\*) Indica il nome del simbolo di flag; il comando viene eseguito impostando questo flag.

## FSeIOM

Comando: - **Select Operation Mode**  
(Seleziona Modalità Operativa)

## FSeIOM

Package Software :PCD9.H3E1

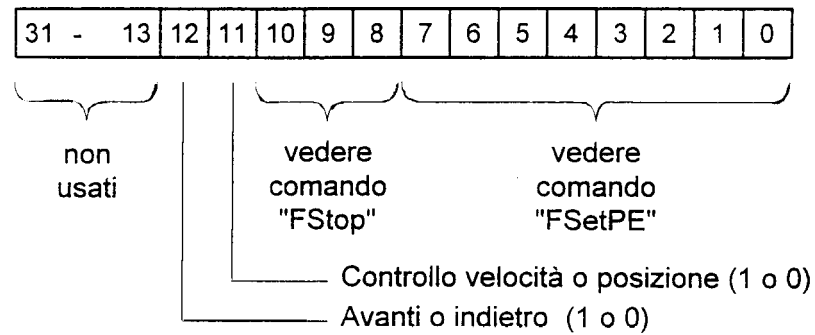


### Descrizione funzionale:

Questo comando definisce la modalità operativa dell'asse (controllo posizione o controllo velocità). La nuova modalità operativa definita viene adottata solo dopo l'esecuzione del prossimo comando "FStart".

### Descrizione degli ingressi e delle uscite:

Simbolo	Descrizione/Funzione	Para- metri	Dati		
			Tipo	Formato	Valore
MCW	<b>Motion Control Word</b> (Parola di Controllo Movimento)	no	R	Binario	vedere pagina seguente

**Significato della parola di controllo movimento "MCW":**

Il comando legge solo i bit 11 e 12 di "MCW"

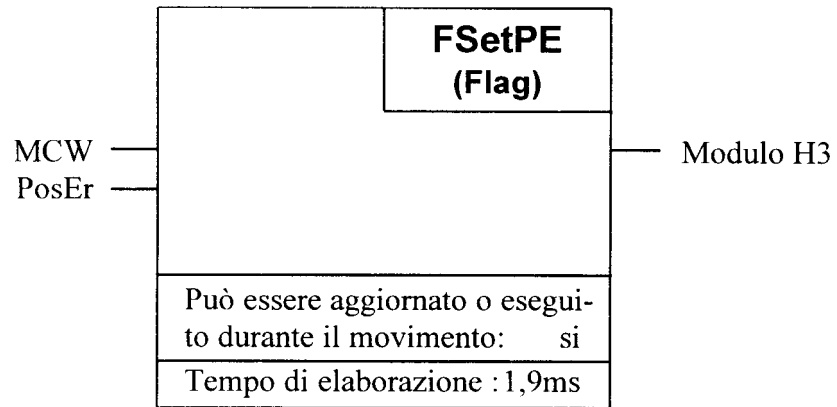
- Bit 11:      0 --> Controllo posizione  
               1 --> Controllo velocità
- Bit 12:      0 --> indietro  
               1 --> avanti

# FSetPE

Comando : - **Set Position Error**  
(Imposta Errore di Posizione)

# FSetPE

Package Software : PCD9.H3E1



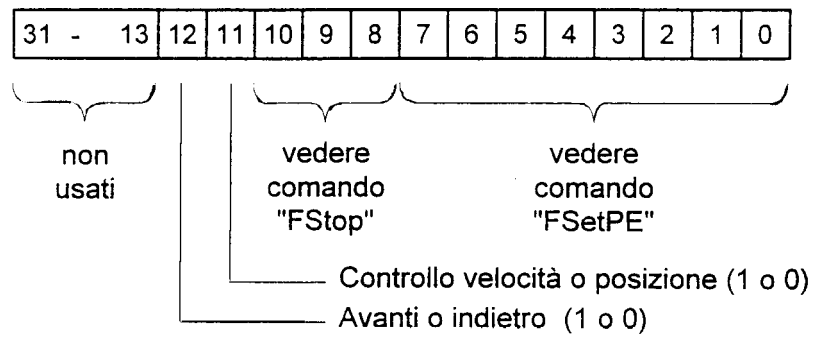
### Descrizione funzionale:

Questo comando carica il valore massimo consentito per quanto riguarda la differenza tra la posizione definita e la posizione effettiva. Se tale differenza raggiunge questo valore, viene impostato il flag di stato "ExcPEr". Il registro "MCW" definisce se viene impostato il solo flag di stato o se, oltre a questo viene spento il regolatore (uscita impostata al valore nullo).

Un errore di posizione denota un problema serio e deve quindi essere tenuto sotto controllo.

### Descrizione degli ingressi e delle uscite:

Simbolo	Descrizione/Funzione	Para- metri	Dati		
			Tipo	Formato	Valore
PosEr	<b>Position Error</b> (Errore di Posizione)	no	R	Intero	0.. 32'767 Imp.
MCW	<b>Motion Control Word</b> (Parola di Controllo Movimento)	no	R	Binario	vedere pagina seguente

**Significato della parola di controllo "MCW":**

a differenza tra il valore impostato e la posizione effettiva viene introdotta direttamente sotto forma di impulsi di encoder. Nel decodificatore di posizione, gli impulsi dell'encoder sono quadruplicati (vengono contati tutti i fronti degli impulsi).

Esempio:

Se la differenza deve essere al massimo di 500 impulsi di encoder, nel registro "PosEr" deve essere caricato un valore di  $4 \cdot 500 = 2000$  impulsi.

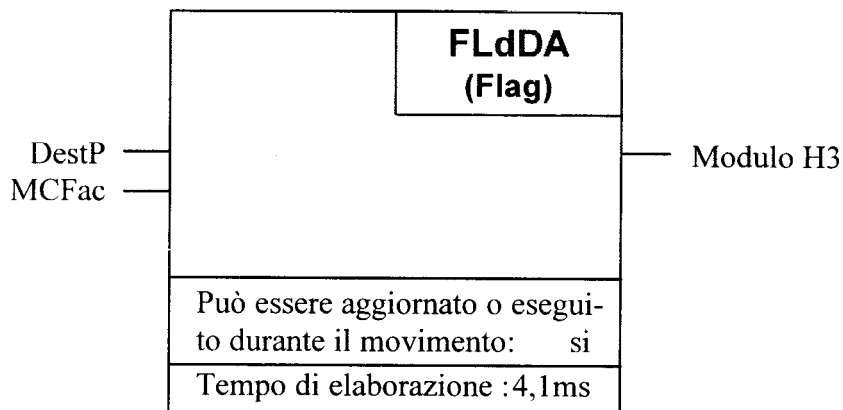
# FLdDA

Comando : - Load Destination Absolute

# FLdDA

(Carica Posizione Assoluta di Destinazione)

Package software : PCD9.H3E1



Descrizione funzionale:

Questo comando carica una nuova posizione assoluta di destinazione nel modulo H3. "Assoluta" significa che il valore è relativo alla posizione zero. La nuova posizione viene utilizzata solo alla ricezione del prossimo comando "FStart".

Descrizione degli ingressi e delle uscite:

Simbolo	Descrizione/Funzione	Para- metri	Dati		
			Tipo	Formato	Valore
DestP	<b>Destination Position</b> (Posizione di Destinazione)  Valore: $[-2^{30} \dots + (2^{30} - 1)]/k * 10^{-3}$ Unità: definita da k	no	R	Intero	--
MCFac	<b>Motion Control Factor</b> (Fattore di Controllo Movimento)	no	R	virgola mobile	$0..9,223371 * 10^{18}$



**Fattore di controllo k nel registro "MCFac":**

Il fattore k determina l'unità da utilizzare per introdurre la posizione di destinazione, la velocità e l'accelerazione. Questo fattore è calcolato partendo dalla risoluzione dell'encoder e dalla trasmissione meccanica. Il fattore k deve essere calcolato e caricato nel registro "MCFac". Tale registro è utilizzato da molti comandi per convertire le misure metriche in impulsi di encoder e viceversa.

La formula è: 
$$k = \frac{4 \cdot In}{s}$$

dove                      In:    impulsi per giro (risoluzione dell'encoder)  
                                   s :    distanza per giro (passo della vite e rapporto di trasmissione)

L'unità utilizzata per definire la distanza definisce anche le unità di misura per la posizione, la velocità e l'accelerazione

Esempio:

Vite con passo di 3mm; risoluzione dell'encoder = 1000 impulsi/giro.

Supponendo di voler raggiungere una posizione di destinazione di 60mm e di fornire il valore di posizione (e di risoluzione) in  $\mu\text{m}$ .

$$k = \frac{4 \cdot In}{s} = \frac{4 \cdot 1000 \text{ imp./giro}}{3000 \mu\text{m/giro}} = 1,33333 \text{ imp./}\mu\text{m}$$

Registro di ingresso "DestP" = 60000 $\mu\text{m}$

L'esempio sopra illustrato può essere utilizzato per fornire la posizione in unità di 1/10mm:

$$k = \frac{4 \cdot In}{s} = \frac{4 \cdot 1000 \text{ imp./giro}}{30 \text{ 1/10mm /giro}} = 133,333 \text{ imp./ 1/10mm}$$

Registro di ingresso "DestP" = 600 1/10mm

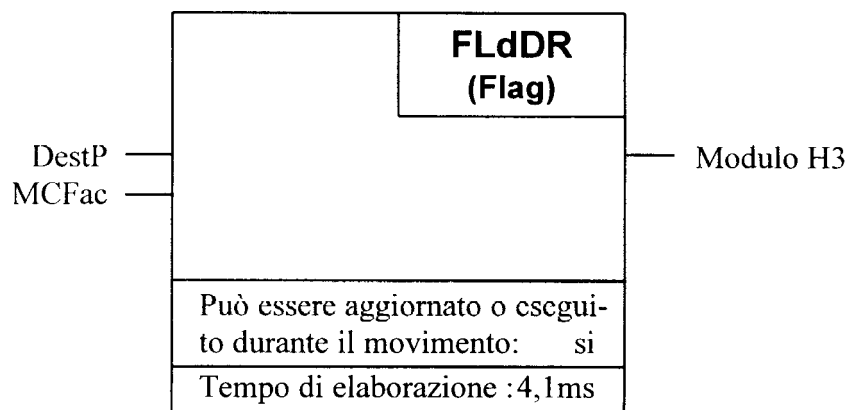
Nota:    Per introdurre la posizione in numero di impulsi, il valore di "k" è 1.0 e deve essere caricato nel registro "MCFac".

## FLdDR

Funzione : -Load Destination Relative  
(Carica Posizione Relativa di Destinazione)

## FLdDR

Package software :PCD9.H3E1



### Descrizione Funzionale:

Questo comando carica la posizione relativa nel modulo H3. "Relativa" significa che il valore si riferisce alla posizione di destinazione corrente. La nuova posizione viene utilizzata solo alla ricezione del comando "FStart".

### Descrizione degli ingressi e delle uscite:

Simbolo	Descrizione/Funzione	Para- metri	Dati		
			Tipo	Formato	Valore
DestP	<b>Destination Position</b> (Posizione di Destinazione)  Valore: $[-2^{30} \dots + (2^{30} - 1)]/k * 10^{-3}$ Unità: definita da k	no	R	Intero	--
MCFac	<b>Motion Control Factor</b> (Fattore di Controllo Movimento)	no	R	virgola mobile	$0..9,223371 * 10^{18}$

Dopo l'inizializzazione del modulo, per il primo movimento dell'asse, si può utilizzare solo una posizione di destinazione assoluta. Se si carica una posizione relativa, il regolatore H3 produce una segnalazione di "Command error" (errore di comando).

**Fattore di controllo k nel registro "MCFac":**

Il fattore k ha lo stesso significato descritto per la funzione "FLdDA".

Esempio:

Vite con passo di 3mm

Risoluzione dell'encoder = 1000 imp./giro

Supponendo di voler coprire una distanza relativa di -60 mm e di fornire il valore di posizione (e di risoluzione) in  $\mu\text{m}$ .

$$k = \frac{4 \cdot \text{In}}{s} = \frac{4 \cdot 1000 \text{ imp./giro}}{3000 \mu\text{m/giro}} = 1,33333 \text{ imp./}\mu\text{m}$$

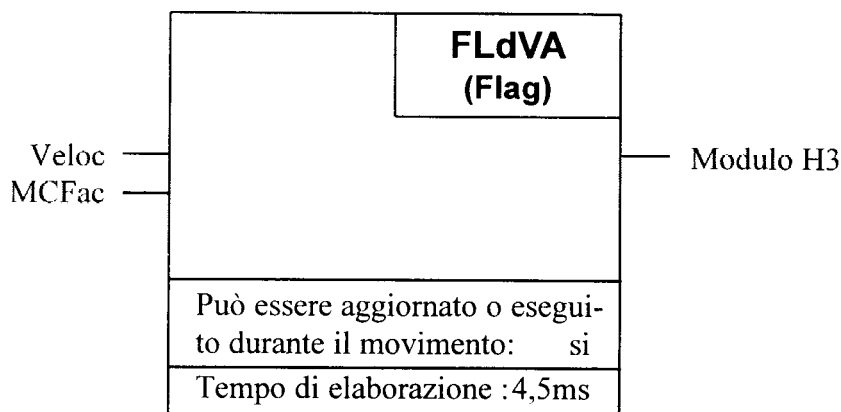
Registro di ingresso "DestP" = -60'000  $\mu\text{m}$

# FLdVA

Funzione : -Load Velocity Absolute  
(Carica Velocità Assoluta)

# FLdVA

Package Software : PCD9.H3E1



### Descrizione funzionale:

Questa funzione è utilizzata per caricare un valore di velocità assoluto in un registro intermedio del modulo H3. Caricamento Assoluto significa che il valore si riferisce alla velocità zero. Il modulo H3 adotta la nuova velocità nel registro di lavoro, al prossimo comando di start.

### Descrizione degli ingressi e delle uscite:

Simbolo	Descrizione/Funzione	Para- metri	Dati		
			Tipo	Formato	Valore
Veloc	<b>Velocità</b>  Valore: $[0..+(2^{30}-1)]/k*22348*10^{-6}$ Unità: definita da k	no	R	Intero	--
MCFac	<b>Motion Control Factor</b> (Fattore di Controllo Movimento)	no	R	virgola mobile	$0..9,223371*10^{18}$

**Fattore di controllo k nel registro "MCFac":**

Il fattore k ha lo stesso significato descritto per la funzione "FLdDA".

Si deve tener presente che questo fattore viene letto dallo stesso registro utilizzato anche per la posizione di destinazione, velocità e accelerazione. Si raccomanda quindi di utilizzare la stessa unità di misura per l'introduzione di questi parametri

Esempio:

Vite con passo di 3mm

Risoluzione dell'encoder = 1000 imp./giro

Supponendo di voler raggiungere una posizione di destinazione alla velocità di 0,1 m/s e di fornire il valore (e la risoluzione) in mm/s.

$$k = \frac{4 \cdot I_n}{s} = \frac{4 \cdot 1000 \text{ imp./giro}}{3000 \text{ } \mu\text{m/giro}} = 1,33333 \text{ imp./} \mu\text{m}$$

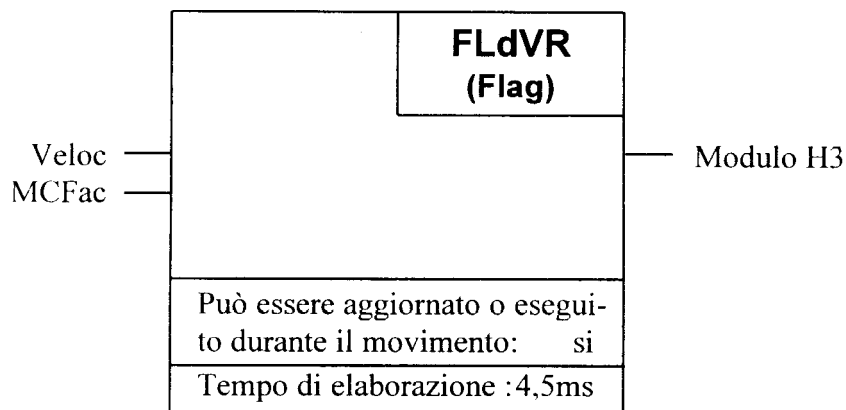
Registro di ingresso "Veloc" = 100 mm/s

# FLdVR

Funzione :- **Load Velocity Relative**  
(Carica Velocità Relativa)

# FLdVR

Package software :PCD9.H3E1



### Descrizione funzionale:

Questa funzione è utilizzata per caricare un valore di velocità relativo in un registro intermedio del modulo H3. Caricamento Relativo significa che il valore si riferisce alla velocità nominale corrente. Il modulo H3 adotta la nuova velocità nel registro di lavoro, al prossimo comando di start.

### Descrizione degli ingressi e delle uscite:

Simbolo	Descrizione/Funzione	Parametri	Dati		
			Tipo	Formato	Valore
Veloc	<b>Velocità</b>  Valore: $[-2^{30} \dots + (2^{30} - 1)]/k * 22348 * 10^{-6}$ Unità: definita da k	no	R	Intero	--
MCFac	<b>Motion Control Factor</b> (Fattore di Controllo Movimento)	no	R	virgola mobile	$0.9,223371 * 10^{18}$

Dopo l'inizializzazione del modulo, per il primo movimento dell'asse, si può utilizzare solo una velocità assoluta. Se si carica una velocità relativa, il controllore H3 produce una segnalazione di "Command error" (errore di comando).

**Fattore di controllo k nel registro "MCFac":**

Il fattore k ha lo stesso significato descritto per la funzione "FLdDA".

Si deve tener presente che questo fattore viene letto dallo stesso registro utilizzato anche per la posizione di destinazione, velocità e accelerazione. Si raccomanda quindi di utilizzare la stessa unità di misura per l'introduzione di questi parametri.

Esempio:

Vite con passo di 3mm

Risoluzione dell'encoder = 1000 imp./giro

Supponendo di voler raggiungere una velocità relativa di -0,1 m/s e di fornire il valore (e la risoluzione) in mm/s.

$$k = \frac{4 \cdot I_n}{s} = \frac{4 \cdot 1000 \text{ imp./giro}}{3 \text{ mm/giro}} = 1333,3 \text{ imp./mm}$$

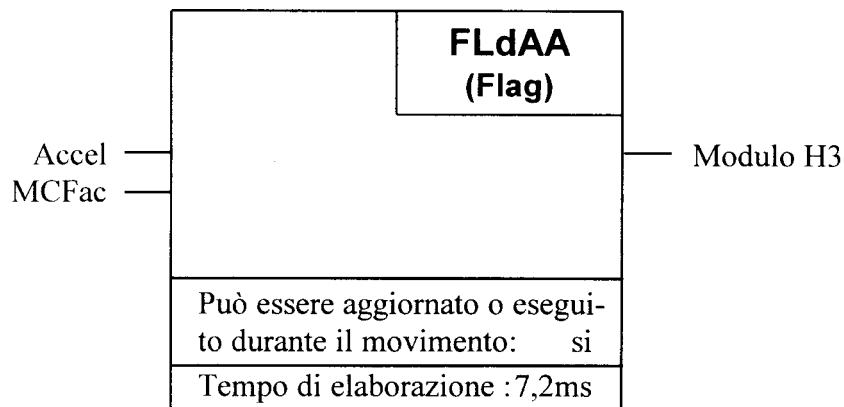
Registro di ingresso "Veloc" = -100 mm/s

**FLdAA**

Funzione :- **Load Acceleration Absolute**  
(Carica Accelerazione Assoluta)

**FLdAA**

Package software :PCD9.H3E1



**Descrizione funzionale:**

Questa funzione è utilizzata per caricare una accelerazione assoluta in un registro intermedio del modulo H3. Caricamento assoluto significa che il valore si riferisce alla accelerazione zero. Il modulo H3 adotta il nuovo valore di accelerazione nel registro di lavoro, al prossimo comando di start. Anche se questa funzione può essere eseguita durante un movimento, il comando "FStart" potrà far funzionare il modulo H3 con la nuova accelerazione solo dopo il completamento del movimento in corso (oppure come risultato di un comando di stop).

NB: Quando viene invocata questa funzione, prima di caricare l'accelerazione viene eseguita la funzione "FMotOff".

- > Dopo l'esecuzione della funzione, viene spento il regolatore.
- > Il regolatore viene riaccessato con "FStart".

**Descrizione degli ingressi e delle uscite:**

Simbolo	Descrizione/Funzione	Para- metri	Dati		
			Tipo	Formato	Valore
Accel	<b>Acceleration</b> (Accelerazione) Valore: $[0..+(2^{-30})]/k*76206*10^{-9}$ Unità: definita da k	no	R	Intero	--
MCFac	<b>Motion Control Factor</b> (Fattore di Controllo Movimento)	no	R	virgola mobile	$0..9,223371*10^{18}$



**Fattore di controllo k nel registro "MCFac":**

Il fattore k ha lo stesso significato descritto per la funzione "FLdDA".

Si deve tener presente che questo fattore viene letto dallo stesso registro utilizzato anche per la posizione di destinazione, velocità e accelerazione. Si raccomanda quindi di utilizzare la stessa unità di misura per l'introduzione di questi parametri.

Esempio:

Vite con passo di 3 mm

Risoluzione dell'encoder = 1000 imp./giro

Supponendo di voler raggiungere una accelerazione di 0,005 m/s<sup>2</sup> e di fornire il valore (e la risoluzione) in mm/s<sup>2</sup>.

$$k = \frac{4 \cdot I_n}{s} = \frac{4 \cdot 1000 \text{ imp./giro}}{3 \text{ mm/giro}} = 1333,3 \text{ imp./mm}$$

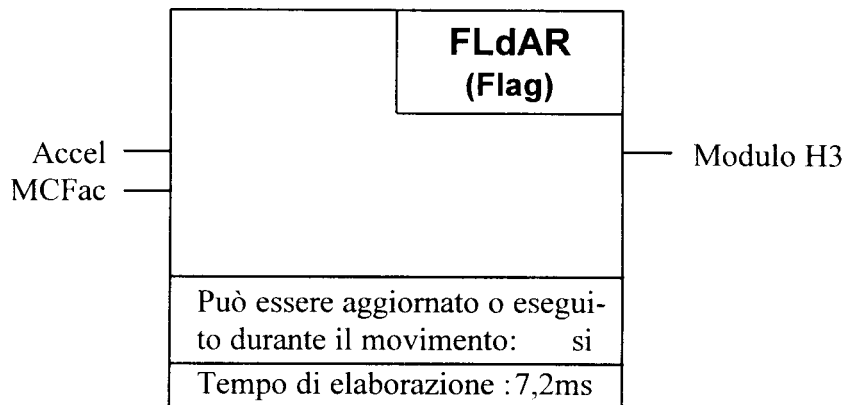
Registro di ingresso "Accel" = 5 mm/s<sup>2</sup>

# FLdAR

Funzione :- **Load Acceleration Relative**  
(Carica Accelerazione Relativa)

# FLdAR

Package software :PCD9.H3E1



### Descrizione funzionale:

Questa funzione è utilizzata per caricare una accelerazione relativa in un registro intermedio del modulo H3. Caricamento relativo significa che il valore si riferisce alla accelerazione nominale corrente. Il modulo H3 adotta il nuovo valore di accelerazione nel registro di lavoro, al prossimo comando di start. Anche se questa funzione può essere eseguita durante un movimento, il comando "FStart" potrà far funzionare il modulo H3 con la nuova accelerazione solo dopo il completamento del movimento in corso (oppure come risultato di un comando di stop).

**NB:** Quando viene invocata questa funzione, prima di caricare l'accelerazione viene eseguita la funzione "FMofOff".

- > Dopo l'esecuzione della funzione, viene spento il regolatore
- > Il regolatore viene riaccessato con "FStart".

### Descrizione degli ingressi e delle uscite:

Simbolo	Descrizione/Funzione	Para- metri	Dati		
			Tipo	Formato	Valore
Accel	<b>Acceleration</b> (Accelerazione) Valore: $[-2 \dots + (2 \dots - 1)] / k * 76206 * 10^{-9}$ Unità: definita da k	no	R	Intero	--
MCFac	<b>Motion Control Factor</b> (Fattore di Controllo Movimento)	no	R	virgola mobile	$0.9,223371 * 10^{18}$

Dopo l'inizializzazione del modulo, per il primo movimento dell'asse, si può utilizzare solo una accelerazione relativa. Se si carica una accelerazione relativa, il controllore H3 produce una segnalazione di "Command error" (errore di comando).

**Fattore di controllo k nel registro "MCFac":**

Il fattore k ha lo stesso significato descritto per la funzione "FLdDA".

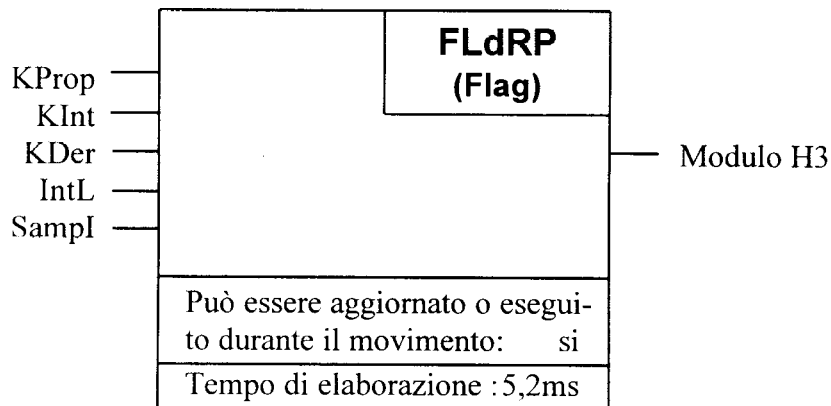
Si deve tener presente che questo fattore viene letto dallo stesso registro utilizzato anche per la posizione di destinazione, velocità e accelerazione. Si raccomanda quindi di utilizzare la stessa unità di misura per l'introduzione di questi parametri.

## FLdRP

Funzione : - **Load Regulator Parameter**  
(Carica Parametri Regolatore)

## FLdRP

Package software :PCD9.H3E1



### Descrizione funzionale:

Questa funzione è utilizzata per caricare i parametri del regolatore dai registri dell'asse in un registro intermedio del modulo H3. Il regolatore adotta questi valori nel registro di lavoro dopo che è stata eseguita la funzione "FUpDRP".

### Descrizione degli ingressi e delle uscite:

Simbolo	Descrizione/Funzione	Para- metri	Dati		
			Tipo	Formato	Valore
KProp	<b>Proportional Factor</b> (Fattore Proporzionale)	no	R	Intero	0.. 32767
KInt	<b>Integral Factor</b> (Fattore Integrale)	no	R	Intero	0.. 32767
KDer	<b>Derivative Factor</b> (Fattore Derivato)	no	R	Intero	0.. 32767
IntL	<b>Integration Limit</b> (Limite di Integrazione)	no	R	Intero	0.. 32767
SampI	<b>Sampling Interval</b> derivative term (Derivata dell'intervallo di Campionamento)	no	R	Intero	0.. 255

L'intervallo di campionamento del termine derivativo può essere programmato in passi di 341,33  $\mu$ s.

Intervallo di campionamento =  $(n+1) * 341,33 \mu$ s

Il valore viene caricato nel registro "SampI".

Esempio: Volendo un intervallo di campionamento di 1024  $\mu$ s

--> Register "SampI" = 2

L'intervallo di campionamento per i termini proporzionale e integrale è di 341,33  $\mu$ s e non può essere programmato.

Se nel registro si carica un valore fuori dal limite consentito, viene generato un errore all'atto dell'esecuzione della funzione.

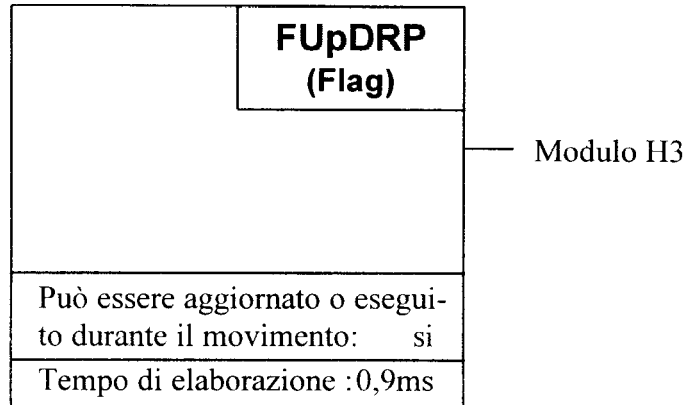
## FUpDRP

Funzione : -Up Date Regulator Parameter  
(Aggiorna i Parametri del Regolatore)

## FUpDRP

---

Package software :PCD9.H3E1



### Descrizione funzionale:

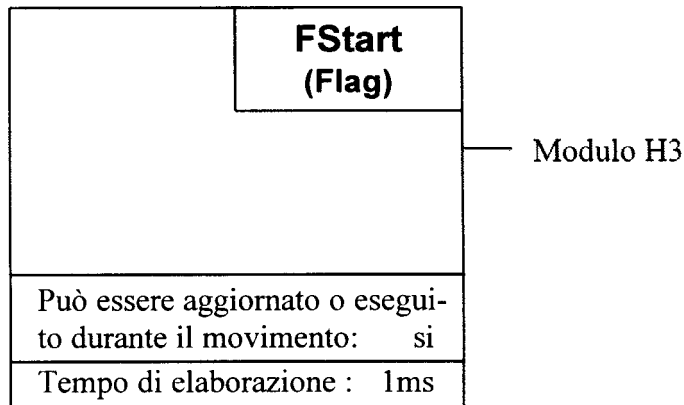
Questa funzione è utilizzata per aggiornare i parametri del regolatore. Il regolatore del modulo H3 adotta i parametri caricati con la funzione "FLdRP" prelevandoli dal registro intermedio e copiandoli nei registri di lavoro.

# FStart

Funzione : -Start motion  
(Inizio movimento)

# FStart

Package software: PCD9.H3E1

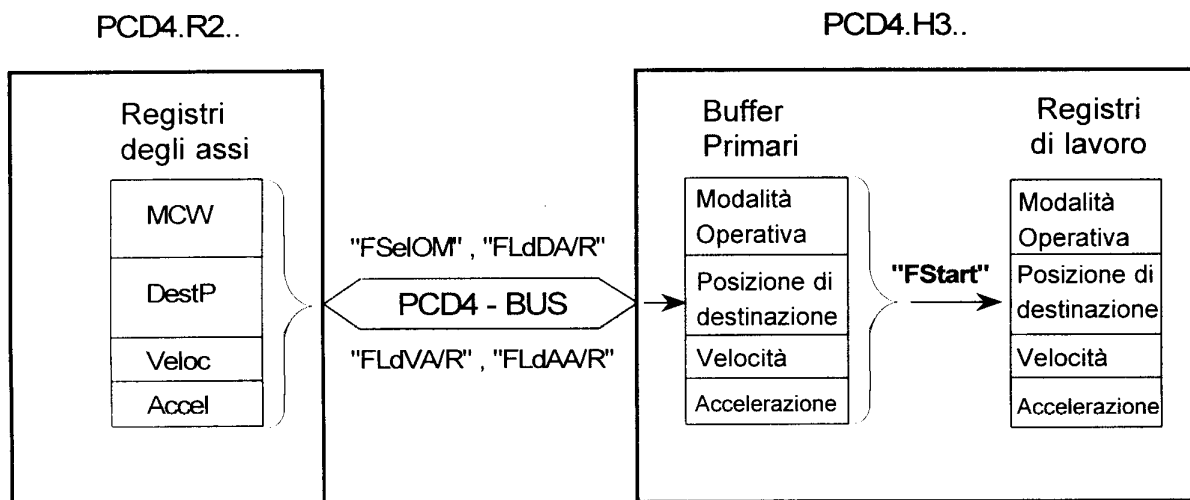


### Descrizione funzionale:

Questa funzione può essere usata per avviare un movimento o per consentire al modulo H3 di lavorare con un nuovo parametro di movimento (es. velocità).

Se durante il movimento viene caricata una nuova accelerazione, il comando di start potrà essere eseguito solo dopo la conclusione del movimento in corso.

Lo schema seguente illustra quali sono i parametri di movimento copiati nel registro di lavoro del modulo H3 dopo un comando di start.

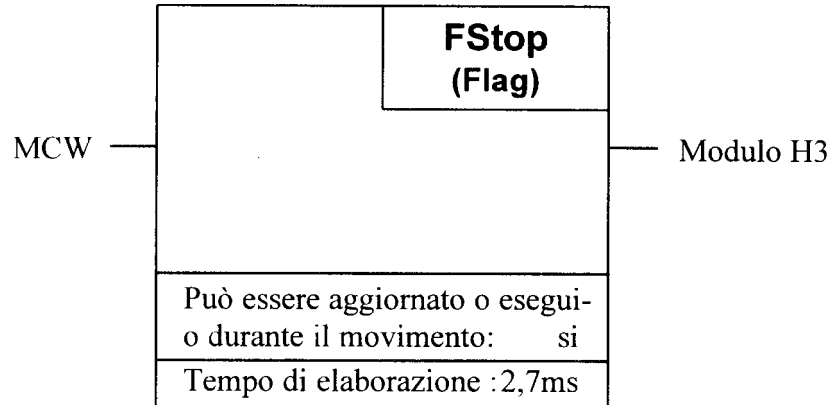


# FStop

Funzione : -**Stop** motion  
(Arresto movimento)

# FStop

Package software :PCD9.H3E1



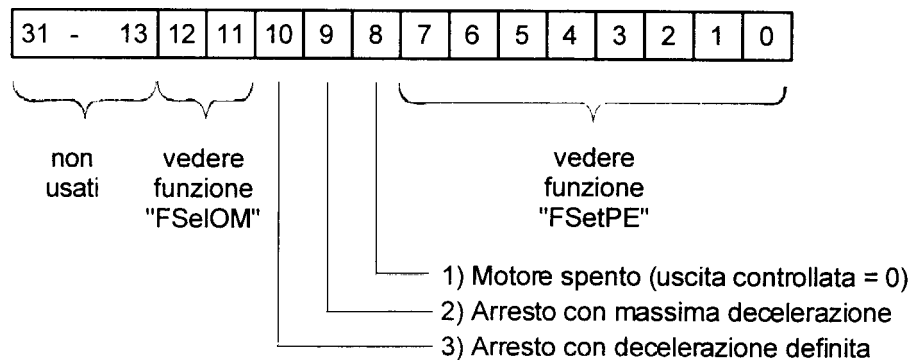
### Descrizione funzionale:

Questa funzione può essere usata per arrestare un movimento in un istante qualsiasi. Il tipo di arresto dipende dalla definizione fornita nel registro "MCW". Dopo l'esecuzione di una funzione di arresto viene impostato il flag "OnDest".

### Descrizione degli ingressi e delle uscite:

Simbolo	Descrizione/Funzione	Para- metri	Dati		
			Tipo	Formato	Valore
MCW	<b>Motion Control Word</b> (Parola di Controllo Movimento)	no	R	Binario	vedere pagina



**Significato della parola di controllo movimento "MCW":**

La funzione legge solo i bit 8 - 9 - 10 del registro "MCW"

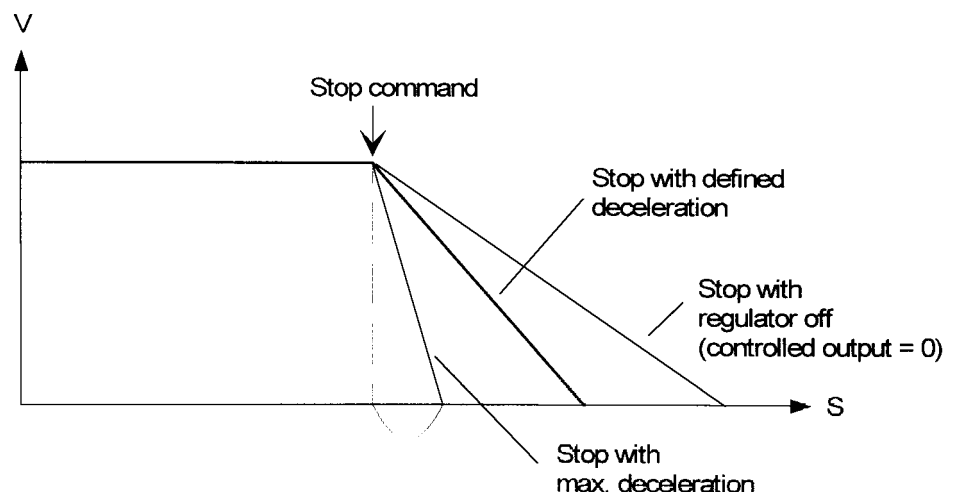
Bit 8 = "1" In caso di comando di stop, il regolatore viene spento quindi l'uscita controllata viene posta a zero.

Bit 9 = "1" In caso di comando di stop, si utilizza la massima decelerazione di frenata impostando la posizione definita uguale alla posizione corrente nel registro di lavoro del controllore H3.

Bit 10 = "1" In caso di comando di stop, viene utilizzata la decelerazione di frenata definita (= accelerazione negativa).

In qualsiasi momento, solo uno di questi tre bit può essere attivato ("1"). Dopo un arresto, il controllore non perde la posizione di destinazione che aveva in precedenza. Tuttavia, prima di poter continuare un movimento interrotto senza dover caricare un nuovo parametro (destinazione, velocità o accelerazione) è necessario ricaricare il comando di modalità operativa ("FSelOM").

Il grafico seguente illustra i tre tipi di arresto.



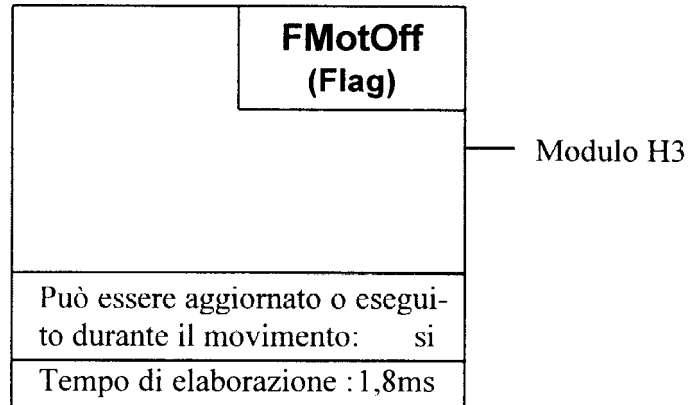
## FMotOff

Funzione : -Motor Off  
(Disattivazione Motore)

## FMotOff

---

Package software :PCD9.H3E1



### Descrizione funzionale:

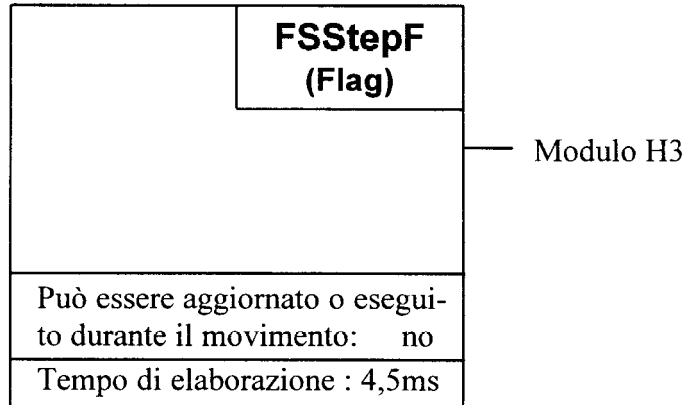
Questa funzione è utilizzata per spegnere il regolatore e quindi l'uscita controllata viene posta a zero. Il comando "FMotOff" ha la stessa funzione del comando "FStop" eseguito con il Bit 8 del registro "MCW" attivo.

**FSStepF**

Funzione : -Single Step Forward  
(Singolo Passo Avanti)

**FSStepF**

Package software :PCD9.H3E1

**Descrizione funzionale:**

Questa funzione è utilizzata per comandare un avanzamento in direzione positiva di una quantità corrispondente ad un singolo impulso. La posizione di destinazione viene incrementata di 1 impulso e viene comandato il movimento. Durante l'esecuzione della funzione, occorre prestare attenzione al valore della posizione di destinazione. La funzione infatti potrebbe non essere eseguita se la posizione assoluta di destinazione ha già raggiunto il limite massimo positivo.

NB: 1 impulso corrisponde a 1/4 di passo dell'encoder  
(l'impulso viene quadruplicato dal decodificatore di posizione).

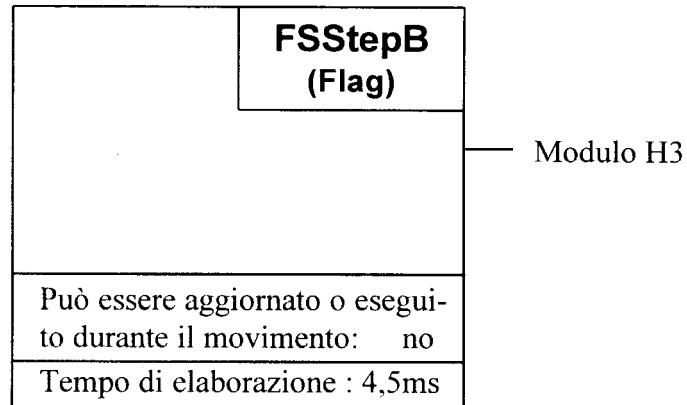
## FSStepB

Funzione : - **Single Step Backwards**  
(Singolo Passo Indietro)

## FSStepB

---

Package software :PCD9.H3E1



### Descrizione funzionale:

Questa funzione è utilizzata per comandare un avanzamento in direzione negativa di una quantità corrispondente ad un singolo impulso. La posizione di destinazione viene decrementata di 1 impulso e viene comandato il movimento. Durante l'esecuzione della funzione, occorre prestare attenzione al valore della posizione di destinazione. La funzione infatti potrebbe non essere eseguita se la posizione assoluta di destinazione ha già raggiunto il limite massimo negativo.

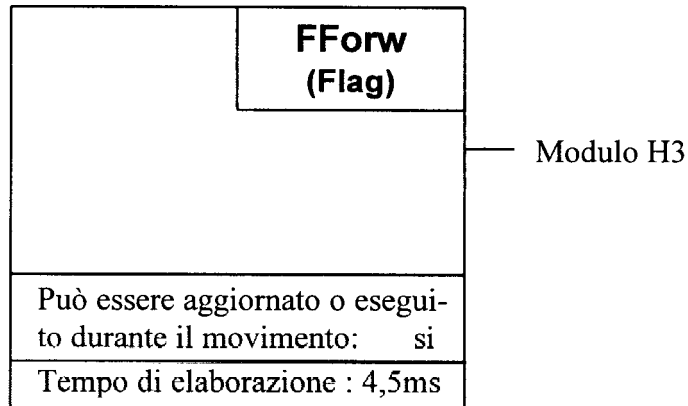
NB: 1 impulso corrisponde a 1/4 di passo dell'encoder (quadruplicato dal decodificatore di posizione).

**FForw**

Funzione : -**Forward** with defined velocity  
(Avanti a velocità definita)

**FForw**

Package software: PCD9.H3E1

**Descrizione funzionale:**

Questa funzione è utilizzata per procedere in direzione positiva alla velocità definita in precedenza. Per arrestare il movimento è richiesto un comando di arresto manuale.

La funzione viene eseguita caricando la più elevata posizione di destinazione positiva e quindi eseguendo il comando di inizio movimento.

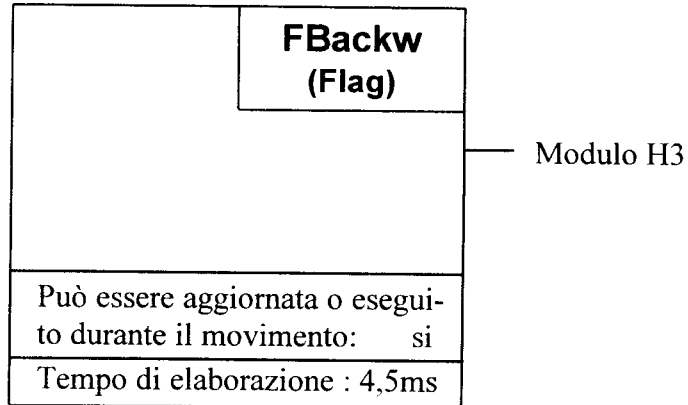
## FBackw

Funzione : - **Backwards** with defined velocity  
(Indietro a velocità definita)

## FBackw

---

Package software: PCD9.H3E1



### Descrizione funzionale:

Questa funzione è utilizzata per procedere in direzione negativa alla velocità definita in precedenza. Per arrestare il movimento è richiesto un comando di arresto manuale.

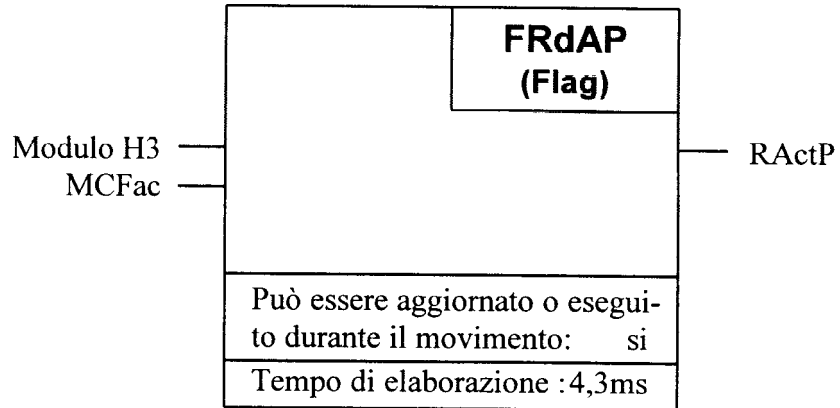
La funzione viene eseguita caricando la più bassa posizione di destinazione negativa e quindi eseguendo il comando di inizio movimento.

**FRdAP**

Funzione : **-Read Actual Position**  
(Lettura Posizione Attuale)

**FRdAP**

Package software :PCD9.H3E1

**Descrizione funzionale:**

Questa funzione legge la posizione corrente dal modulo H3 e la copia nel registro "RActP".

**Descrizione degli ingressi e delle uscite:**

Simbolo	Descrizione/Funzione	Para- metri	Dati		
			Tipo	Formato	Valore
RActP	<b>Register Actual Position</b> (Registro di Posizione Attuale)  Valore: $[-2^{30} \dots + (2^{30} - 1)]/k * 10^{-3}$ Unità: definita da k	no	R	Intero	--
MCFac	<b>Motion Control Factor</b> (Fattore di Controllo Movimento)	no	R	virgola mobile	$0..9,223371 * 10^{18}$

**Fattore di controllo k nel registro "MCFac":**

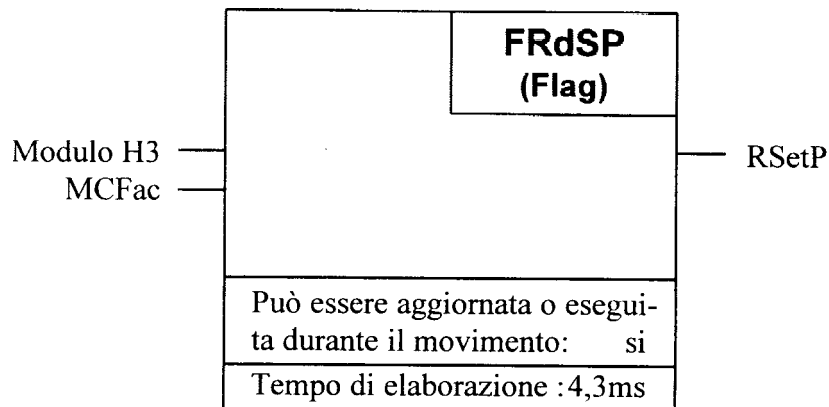
Il fattore k ha lo stesso significato descritto per la funzione "FLdDA". Il registro "MCFac" viene letto questa funzione e viene utilizzato per convertire la posizione da un numero di impulsi ad una misura metrica.

**FRdSP**

Funzione : -Read Setpoint Position  
(Lettura Posizione Impostata)

**FRdSP**

Package software :PCD9.H3E1



**Descrizione funzionale:**

Questa funzione legge la posizione corrente sull'uscita del generatore di profilo velocità e la copia nel registro "RSetP". La differenza tra posizione impostata e posizione effettiva viene fornita al regolatore PID.

**Descrizione degli ingressi e delle uscite:**

Simbolo	Descrizione/Funzione	Parametri	Dati		
			Tipo	Formato	Valore
RSetP	<b>Register Setpoint Position (Registro Posizione attuale)</b>  Valore: $[-2^{30} \dots + (2^{30} - 1)]/k * 10^{-3}$ Unità: definita da k	no	R	Intero	--
MCFac	<b>Motion Control Factor (Fattore di Controllo Movimento)</b>	no	R	virgola mobile	$0..9,223371 * 10^{18}$

**Fattore di controllo k nel registro "MCFac":**

Il fattore k ha lo stesso significato descritto per la funzione "FLdDA". Il registro "MCFac" viene letto da questa funzione e viene utilizzato per convertire la posizione da un numero di impulsi ad una misura metrica.

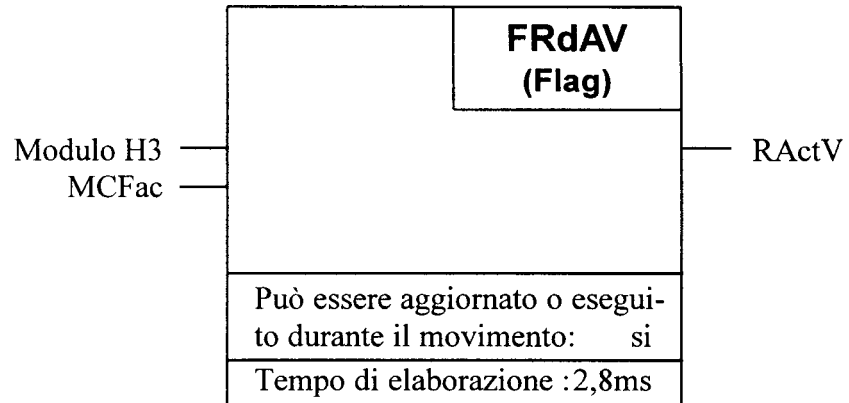


**FRdAV**

Funzione : -Read Actual Velocity  
(Lettura Velocità Attuale)

**FRdAV**

Package software :PCD9.H3E1

**Descrizione funzionale:**

Questa funzione legge la velocità effettiva dell'asse dal modulo H3 e la co-pia nel registro "RActV".

Il controllore può leggere comunque solo i 14 bit più significativi nel modulo H3. Per basse velocità, quindi, il valore letto non è attendibile per cui si consiglia di leggere la velocità di setpoint anzichè la velocità effettiva.

**Descrizione degli ingressi e delle uscite:**

Simbolo	Descrizione/Funzione	Para- metri	Dati		
			Tipo	Formato	Valore
RActV	<b>Register Actual Velocity</b> (Registro Velocità Effettiva)  Valore: $[0..+(2^{30}-1)]/k*22348*10^{-6}$ Unità: definita da k	no	R	Intero	--
MCFac	<b>Motion Control Factor</b> (Fattore di Controllo Movimento)	no	R	virgola mobile	$0..9,223371*10^{18}$

**Fattore di controllo k nel registro "MCFac":**

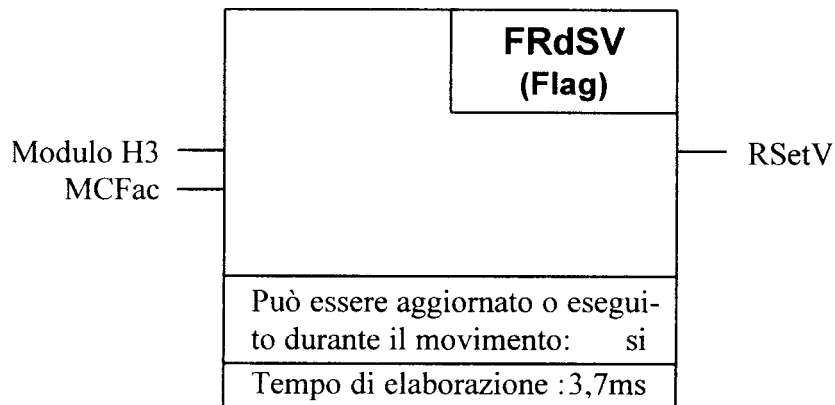
Il fattore k ha lo stesso significato descritto per la funzione "FLdDA". Il registro "MCFac" viene letto da questa funzione e viene utilizzato per convertire la posizione da un numero di impulsi ad una misura metrica.

**FRdSV**

Funzione : -Read Setpoint Velocity  
(Lettura Velocità Impostata)

**FRdSV**

Package software :PCD9.H3E1



**Descrizione funzionale:**

Questa funzione legge la velocità di setpoint corrente dal generatore di profilo e la copia nel registro "RSetV".

**Descrizione degli ingressi e delle uscite:**

Simbolo	Descrizione/Funzione	Para- metri	Dati		
			Tipo	Formato	Valore
RSetV	<b>Register Setpoint Velocity (Registro Velocità di Setpoint)</b>  Valore: $[0..+(2^{30}-1)]/k*22348*10^{-6}$ Unità: definita da k	no	R	Intero	--
MCFac	<b>Motion Control Factor (Fattore di Controllo Movimento)</b>	no	R	virgola mobile	$0..9,223371*10^{18}$

**Fattore di controllo k nel registro "MCFac":**

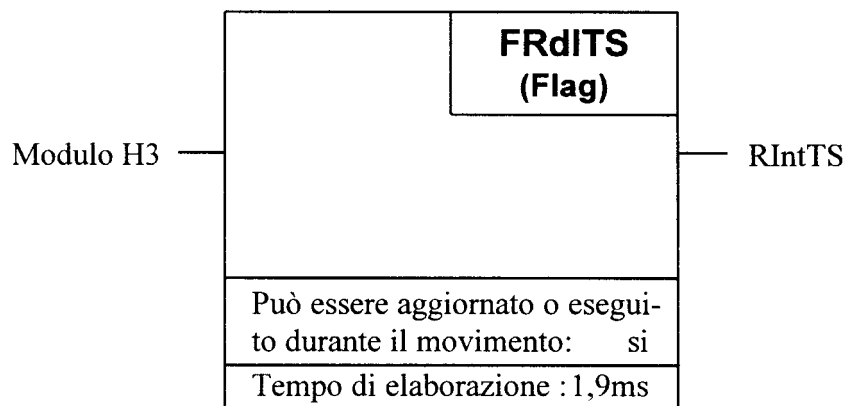
Il fattore k ha lo stesso significato descritto per la funzione "FLdDA". Il registro "MCFac" viene letto da questa funzione e viene utilizzato per convertire la posizione da un numero di impulsi ad una misura metrica.

**FRdITS**

Funzione : **-Read Integration Term Sum**  
(Lettura Somma di Integrazione)

**FRdITS**

Package software: PCD9.H3E1

**Descrizione funzionale:**

Questa funzione legge il valore dell'integrale  $e(n)$  dal regolatore PID e la copia nel registro "RIntS". La funzione è utilizzata soprattutto per la ricerca dei parametri del regolatore durante la messa a punto.

**Descrizione degli ingressi e delle uscite:**

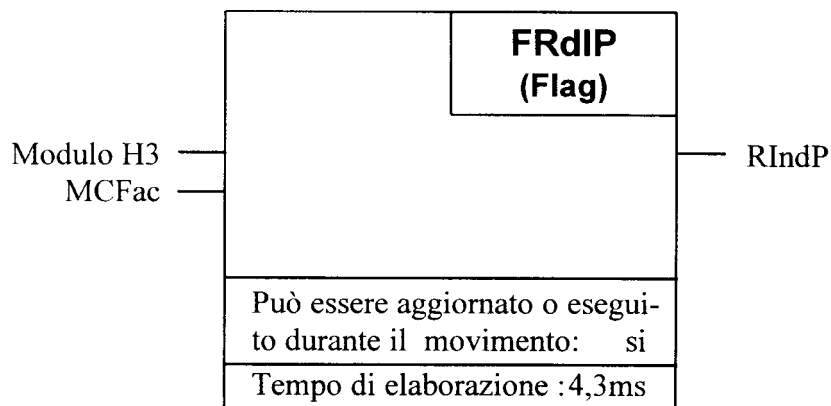
Simbolo	Descrizione/Funzione	Para- metri	Dati		
			Tipo	Formato	Valore
RIntTS	<b>Register Integration Term Sum</b> (Registro Somma di Integrazione)  Valore: Il valore è compreso nel range di integrazione definito con la funzione "FLdRP" (reg. "IntL")	no	R	Intero	--

**FRdIP**

Funzione : -**Read Index Position**  
(Lettura Posizione Indice)

**FRdIP**

Package software :PCD9.H3E1



**Descrizione funzionale:**

Questa funzione legge la posizione dell'indice dal modulo H3 e la copia nel registro "RIndP" (vedere anche la funzione "FSetIP").

**Descrizione degli ingressi e delle uscite:**

Simbolo	Descrizione/Funzione	Para- metri	Dati		
			Tipo	Formato	Valore
RIndP	<b>Register Index Position</b> (Registro Posizione Indice)  Valore: $[(-2^{30} ..+(2^{30}-1)]/k*10^{-3}$ Unità: definita da k	no	R	Intero	--
MCFac	<b>Motion Control Factor</b> (Fattore di Controllo Movimento)	no	R	virgola mobile	$0..9,223371*10^{18}$

**Fattore di controllo k nel registro "MCFac":**

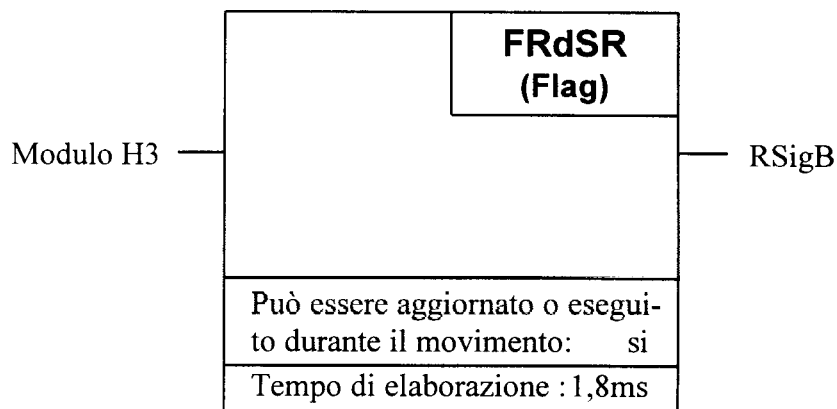
Il fattore k ha lo stesso significato descritto per la funzione "FLdDA". Il registro "MCFac" viene letto da questa funzione e viene utilizzato per convertire la posizione da un numero di impulsi ad una misura metrica.

**FRdSR**

Funzione : -Read Signal Register  
(Lettura Registro Segnali)

**FRdSR**

Package software :PCD9.H3E1

**Descrizione funzionale:**

Questa funzione è utilizzata per leggere il registro segnali di un asse dal modulo H3.

**Descrizione degli ingressi e delle uscite:**

Simbolo	Descrizione/Funzione	Parametri	Dati		
			Tipo	Formato	Valore
RSigB	<b>Register Signalisation Bits</b> (Registro dei Bit Segnalazione)	no	R	Intero	vedere pagina seguente

Il significato dei singoli bit del registro "RSigB" è il seguente:

- Bit 0: Impostato al valore "1" dopo l'esecuzione della funzione "SetIP" (set index position). Il bit viene ripristinato dopo l'introduzione di una successiva posizione dell'indice (impulso indice).
- Bit 1: Non usato
- Bit 2 bit 6: Riportano la condizione dei Flag di Stato (vedere la funzione "FResSF")
- Bit 7: É a "1" se il regolatore è spento (uscita controllata = 0).  
Il regolatore può essere spento dai seguenti eventi:  
- accensione dell'alimentatore  
- dopo l'esecuzione di un blocco funzione "AxInit"  
- in caso di errore di posizione (se questo è stato definito)  
- esecuzione della funzione "MotOff" (comando di spegnimento motore)  
- esecuzione della funzione "FStop" (se è stato definito il tipo di stop)  
Il bit viene ripristinato dal prossimo comando di start ("FStart").
- Bit 8: É a "1" all'accensione dell'alimentatore oppure nel caso in cui l'uscita controllata sia stata definita come uscita PWM dal blocco funzione "AxInit". Il bit è a "0" nel caso in cui l'uscita sia stata definita da "AxInit".
- Bit 9: Indica la procedura definita in caso di errore per superamento della posizione massima.  
"0" --> viene impostato solo il flag "ExcEr"  
"1" --> viene impostato il flag e spento il regolatore
- Bit 10: É a "1" quando il generatore ha completato il profilo di velocità calcolato. Viene ripristinato al prossimo comando di start ("FStart").
- Bit 11: Indica la modalità operativa selezionata dalla funzione "FSelOM".  
"0" --> controllo posizione  
"1" --> controllo velocità  
Il bit viene impostato e ripristinato solo a seguito di un successivo comando di start.

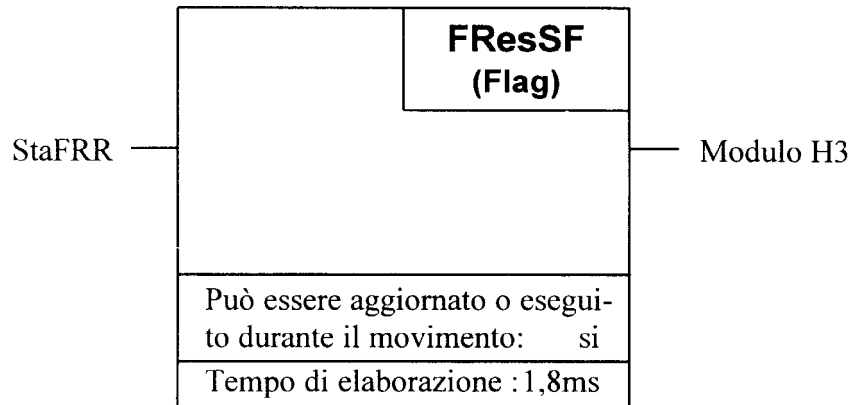
- Bit 12: Indica la direzione in modalità controllo velocità.  
"0" --> avanti  
"1" --> indietro  
Il bit viene impostato e ripristinato solo a seguito di un successivo comando di start.
- Bit 13: Non usato
- Bit 14: Indica che è stata caricata una nuova velocità con le funzioni "FLdAA/R".  
Il bit viene ripristinato solo a seguito di un successivo comando di start.
- Bit 15 bis 31: Non usati

# FResSF

Funzione : -Reset Status Flag  
(Azzera Flag di Stato)

# FResSF

Package software :PCD9.H3E1



### Descrizione funzionale:

Questa funzione è utilizzata per ripristinare i flag di stato di un asse. I flag possono essere ripristinati individualmente o tutti insieme.

### Descrizione degli ingressi e delle uscite:

Simbolo	Descrizione/Funzione	Para- metri	Dati		
			Tipo	Formato	Valore
StaFRR	<b>Status Flag Reset Register</b> (Registro Reset Flag di seguenteS tato)	no	R	Binario	vedere pagina seguente

Questa funzione legge dal registro "StaFRR", quali flag di stato devono essere azzerati. Il flag viene azzerato se il corrispondente bit del registro è a zero.



**Significato del registro "StaFRR":****Significato dei flag di stato:**

"OnDest"      **On Destination** (posizione di destinazione raggiunta)

Portato a "1" dai seguenti eventi:

- Il generatore ha terminato il profilo di velocità calcolato. Può accadere che a causa di errori nella definizione dei parametri oppure a causa di problemi meccanici, il flag venga impostato anche se la posizione di destinazione non sia stata completamente raggiunta pur avendo terminato il profilo definito.
- il regolatore è spento (ad esempio dopo la funzione "FMotOff")
- dopo un arresto manuale (funzione "FStop")  
Un comando di start (funzione "FStart") ripristina automaticamente il flag

"IPuls"        **Index Puls** noted (rilevamento impulso indice)

É a "1" quando è stato rilevato un impulso indice e la posizione attuale è stata scritta nel registro di posizione del modulo H3 (vedere anche la funzione "FSetIP").

"WrapOc"      **Wrap** around **O**ccured (superamento registro posizione)

Portato a "1" quando il registro di posizione supera il valore massimo.  
Questa situazione può verificarsi solo in funzionamento a controllo di velocità

"ExcPEr"      **Excessive Position Error** (errore di posizione eccessivo)

Portato a "1" se l'entità dell'errore di posizionamento, è superiore al valore definito dalla funzione "FSetPE".

"BrkPos"      **Break Position** (posizione di break)

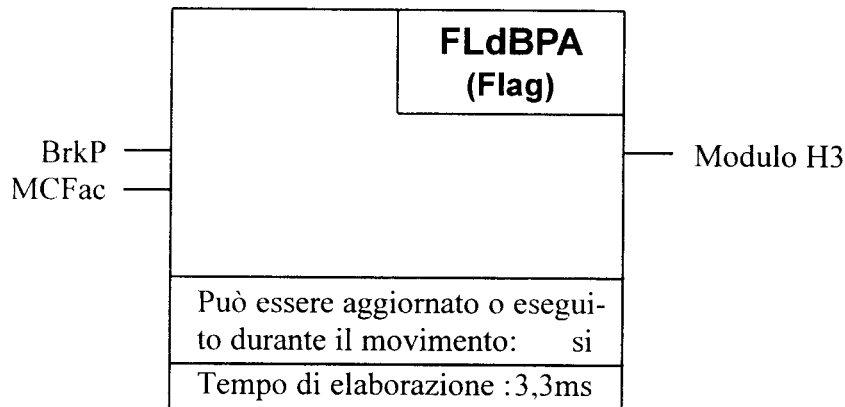
Portato a "1" appena la posizione effettiva viene a trovarsi oltre la posizione di breakpoint caricata con la funzione "FLdBPA/R"

# FLdBPA

Funzione : -Load Break Position Absolute  
(Carica Posizione Assoluta di Break)

# FLdBPA

Package software :PCD9.H3E1



### Descrizione funzionale:

Questa funzione è utilizzata per caricare una posizione assoluta di breakpoint nel modulo H3. Caricamento assoluto significa che il valore è riferito allo zero. Il modulo H3 assume immediatamente la nuova posizione nel registro di lavoro. Se viene raggiunta la posizione di breakpoint, si imposta il flag di stato "BrkPos". Quest'ultimo viene ripristinato con la funzione "FResSF".

Questa funzione consente di ricevere un messaggio ad una posizione particolare, in modo tale che, ad esempio si possano modificare la velocità o i parametri del regolatore.

### Descrizione degli ingressi e delle uscite:

Simbolo	Descrizione/Funzione	Parametri	Dati		
			Tipo	Formato	Valore
BrkP	<b>Break Position</b> (Posizione di Break)  Valore: $[(-2^{30} ..+(2^{30}-1)]/k*10^{-3}$ Unità: definita da k	no	R	Intero	--
MCFac	<b>Motion Control Factor</b> (Fattore di Controllo Movimento)	no	R	virgola mobile	$0..9,223371*10^{18}$

**Fattore di controllo k nel registro "MCFac":**

Il fattore k ha lo stesso significato descritto per la funzione "FLdDA". Si fa notare che questo fattore viene letto dallo stesso registro sia per quanto riguarda la posizione di destinazione che per quanto riguarda la velocità e l'accelerazione.

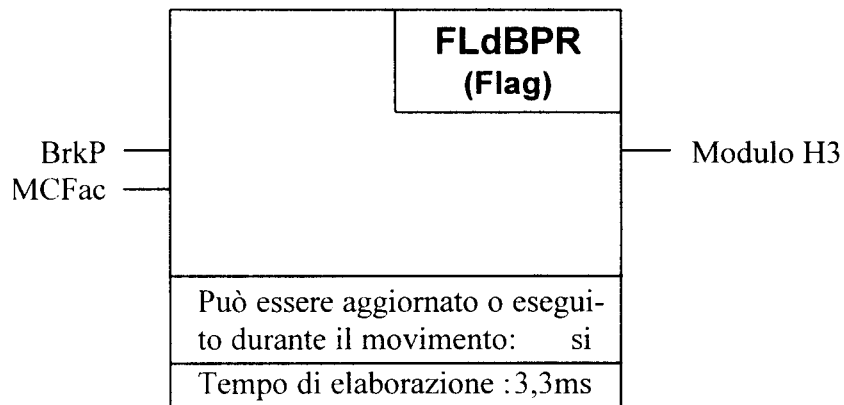
É quindi consigliabile adottare la stessa unità di misura per l'introduzione di tali parametri.

# FLdBPR

Funzione : -Load Break Position Relative  
(Carica Posizione Relativa di Break)

# FLdBPR

Package software :PCD9.H3E1



### Descrizione funzionale:

Questa funzione è utilizzata per caricare una posizione relativa di breakpoint nel modulo H3. Caricamento relativo significa che il valore è riferito alla posizione di destinazione corrente. Il modulo H3 assume immediatamente la nuova posizione nel registro di lavoro. Se viene raggiunta la posizione di breakpoint, si imposta il flag di stato "BrkPos". Quest'ultimo viene ripristinato con la funzione "FResSF".

Questa funzione consente di ricevere un messaggio ad una posizione particolare, in modo tale che, ad esempio si possano modificare la velocità o i parametri del regolatore.

### Descrizione degli ingressi e delle uscite:

Simbolo	Descrizione/Funzione	Para- metri	Dati		
			Tipo	Formato	Valore
BrkP	<b>Break Position</b> (Posizione di Break)  Valore: $[(-2^{30} ..+(2^{30}-1)]/k*10^{-3}$ Unità: definita da k	no	R	Intero	--
MCFac	<b>Motion Control Factor</b> (Fattore di Controllo Movimento)	no	R	virgola mobile	$0..9,223371*10^{18}$

**Fattore di controllo k nel registro "MCFac":**

Il fattore k ha lo stesso significato descritto per la funzione "FLdDA". Si fa notare che questo fattore viene letto dallo stesso registro sia per quanto riguarda la posizione di destinazione che per quanto riguarda la velocità e l'accelerazione.

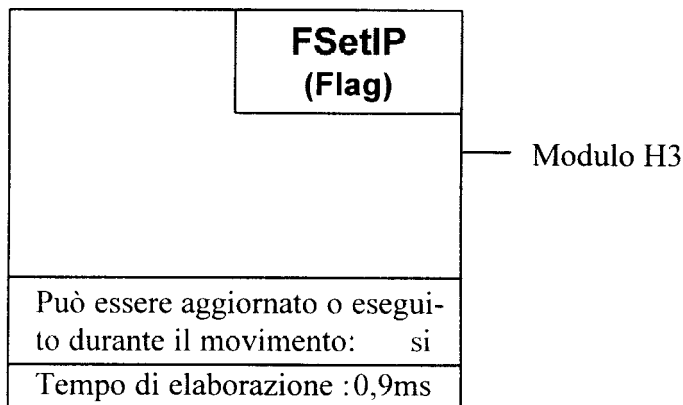
É quindi consigliabile adottare la stessa unità di misura per l'introduzione di tali parametri.

# FSetIP

Funzione : - **Set Index Position**  
(Imposta Posizione Indice)

# FSetIP

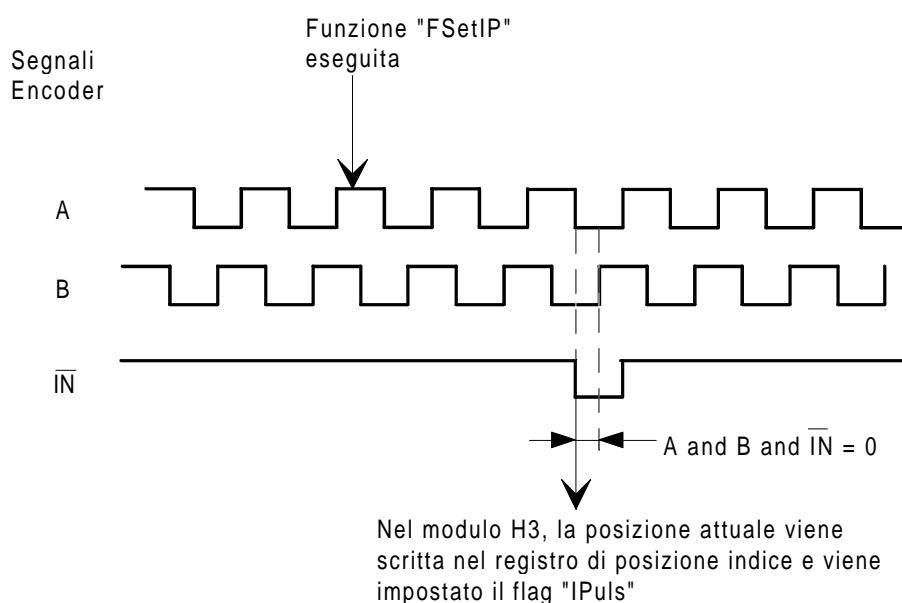
Package software :PCD9.H3E1



### Descrizione funzionale:

Dopo che è stata eseguita questa funzione, alla prima circostanza in cui i segnali dell'encoder A, B e index sono tutti contemporaneamente a zero, la posizione attuale viene scritta nel registro di posizione indice del modulo H3. Al rilevamento dell'impulso di indice, viene impostato il flag "IPuls". La posizione indice può essere letta dal modulo H3 usando la funzione "FRdIP".

Il diagramma riportato di seguito illustra cosa accade nel modulo H3 per quanto riguarda i segnali dell'encoder, dopo che è stata eseguita questa funzione.

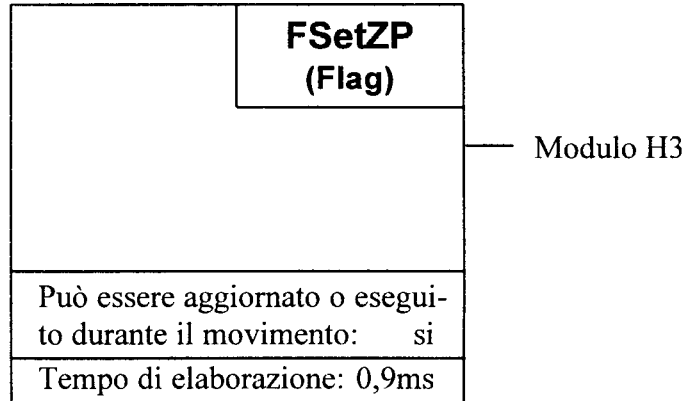


**FSetZP**

Funzione : **-Set Zero Position**  
(Imposta Posizione Zero)

**FSetZP**

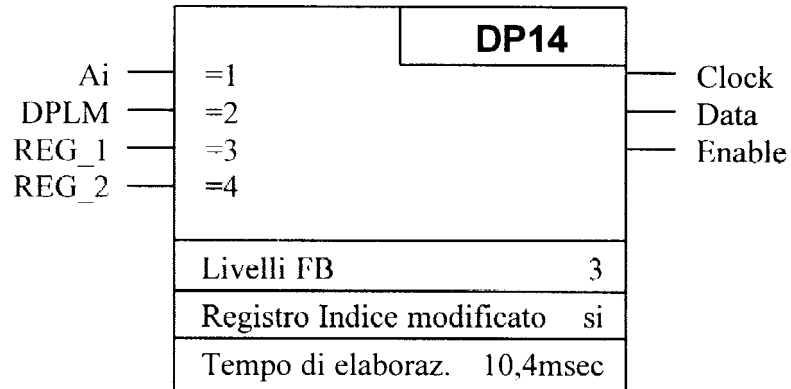
Package software :PCD9.H3E1

**Descrizione funzionale:**

Questa funzione definisce la posizione attuale come posizione zero. Se viene eseguita durante un movimento, la destinazione corrente non viene modificata, fin tanto che non viene eseguito un comando di start "FStart".

**DP14** Funzione : -Display Contents of Register on PCA2.D14 **DP14**  
 (Visualizza Contenuto dei Registri su PCA2.D14)

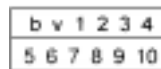
Package software :PCD9.H3E1



**Descrizione funzionale :**

Questo comando consente di visualizzare il valore di un registro di 1\*10 digit oppure di 2\*6 digit sul modulo display PCA2.D14. I possibili formati di visualizzazione sono i seguenti:

**1\*10 digit**  
(1 registro)



b = blank ; v = blank o "-"  
1 .. 10 = digit

Campo di visualizzazione = range di valori del registro ± 2'147'483'647

**2\*6 digit**  
(2 registri)



w = contenuto del registro  
 $0 \leq w \leq + 999'999$  : v = MSD  
 $w > + 999'999$  : v = "A"  
 $0 > w \geq - 99'999$  : v = "-"  
 $w < - 99'999$  : v = "U"

Campo di visualizzazione: - 99'999 ... + 999'999

Il flag DPLM determina il tipo di visualizzazione:

DPLM = "0" --> 1\*10 digit  
 DPLM = "1" --> 2\* 6 digit

Parametri: REG\_1 : Registro per il primo valore da visualizzare  
 REG\_2 : Registro per il secondo valore da visualizzare



**Descrizione degli ingressi e delle uscite :**

Simbolo	Descrizione/Funzione	Para- metri	Dati		
			Tipo	Formato	Valore
Ai	Asse numero i		K	Intero	1... 32
DPLM	Modo di visualizzazione		F	Binario	0, 1
REG_1	primo valore		R	Intero	0.. $10^9-1$
REG_2	secondo valore		R	Intero	0.. $10^6-1$
Clock	Uscita su D14		O	Binario	0, 1
Data	" "		O	Binario	0, 1
Enable	" "		O	Binario	0, 1

**Esempio di programma** per visualizzare la posizione effettiva e la velocità di setpoint dell'asse 1 utilizzando un modulo display PCA2.D14:

```

|
ACC      H
SET      F      DplM+FA1 ; Display Mode = 2*6 Digits
CFB      DP14 ; Display on PCA2.D14
          1      ; Axis 1
          F      DplM+FA1 ; Display Mode
          R      RActP+RA1 ; Actual Position
          R      RSetV+RA1 ; Setpoint Position
|

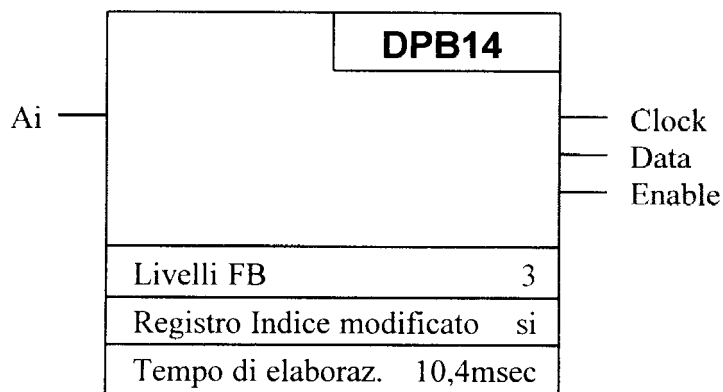
```

## DPB14

Funzione : -Clear Display on PCA2.D14  
(Cancella il display del PCA2.D14)

## DPB14

Package software :PCD9.H3E1



### Descrizione funzionale:

Il comando "DPB14" cancella tutti i caratteri visualizzati sul modulo PCA2.D14 trasformandoli in spazi.

### Descrizione degli ingressi e delle uscite:

Simbolo	Descrizione/Funzione	Para- metri	Dati		
			Tipo	Formato	Valore
Ai	Asse numero i		K	Intero	1... 32
Clock	Uscita su D14		O	Binario	0, 1
Data	" "		O	Binario	0, 1
Enable	" "		O	Binario	0, 1

## 8. Riconoscimento e Gestione errori

---

Il controllore H3 fornisce un messaggio di errore impostando il flag "errore di comando". Tale flag viene impostato quando il controllore non riesce ad interpretare un comando o i dati ricevuti.

Di solito la causa di un "errore di comando" è da ricercarsi in un errore nel programma utente come ad esempio il tentativo di caricare, nel modulo H3, un valore che non rientra nei limiti consentiti.

Esempi di errori:

- Con la funzione "FLdVA" (carica velocità assoluta) viene caricata una velocità negativa.
- Il fattore k di controllo movimento non è stato caricato nel registro "MCFac" nel formato "virgola mobile" --> Il risultato di questo errore può essere il superamento del valore consentito durante il calcolo dell'unità di misura di un parametro.
- Deve essere caricato un valore di controllo  $> 32767$  nel modulo H3.
- Si è tentato di caricare come primo movimento un parametro di movimento relativo (FLdDR, FLdVR, FLdAR e FLdBPR) (prima di aver eseguito il primo comando di start)
- Malfunzionamento del bus PCD4.

### Gestione degli errori di comando

Se si verifica un errore, il controllore H3 ignora il comando che è appena stato inviato e imposta il flag "errore di comando". Questo flag è tenuto sotto controllo da tutti i blocchi funzione che comunicano con il modulo H3. L'utente non può accedere direttamente al flag di errore. In caso di errore il comando precedente viene ripetuto per un massimo di due volte. Dopo tre errori consecutivi viene chiamato il blocco funzione di gestione errori "ComErS". Dopo l'elaborazione di questo blocco funzione, l'esecuzione prosegue con la riga di programma successiva. Il blocco funzione "ComErS" è incluso nel file H3FB.SRC. L'utente può determinare le azioni da intraprendere in caso di errore, inserendo il proprio programma nel blocco funzione.

In ogni caso occorre tenere presente che se viene chiamato il blocco funzione, significa che il comando non è stato eseguito e che quindi non può essere garantito il corretto funzionamento del programma H3. Si consiglia quindi di richiamare XOB 13 forzando un errore di "divisione per zero".

Esempio:

```
FB      ComErS      ; Command Error Stop FB
                          ; (FB stop per errore di comando)

DIV     R    0      ; Forces a divide-by-zero (Forza errore di
      K    0      ; divisione per 0 che chiama XOB 13)
      R    0      ;
      R    0      ;
```

EFB

Se in XOB 13 è presente l'istruzione "DIAG", si può stabilire molto rapidamente dove è avvenuto l'errore nel programma utente.

Esempio:

```
XOB     13

DIAG    R    1      ; Load Diagnostic Registers
                          ; (Carica i Registri Diagnostici)
```

Possibilmente spegnere l'azionamento ed il motore
---

HALT

EXOB

#### **Procedura per localizzare un errore**

Stabilire quale comando ha generato l'errore utilizzando il debugger e il contenuto del Registro di Diagnostica.

Fare riferimento alla descrizione dell'istruzione "DIAG" del PCD.

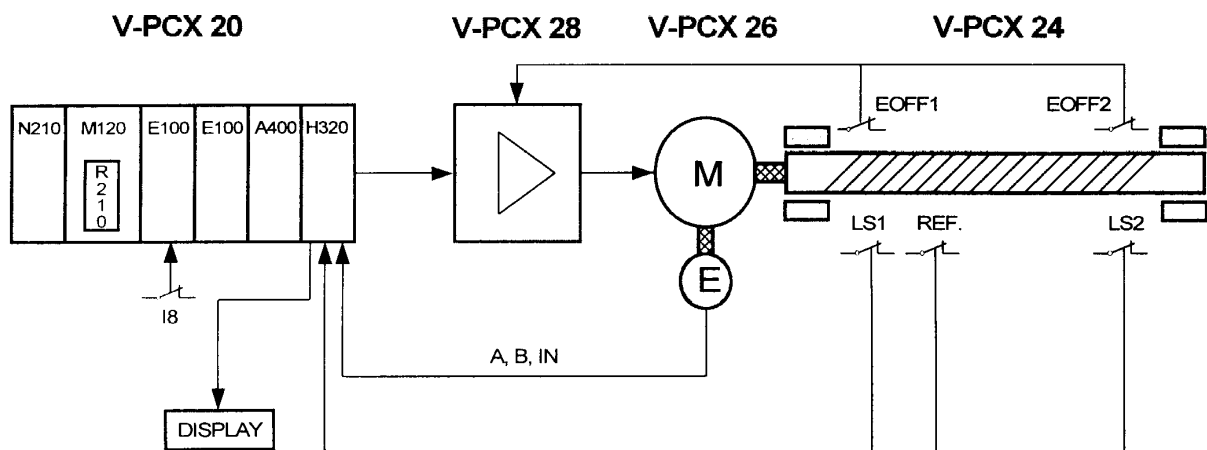
## 9. Esempi

### 9.1 Esempio 1

L'esempio riguarda una applicazione molto semplice ad un solo asse. L'obiettivo è quello di illustrare i passi da compiere e la sequenza con cui questi passi devono essere eseguiti per compiere un semplice movimento.

Hardware:

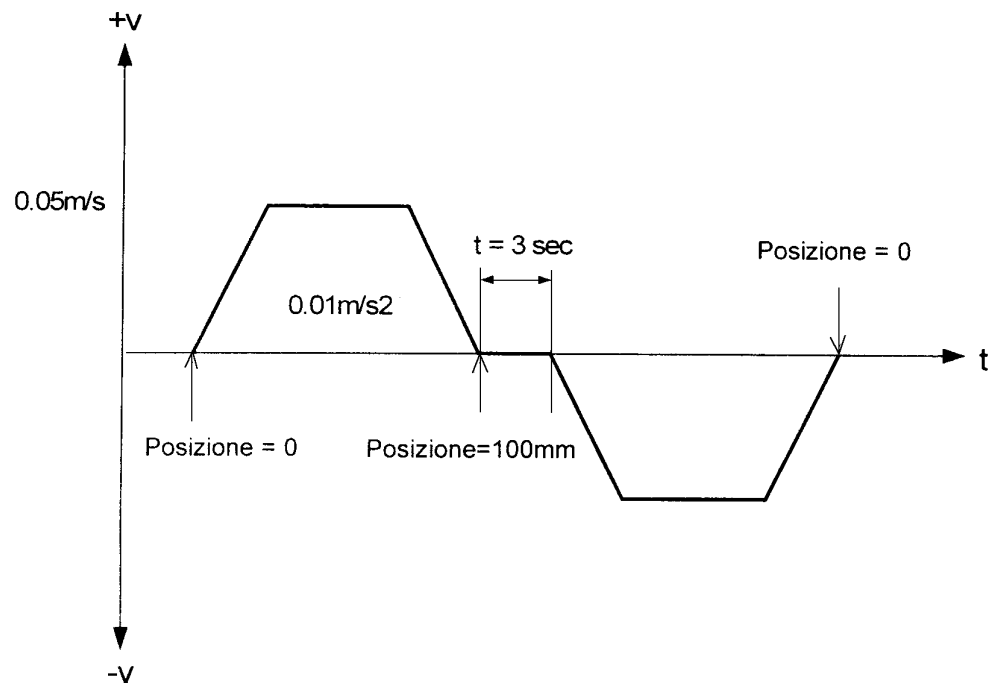
L'esempio si basa sull'hardware dimostrativo V-PCX 20, V-PCX 24, V-PCX 26 e V-PCX 28.



Dati dell'asse:

- Encoder 500 imp./giro
- Vite con passo di 2mm
- Destinazione introdotta con risoluzione 1/10mm
- Velocità massima 0.1m/s
- Accelerazione massima consentita 0.05m/s<sup>2</sup>

Il movimento dovrà avere il seguente profilo:



Per far partire il movimento si deve utilizzare il pulsante collegato all'ingresso 8.

Si può accettare che la posizione zero sia già stata definita dopo l'accensione del controllore e che quindi non sia necessario fornire alcun riferimento. Questo esempio si può utilizzare anche senza il controllo dei fine corsa e senza il modulo display PCA2.D14.

**Soluzione**

Per la soluzione di questo esempio si possono prendere riferimento l'hardware indicato e il package software PCD9.H3E1. Si suppone che l'hardware sia già installato e funzionante. Nella programmazione occorre prendere in considerazione i passi di seguito riportati:

**a) Installazione software**

I due file di H3 H3DEF.SRC e H3FB.SRC vengono copiati nella directory di lavoro. Per prima cosa occorre configurare l'installazione di H3 nel file H3DEF.SRC nel modo seguente:

FMAH3	EQU	48	; Indirizzo di base del modulo H3
IMode	EQU	6	; inizializza la porta di uscita (analogica)
MNA	EQU	1	; imposta 1 asse
BAF	EQU	200	; indirizzo di base dei flag
BAR	EQU	200	; indirizzo di base dei registri
BAC	EQU	40	; indirizzo di base dei contatori
BAFB	EQU	900	; indirizzo di base dei blocchi
funzione			
RA1	EQU	0*NoRfeA	; costante blocco di registri asse 1
FA1	EQU	0*NoFfeA	; costante blocco di flag asse 1

Dal momento che non viene utilizzata l'allocazione di simboli esterni occorrerà definire:

```
PUBLSYM      EQU      0
```

Tutti gli altri simboli devono rimanere inalterati

Nel file H3FB.SRC deve essere definito il simbolo

```
EXTNSYM      EQU      0
```

A seguito di questa dichiarazione, il file di definizione simboli H3DEF.SRC viene incluso automaticamente dall'istruzione \$INCLUDE durante l'assemblaggio. Includendo il file di definizione e il file H3FB.SRC nel file utente si ottiene il vantaggio di avere a disposizione gli indirizzi assoluti, immediatamente dopo l'assemblaggio. Questo requisito è indispensabile se si vuole avere la possibilità di visualizzare un valore durante la verifica del programma con il debugger.

Per contro, se si utilizza l'allocazione esterna dei simboli, gli indirizzi assoluti non sarebbero disponibili se non dopo la generazione del file di documentazione (.DOC).

Alla fine del file H3FB.SRC viene posto il Blocco Funzione "ComErS" che verrà chiamato nel caso di errore di comando ripetuto. In questo blocco funzione, l'utente può determinare quali misure dovranno essere prese in caso di errore. Si consiglia di chiamare XOB 13 forzando una operazione di divisione per zero. All'interno di XOB 13 si può programmare una istruzione "DIAG". In questo modo si può stabilire velocemente dove, all'interno del programma utente è avvenuto un errore. Per ulteriori dettagli fare riferimento al Capitolo 8.

```

FB      ComErS          ; Stop per Errore di Comando
DIV     R    0
        K    0          ; Forza una divisione per zero
        R    0          ; in modo da richiamare XOB 13
        R    0
EFB

```

Ad eccezione di queste due modifiche, il file H3FB.SRC non deve essere alterato.

Nel passo successivo viene assemblato il file per determinare se le sopra citate modifiche sono state eseguite correttamente. Se l'operazione di assemblaggio viene eseguita con successo, il file non dovrà essere riassembleto, ma dovrà semplicemente essere linkato al programma utente.

## b) Programma Utente

Per creare il programma, si può fare riferimento alla struttura di programma descritto nel Capitolo 7.2:

### 1. Inizializzazione in XOB 16

Come primo passo, devono essere determinati i valori dei seguenti registri di inizializzazione: (i registri di inizializzazione vengono letti dal Blocco Funzione "AxInit")

#### - Parola di Controllo Movimento "MCW"

In questo registro vengono definiti il tipo di stop, la modalità operativa e le misure da prendere in caso di errore di posizione.

Tipo di stop:            Vedere anche la funzione "FStop"  
                           In questo primo esempio, il tipo di stop non è importante dal momento che non sono ammessi comandi di stop normali nel programma.  
                           In ogni caso, si definisce il bit 10 = "1" -> stop con decelerazione definita

Modi di Funzionamento: Vedere anche la funzione "FSelOM"  
                               Si opera in modalità di controllo posizione -  
                               > bits 11 e 12 = "0"



Errore di posizione: Vedere anche la funzione "FSetPE"  
 In caso di errore di posizione, deve essere impostato solo il  
 flag di stato "ExcPEr".  
 -> valore dei bit da 0 a 7 = 1BH (esadecimale)

Registro "MCW"

Bit      31      13 12 11 10 9 8 7      0

non usato	0	0	1	0	0	1BH
-----------	---	---	---	---	---	-----

-> Caricamento del registro "MCW" con il valore 41BH (esadecimale)

**- Errore di Posizione "PosEr"**    --> vedere la funzione "FSetPE"

In questo primo esempio, l'informazione relativa all'errore di posizione non è significativa in quanto il flag di stato "ExcPEr" non viene controllato dal programma utente (ad esempio per spegnere il drive). Dal momento però che l'errore di posizione deve essere comunque caricato nel modulo H3 durante l'inizializzazione, e deve essere nel range di valori permessi, si definisce come differenza massima tra una posizione impostata e una posizione effettiva un intero giro; in questa situazione verrebbe impostato il flag "ExcPEr".

--> Caricamento del registro "PosEr" con il valore 2000 = 4\*500 imp./giro

**- Fattori PID**    --> vedere la funzione "FLdRP"

Per semplicità, la procedura che determina i fattori PID viene illustrata ricorrendo ad un altro esempio. Si suppone che i fattori siano già stati forniti.

I registri devono essere caricati con i seguenti valori:

Fattore di proporzione "KProp":                      150

Fattore di Integrazione "KInt":                      50

Fattore di Derivazione "KDer":                      50

Limite di Integrazione "IntL":                      500

Intervallo di campionamento del termine derivativo "SampI":    15  
 (5.46ms=16\*0.341ms)

**- Fattore k di controllo movimento "MCFac":** --> vedere la funzione "FLdDA"

Il fattore di controllo movimento viene utilizzato per determinare le unità di misura dei valori di ingresso e uscita per quanto riguarda la posizione, la velocità e l'accelerazione. Si richiede una risoluzione di 1/10mm come valore della posizione di destinazione.

$$\frac{4 * In}{s} = \frac{4 * 500 \text{ imp./giro.}}{20 * 1/10\text{mm/giro}} = 100 \text{ imp. per } 1/10\text{mm}$$

—> Caricamento del registro "MCFac" con il valore 100 (formato virgola mobile).

**- Velocità "Veloc"** —> vedere la funzione "FLdVA"

Siccome per molte applicazioni si usa spesso una sola velocità, in fase di inizializzazione viene caricata una velocità assoluta.

Nel nostro esempio si stabilisce una velocità di 0,05 m/s.

—> Caricamento del registro "Veloc" con valore 500 (unità 1/10 mm/s)

**- Accelerazione "Accel"** —> vedere la funzione "FLdAA"

L'accelerazione è di 0,01 m/s<sup>2</sup>

—> Caricamento del registro "Accel" con valore 100 (unità 1/10 mm/s<sup>2</sup>)

Questi valori devono essere caricati nei registri all'interno XOB 16 prima di chiamare il Blocco Funzione "AxInit".

## 2. Gestione ciclica dell'asse

In COB 0, per la gestione degli assi viene chiamato il Blocco Funzione "AxHndlg".

## 3. Definizione del programma di controllo movimento

Il programma di controllo movimento è scritto tenendo conto del profilo di velocità, all'interno della struttura GRAFTEC (SB 0).

Le pagine seguenti illustrano il file sorgente (BSP01.SRC) del programma utente relativo a questo esempio.

```

; Demo programm for the motion control module PCD4.H3..
; =====
; Name   : BSP01.SRC
; U. Jäggi 21.08.90

$ include H3DEF.SRC

;***** Cold-Start (Initialisation)
XOB      16
;----- Cold-Start Definitions
;----- loading of the initialisation
registers

Ld      R    MCW+RA1    ; Motion Control Word
          41BH         ; Stop smothly, Position mode
                   ; only statusflag (Pos.error)
Ld      R    PosEr+RA1 ; Position Error
          2000        ; 4 * 500 pulses
Ld      R    KProp+RA1 ;Proportional factor
          150
Ld      R    KInt+RA1  ; Integral factor
          50
Ld      R    KDer+RA1  ; Derivative factor
          50
Ld      R    IntL+RA1  ; Integration Limit
          500
Ld      R    SampI+RA1 ; Sampling Interval
          15          ; 5.46ms
Ld      R    MCFac+RA1 ; Motion Control Factor
          100.0       ; 100 Imp./1/10mm
Ld      R    Veloc+RA1 ; Velocity
          500         ; 0.05m/s
Ld      R    Accel+RA1 ; Acceleration
          100         ; 0.01m/s2

;-----
CFB      AxInit        ; Axis Initialisation
          1            ; X axis
          RA1
          FA1
          IMode        ; Initialisation mode: Analog/PWM
;----- End XOB 16
EXOB

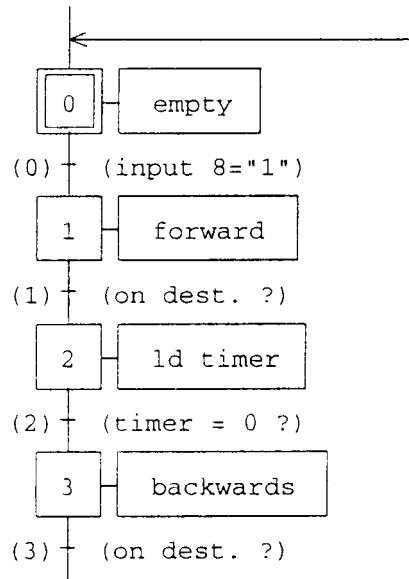
;***** Cyclic program
COB      0
          0
;-----
CFB      AxHndlg       ; Axis Handling
          1            ; X axis
          RA1
          FA1
;-----
CSB      0             ; Call of the motion control program
;----- ; End cyclic program

ECOB

```

\*\*\* SAIA PCD GRAFTEC EDITOR \$113 \*\*\*  
FILE: BSP01.GLS (29.08.90 10.51)  
\*\*\* SAIA AG - CH-3280 MURTEN \*\*\*  
SB-NUMBER: 0  
PAGE-NB: 0

PAGE: 1  
PRODUCED: 29.08.90 10.51



```

;***** Motion control program
SB      0
;-----
IST     0      ; empty
        I      3      ; on dest. ?
        O      0      ; input 8="1"
EST
;-----
ST      1      ; forward
        I      0      ; input 8="1"
        O      1      ; on dest. ?
ld    r      DestP+RA1 ;Destination Position
        1000      ;100mm = 1000*1/10mm
set   f      FLdDA+FA1 ;Load Destination Absolute
set   f      FStart+FA1 ;Start motion
EST
;-----
ST      2      ; ld timer
        I      1      ; on dest. ?
        O      2      ; timer = 0 ?
ld    t      0      ; timer 0
        30      ; 3s
EST
;-----
ST      3      ; backwards
        I      2      ; timer = 0 ?
        O      3      ; on dest. ?
ld    r      DestP+RA1 ;Destination Position
        0      ; Position 0
set   f      FLdDA+FA1 ;Load Destination Absolute
set   f      FStart+FA1 ;Start motion
EST
;-----
TR      0      ; input 8="1"
        I      0      ; empty
        O      1      ; forward
sth   i      8      ; motion free ?
ETR
;-----
TR      1      ; on dest. ?
        I      1      ; forward
        O      2      ; ld timer
sth   f      OnDest+FA1 ;On Destination ?
ETR
;-----
TR      2      ; timer = 0 ?
        I      2      ; ld timer
        O      3      ; backwards
stl   t      0      ; timer = 0 ?
ETR
;-----
TR      3      ; on dest. ?
        I      3      ; backwards
        O      0      ; empty
sth   f      OnDest+FA1 ;On Destination ?
ETR
;-----
ESB

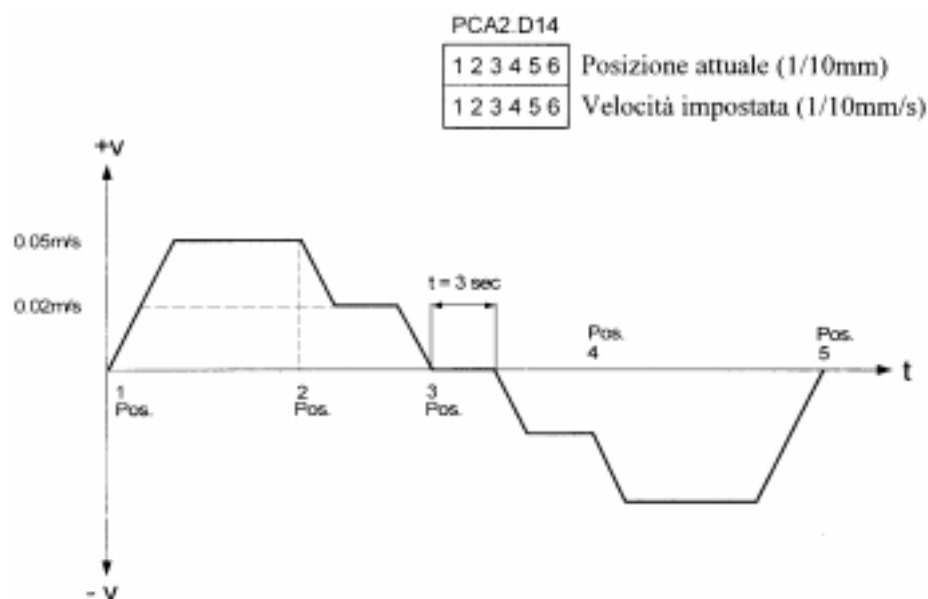
```

## 9.2 Esempio 2

L'esempio è costituito su quanto si è visto nell'esempio 1. Quest'ultimo è stato esteso aggiungendo la modifica alla velocità durante il movimento e la visualizzazione sul modulo PCA2.D14.

Hardware: Identico a quanto già illustrato per l'esempio 1.

Il movimento deve avere il seguente andamento:



Posizione 1:	0mm	(posizione iniziale)
Posizione 2:	70mm	(posizione di breakpoint)
Posizione 3:	100mm	(destinazione)
Posizione 4:	70mm	(posizione di breakpoint)
Posizione 5:	0mm	(destinazione)

- La velocità per il movimento indietro lento e veloce è la stessa adottata per il movimento avanti.
- Accelerazione costante di 0,01 m/s<sup>2</sup>.
- Il movimento inizia premendo il pulsante collegato all'ingresso 8.
- Il display PCA2.D14 dovrà visualizzare la posizione attuale nella riga superiore e la velocità impostata nella riga inferiore.

## **Soluzione**

### **a) Installazione software**

Si suppone di aver già installato l'hardware e il software come per l'esempio 1.

(I file H3DEF.SRC e H3FB.SRC possono essere presi senza modifiche).

### **b) Programma utente**

#### **1. Inizializzazione in XOB 16**

Può essere presa dall'esempio 1 senza modifiche.

#### **2. Gestione ciclica dell'asse**

Rispetto all'esempio 1 vengono utilizzate in aggiunta le funzioni "FRdAP" e "FRdSV" per leggere la posizione attuale e la velocità impostata dal modulo H3 e visualizzarne i valori sul display PCA2.D14 con il blocco funzione "DP14".

#### **3. Definizione del programma di controllo movimento**

Il programma di controllo movimento è scritto in base al profilo di velocità, all'interno di una struttura GRAFTEC (SB 0). Se occorre modificare la velocità durante il movimento ad una particolare posizione, ciò si può ottenere caricando una posizione di breakpoint e interrogando il flag di stato "BrkPos". Dopo aver caricato la posizione di breakpoint, occorre preoccuparsi di ripristinare il flag di stato "BrkPos" con la funzione "FResSF".

Vedere anche la descrizione delle funzioni "FLdBPA" e "FResSF".

Le pagine seguenti riportano il file sorgente (BSP02.SRC) del programma utente utilizzato per questo esempio.

```

; Demo programm for the motion control module PCD4.H3..
; =====
; Name : BSP02.SRC
; U. Jäggi 27.08.90

$ include H3DEF.SRC

;***** Cold-Start (Initialisation)
XOB          16
;----- Cold-Start Definitions
;----- loading of the initialisation
registers

Ld          R      MCW+RA1      ;Motion Control Word
           41BH                ;Stop smothly, Position mode
           ;only statusflag (Pos.error)

Ld          R      PosEr+RA1    ;Position Error
           2000                ;4 * 500 pulses

Ld          R      KProp+RA1    ;Proportional factor
           150

Ld          R      KInt+RA1     ;Integral factor
           50

Ld          R      KDer+RA1     ;Derivative factor
           50

Ld          R      IntL+RA1     ;Integration Limit
           500

Ld          R      SampI+RA1    ;Sampling Interval
           15                  ;5.46ms

Ld          R      MCFac+RA1    ;Motion Control Factor
           100.0               ;100 Imp./1/10mm

Ld          R      Veloc+RA1    ;Velocity
           500                 ;0.05m/s

Ld          R      Accel+RA1    ;Acceleration
           100                 ;0.01m/s2

;-----
CFB          AxInit            ;Axis Initialisation
           1                    ;X axis
           RA1
           FA1
           IMode                ;Initialisation mode: Analog/PWM
;----- End XOB 16
EXOB

```



```

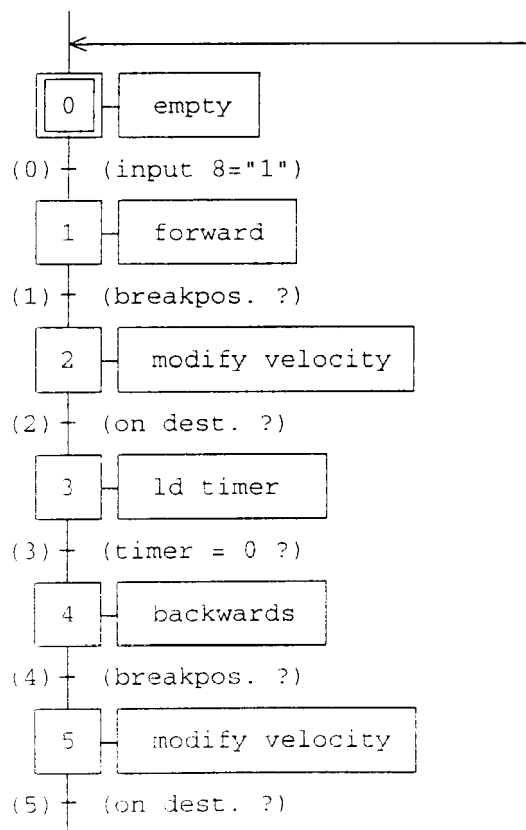
;***** Cyclic program
COB      0
        0
;-----
CFB      AxHndlg      ;Axis Handling
        1              ;X axis
        RA1
        FA1
;-----
SET      F      FRdAP+FA1      ;Read Actual Position
SET      F      FRdSV+FA1      ;Read Setpoint Velocity
;-----
SET      F      DplM+FA1      ;Diplay Mode = 2*6 digits
CFB      DP14          ;Display on PCA2.D14
        1              ;Axis 1
        F      DplM+FA1      ;Display Mode
        R      RActP+RA1     ;Actual Position
        R      RSetV+RA1     ;Setpoint Velocity
;-----
CSB      0              ;Call of the motion control
program
;----- ;End cyclic program
ECOB

```

```

*** SAIA PCD GRAFTEC EDITOR $113 ***
FILE: BSP01.GLS (29.08.90 10.46)
*** SAIA AG - CH-3280 MURTEN ***
PAGE: 1
PRODUCED: 29.08.90 10.52
SB-NUMBER: 0
PAGE-NB: 0

```



```

;***** Motion control program
SB      0
;-----
IST     0      ; empty
        I      5      ; on dest. ?
        O      0      ; input 8="1"

EST
;-----
ST      1      ; forward
        I      0      ; input 8="1"
        O      1      ; breakpos. ?
ld      r      DestP+RA1 ; Destination Position
        1000      ; 100mm = 1000*1/10mm
set     f      FLdDA+FA1 ; Load Destination Absolute
ld      r      BrkP+RA1  ; Break Position
        700      ; 70mm
set     f      FLdBPA+FA1 ; Load Break Position Absolute
ld      r      StaFRR+RA1 ; Status Flag Reset Register
        0      ; reset all Status Flag
set     f      FResSF+FA1 ; Reset Status Flag
set     f      FStart+FA1 ; Start motion
EST
;-----
ST      2      ; modify velocity
        I      1      ; breakpos. ?
        O      2      ; on dest. ?
ld      r      Veloc+RA1 ; Velocity
        200      ; 0.02m/s
set     f      FLdVA+FA1 ; Load Velocity Absolute
set     f      FStart+FA1 ; Update Velocity
EST
;-----
ST      3      ; ld timer
        I      2      ; on dest. ?
        O      3      ; timer = 0 ?
ld      t      0      ; timer 0
        30      ; 3 sec.
EST
;-----
ST      4      ; backwards
        I      3      ; timer = 0 ?
        O      4      ; breakpos. ?
ld      r      DestP+RA1 ; Destination Position
        0      ; Position 0
set     f      FLdDA+FA1 ; Load Destination Absolute
set     f      FResSF+FA1 ; Reset Status Flag
set     f      FStart+FA1 ; Start motion
EST
;-----
ST      5      ; modify velocity
        I      4      ; breakpos. ?
        O      5      ; on dest. ?
ld      r      Veloc+RA1 ; Velocity
        500      ; 0.05m/s
set     f      FLdVA+FA1 ; Load Velocity Absolute
set     f      FStart+FA1 ; Update Velocity
EST

```

```

;-----
TR      0
        I    0      ;input 8="1"
        O    1      ;empty
        O    1      ;forward
sth   i    8      ;motion free ?
ETR
;-----
TR      1
        I    1      ;breakpos. ?
        O    2      ;forward
        O    2      ;modify velocity
sth   f    BrkPos+FA1 ;Break Position reached ?
ETR
;-----
TR      2
        I    2      ;on dest. ?
        O    3      ;modify velocity
        O    3      ;ld timer
sth   f    OnDest+FA1 ;On Destination ?
ETR
;-----
TR      3
        I    3      ;timer = 0 ?
        O    4      ;ld timer
        O    4      ;backwards
stl   t    0      ;timer = 0 ?
ETR
;-----
TR      4
        I    4      ;breakpos. ?
        O    5      ;forward
        O    5      ;modify velocity
sth   f    BrkPos+FA1 ;Break Position reached ?
ETR
;-----
TR      5
        I    5      ;on dest. ?
        O    0      ;modify velocity
        O    0      ;empty
sth   f    OnDest+FA1 ;On Destination ?
ETR
;-----
ESB

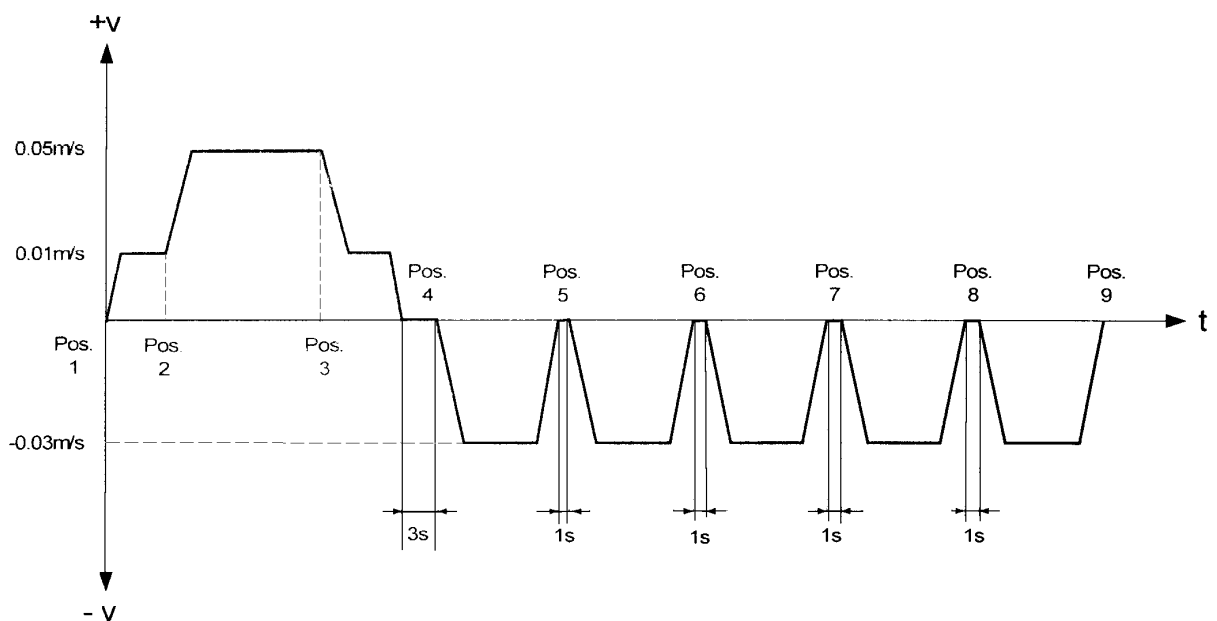
```

### 9.3 Esempio 3

L'esempio è stato costruito sulla base di quanto è stato acquisito con l'esempio 2. Quest'ultimo è stato esteso con un programma più complesso di controllo movimento (manuale e automatico).

Hardware: Identico a quanto già illustrato per l'esempio 1.

Nel controllo automatico, il movimento dovrà avere il seguente andamento:



Posizione 1:	0mm	(posizione iniziale)
Posizione 2:	30mm	(posizione di breakpoint)
Posizione 3:	110mm	(posizione di breakpoint)
Posizione 4:	150mm	(destinazione)
Posizione 5:	120mm	(destinazione)
Posizione 6:	90mm	(destinazione)
Posizione 7:	60mm	(destinazione)
Posizione 8:	30mm	(destinazione)
Posizione 9:	0mm	(destinazione)

**Condizioni:**

- Accelerazione costante di 0,01 m/s<sup>2</sup>.
- In qualsiasi momento deve essere possibile commutare da controllo automatico a controllo manuale.
- Il programma deve partire automaticamente alla pressione di un tasto e deve girare continuamente finchè non si passa al controllo manuale oppure finchè non viene premuto il tasto di stop.
- In qualsiasi momento deve essere possibile fermare l'asse con la pressione di un tasto.

**Controllo manuale:**

- Premendo un tasto deve essere possibile operare in direzione positiva e negativa alla velocità +/- 0,02 m/s. Appena si rilascia il tasto deve essere generato uno stop del tipo definito.
- Deve essere possibile definire la posizione corrente come posizione zero con la pressione di un tasto.

**Definizione degli ingressi:**

Ingresso 0	-->	Controllo automatico/manuale (0/1)
Ingresso 8	-->	Avvio programma automatico
Ingresso 9	-->	Avanti con controllo manuale
Ingresso 10	-->	Indietro con controllo manuale
Ingresso 11	-->	Definisce posizione zero
Ingresso 15	-->	Stop

**Soluzione****a) Installazione software**

Si suppone che l'hardware e il software siano già stati installati come nell'esempio 2.

(I file H3DEF.SRC e H3FB.SRC possono rimanere inalterati).

**b) Programma utente****1. Inizializzazione in XOB 16**

É la stessa già esaminata nell'esempio 2.

## **2. Gestione ciclica degli assi**

Il controllo della modalità operativa si può ottenere anche in questa fase. Per consentire in qualsiasi momento la commutazione tra controllo automatico e controllo manuale, l'attivazione dello switch all'ingresso 0 fa ripartire il programma GRAFTEC dal passo 0. In caso di commutazione tra controllo manuale e controllo automatico, questo significa che il movimento corrente continua fino all'arresto del motore.

Se viene attivato il tasto di stop all'ingresso 15 il motore si ferma e il programma GRAFTEC riparte dal passo 0.

## **3. Definizione del programma di controllo movimento**

Il programma di controllo movimento viene scritto in una struttura GRAFTEC (SB 0).

Programma automatico:

Dal momento che vengono raggiunte ogni volta le posizioni da 5 a 9, e che lo stesso percorso viene eseguito alla stessa velocità questi movimenti possono essere programmati all'interno di un loop (posizione di destinazione caricata come valore relativo).

Le pagine seguenti illustrano il file sorgente (BSP03.SRC) del programma utente relativo a questo esempio.

```

; Demo programm for the motion control module PCD4.H3..
; =====
; Name   : BSP03.SRC
; U. Jäggi 27.08.90

$ include H3DEF.SRC

;***** Cold-Start (Initialisation)
XOB      16
;----- Cold-Start Definitions
;----- loading of the initialisation registers

Ld      R    MCW+RA1    ;Motion Control Word
          41BH        ;Stop smoothly, Position mode
                   ;only statusflag (Pos.error)
Ld      R    PosEr+RA1 ;Position Error
          2000        ;4 * 500 pulses
Ld      R    KProp+RA1 ;Proportional factor
          150
Ld      R    KInt+RA1  ;Integral factor
          50
Ld      R    KDer+RA1  ;Derivative factor
          50
Ld      R    IntL+RA1  ;Integration Limit
          500
Ld      R    SampI+RA1 ;Sampling Interval
          15          ;5.46ms
Ld      R    MCFac+RA1 ;Motion Control Factor
          100.0       ;100 Imp./1/10mm
Ld      R    Veloc+RA1 ;Velocity
          500         ;0.05m/s
Ld      R    Accel+RA1 ;Acceleration
          100         ;0.01m/s2

;-----
CFB      AxInit      ;Axis Initialisation
          1          ;X axis
          RA1
          FA1
          IMode      ;Initialisation mode: Analog/PWM

;----- End XOB 16
EXOB

```

```

;***** Cyclic program
COB      0
         0

;-----
CFB      AxHndlg ;Axis Handling
         1       ;X axis
         RA1
         FA1

;-----
SET      F      FRdAP+FA1 ;Read Actual Position
SET      F      FRdSV+FA1 ;Read Setpoint Velocity
;-----
SET      F      DplM+FA1  ;Diplay Mode = 2*6 digits
CFB      DP14       ;Display on PCA2.D14
         1         ;Axis 1
         F      DplM+FA1  ;Display Mode
         R      RActP+RA1 ;Actual Position
         R      RSetV+RA1 ;Setpoint Velocity
;-----

CSB      0         ;Call of the motion control program

;-----
STH      I      0         ;Manual operation mode
DYN      F      0
RSB      H      0         ;Restart SB
         0         ;at Step 0
STL      I      0         ;Automatic operation mode
DYN      F      1
RSB      H      0         ;Restart SB
         0         ;at Step 0
STH      I      15        ;Stop switch
DYN      F      15
JR       L      END
SET      F      FStop+FA1 ;Stop X axis
RSB      0       ;Restart SB
         0       ;at Step 0

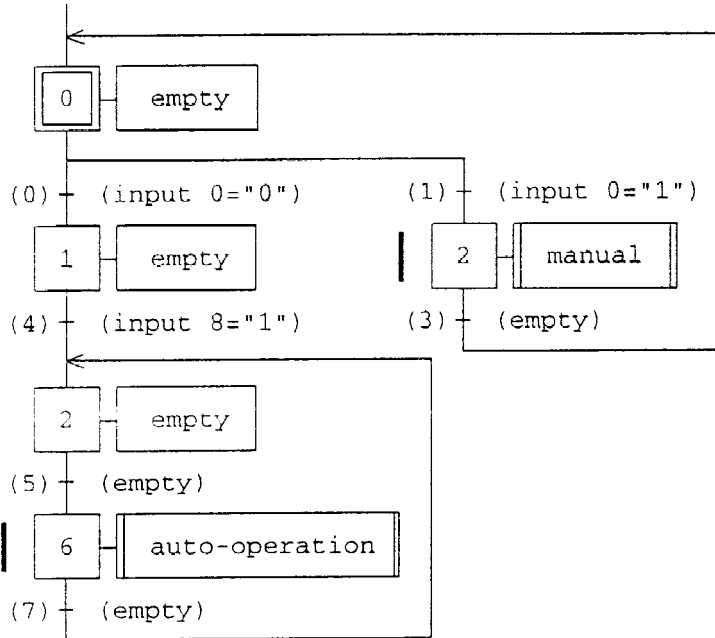
;----- ; End cyclic program
END:     ECOB

```

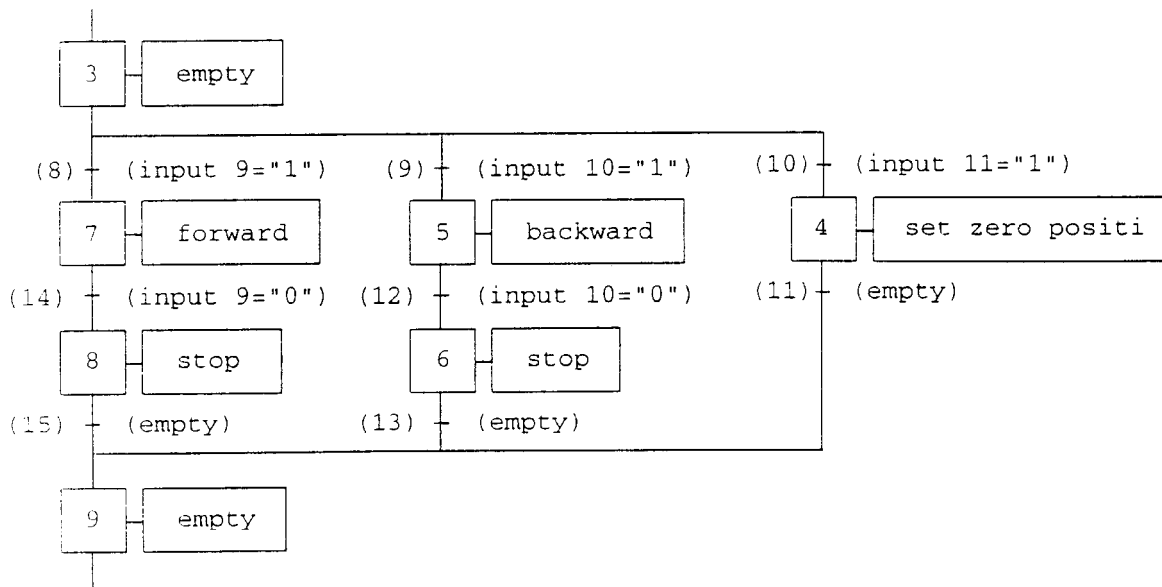


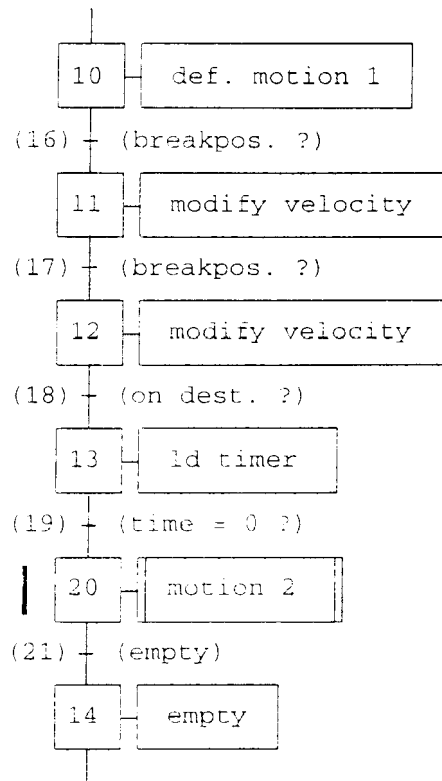
\*\*\* SAIA PCD GRAFTEC EDITOR \$113 \*\*\*  
 FILE: BSP03.GLS (29.08.90 11.08)  
 \*\*\* SAIA AG - CH-3280 MURTEN \*\*\*  
 SB-NUMBER: 0  
 PAGE-NB: 0

PAGE: 1  
 PRODUCED: 29.08.90 11.11

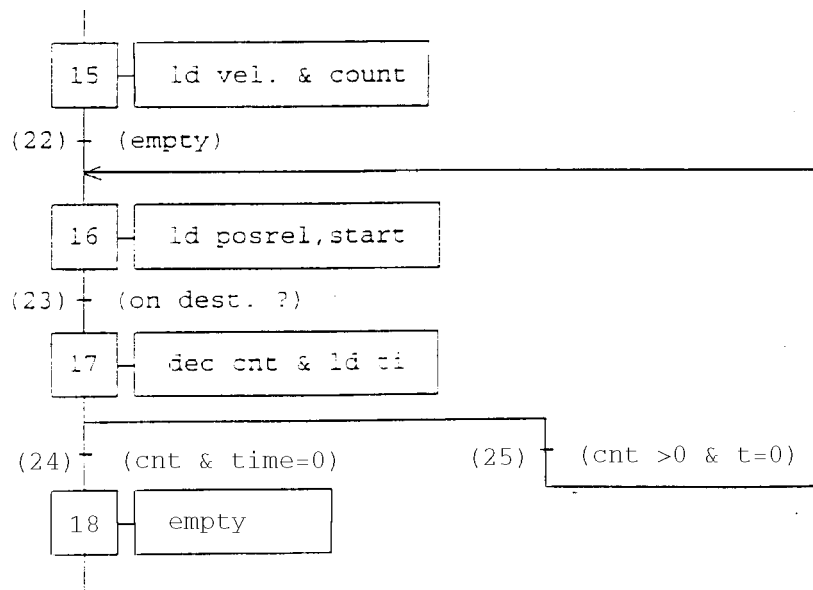


PAGE-NB: 2 manual





PAGE-NB: 20 motion 2



```

;***** Motion control program
SB      0
;-----
IST     0          ; empty
        I      3    ; empty
        O      0    ; input 0="0"
EST
;-----
ST      1          ; empty
        I      0    ; input 0="0"
        O      4    ; input 8="1"
EST
;-----
ST      2          ; empty
        I      4    ; input 8="1"
        I      7    ; empty
        O      5    ; empty
EST
;-----
ST      3          ; empty
        I      1    ; input 0="1"
        O      8    ; input 9="1"
        O      9    ; input 10="1"
        O     10    ; input 11="1"
EST
;-----
ST      4          ; set zero position
        I     10    ; input 11="1"
        O     11    ; empty
set    f    FSetZP+FA1 ; set zero position X axis
EST
;-----
ST      5          ; backward
        I      9    ; input 10="1"
        O     12    ; input 10="0"

ld      r    Veloc+RA1 ; velocity
                200 ; 0.02m/s
set    f    FLdVA+FA1 ; load velocity
set    f    FBackW+FA1 ; backward X axis
EST
;-----
ST      6          ; stop
        I     12    ; input 10="0"
        O     13    ; empty
set    f    FStop+FA1 ; stop X axis
EST
;-----
ST      7          ; forward
        I      8    ; input 9="1"
        O     14    ; input 9="0"

ld      r    Veloc+RA1 ; velocity
                200 ; 0.02m/s
set    f    FLdVA+FA1 ; load velocity
set    f    FForw+FA1 ; forward X axis
EST

```

```

;-----
ST      8          ; stop
        I      14    ; input 9="0"
        O      15    ; empty
set    f      FStop+FA1 ; stop X axis
EST
;-----
ST      9          ; empty
        I      15    ; empty
        I      13    ; empty
        I      11    ; empty
        O      3     ; empty
EST
;-----
ST      10         ; def. motion 1
        I      5     ; empty
        O      16    ; breakpos. ?
ld     r      Veloc+RA1 ; Velocity
        100        ; 0.01m/s
set    f      FLdVA+FA1 ; Load Velocity Absolute
ld     r      DestP+RA1 ; Destination Position
        1500       ; 150mm
set    f      FLdDA+FA1 ; Load Destination Absolute
ld     r      BrkP+RA1  ; Break Position
        300        ; 30mm
set    f      FLdBPA+FA1 ; Load Break Position Absolute
ld     r      StaFRR+RA1 ; Status Flag Reset Register
        0          ; reset all Status Flag
set    f      FResSF+FA1 ; Reset Status Flag
set    f      FStart+FA1 ; Start motion
EST
;-----
ST      11         ; modify velocity
        I      16    ; breakpos. ?
        O      17    ; breakpos. ?
ld     r      Veloc+RA1 ; Velocity
        500        ; 0.05m/s
set    f      FLdVA+FA1 ; Load Velocity Absolute
ld     r      BrkP+RA1  ; Break Position
        1100       ; 110mm
set    f      FLdBPA+FA1 ; Load Break Position Absolute
set    f      FResSF+FA1 ; Reset Status Flag
set    f      FStart+FA1 ; Start motion
EST
;-----
ST      12         ; modify velocity
        I      17    ; breakpos. ?
        O      18    ; on dest. ?
ld     r      Veloc+RA1 ; Velocity
        100        ; 0.01m/s
set    f      FLdVA+FA1 ; Load Velocity Absolute
set    f      FStart+FA1 ; Start motion
EST

```

```

;-----
ST      13                ;ld time
                        I  18      :      on dest. ?

      O  19                ;time = 0 ?
ld    t  0
      O  20
EST
;-----
ST      14                ;empty
                        I  21                ;empty
                        O  7                ;empty
EST
;-----
ST      15                ;ld vel. & counter
                        I  19                :time = 0 ?
                        O  22                ;empty
ld    r  Veloc+RA1      ;Velocity
                        O  300              ;0.03m/s
set   f  FLdVA+FA1     ;Load Velocity Absolute
ld    c  100
                        O  5
EST
;-----
ST      16                ;ld posrel,start
                        I  22                ;empty
                        I  25                :cnt>0 & t=0
                        O  23                ;on dest. ?
ld    r  DestP+RA1     ;Destination Position
                        O  -300             ;-30mm
set   f  FLdDR+FA1     ;Load Destination Relative
set   f  FStart+FA1    ;Start motion
EST
;-----
ST      17                ;dec cnt & ld timer
                        I  23                :on dest. ?
                        O  24                ;cnt & time=0
                        O  25                ;cnt>0 & t=0
dec   c  100
ld    t  0
                        O  10
EST
;-----
ST      18                ;empty
                        I  24                ;cnt & time=0
                        O  21                ;empty
EST
;-----

```

```

;-----
TR      0          ;input 0="0"
        I      0    ;empty
        O      1    ;empty
sth    i      0    ;auto op. ?
ETR
;-----
TR      1          ;input 0="1"
        I      0    ;empty
        O      3    ;empty
sth    i      0    ;manual op. ?
ETR
;-----
TR      2          ;manual
        I      3    ;empty
        O      9    ;empty
ETR
;-----
TR      3          ;empty
        I      9    ;empty
        O      0    ;empty
ETR
;-----
TR      4          ;input 8="1"
        I      1    ;empty
        O      2    ;empty
sth    i      8    ;start auto op. ?
ETR
;-----
TR      5          ;empty
        I      2    ;empty
        O     10    ;def. motion 1
ETR
;-----
TR      6          ;auto-operation
        I     10    ;def. motion 1
        O     14    ;empty
ETR
;-----
TR      7          ;empty
        I     14    ;empty
        O      2    ;empty
ETR
;-----
TR      8          ;input 9="1"
        I      3    ;empty
        O      7    ;forward
sth    i      9    ;manual forward ?
ETR
;-----
TR      9          ;input 10="1"
        I      3    ;empty
        O      5    ;backward
sth    i     10   ;manual backwards ?
ETR

```

```

;-----
TR      10          ;input 11="1"
        I      3          ;empty
        O      4          ;set zero position
sth    i    11      ;define zero pos. ?
ETR
;-----
TR      11          ;empty
        I      4          ;set zero position
        O      9          ;empty
ETR
;-----
TR      12          ;input 10="0"
        I      5          ;backward
        O      6          ;stop
stl    i    10      ;end zero pos. ?
ETR
;-----
TR      13          ;empty
        I      6          ;stop
        O      9          ;empty
ETR
;-----
TR      14          ;input 9="0"
        I      7          ;forward
        O      8          ;stop
stl    i    9       ;end manual forward ?
ETR
;-----
TR      15          ;empty
        I      8          ;stop
        O      9          ;empty
ETR
;-----
TR      16          ;breakpos. ?
        I      10         ;def. motion 1
        O      11         ;modify velocity
sth    f    BrkPos+FA1 ;Break Position reached ?
ETR
;-----
TR      17          ;breakpos. ?
        I      11         ;modify velocity
        O      12         ;modify velocity
sth    f    BrkPos+FA1 ;Break Position reached ?
ETR
;-----
TR      18          ;on dest. ?
        I      12         ;modify velocity
        O      13         ;ld time
sth    f    OnDest+FA1 ;on destination ?
ETR
;-----
TR      19          ;time = 0 ?
        I      13         ;ld time
        O      15         ;ld vel. & counter
stl    t    0
ETR
;-----
ETR

```

```

;-----
TR      20          ;motion 2
        I    15      ;ld vel. & counter
        O    18      ;empty
ETR
;-----
TR      21          ;empty
        I    18      ;empty
        O    14      ;empty
ETR
;-----
TR      22          ;empty
        I    15      ;ld vel. & counter
        O    16      ;ld posrel,start
ETR
;-----
TR      23          ;on dest ?
        I    16      ;ld posrel,start
        O    17      ;dec cnt & ld timer
sth f OnDest+FA1 ;on destination ?
ETR
;-----
TR      24          ;cnt & time=0
        I    17      ;dec cnt & ld timer
        O    18      ;empty
stl c 100
anl t 0
ETR
;-----
TR      25          ;cnt>0 & t=0
        I    17      ;dec cnt & ld timer
        O    16      ;ld posrel,start
sth c 100
anl t 0
ETR
;-----

ESB

```



## 10. Riassunto dei comandi e dei simboli

### Parametri operativi

FB Flag di Comando	Descrizione / Funzione	Tempo di elabo- razione	Può essere aggiornato o eseguito du- rante il mov.	Pagina	Valori di Ingresso /Uscita			
					Simbolo	Tipo	Formato	Descrizione / Funzione
FSeIOM	Select Operation Mode (Seleziona Modalità Operativa)	1,9ms	si	7-24	MCW	R	Binario	Motion Control Word (Parola di Controllo Movimento)
FSetPE	Set Position Error (Imposta Errore di Posizione)	1,9ms	si	7-26	PosEr MCW	R R	Intero Binario	Position Error (Errore di Posizione) Motion Control Word (Parola di Controllo Movimento)

### Comandi di movimento

FB Flag di Comando	Descrizione / Funzione	Tempo di elabo- razione	Può essere aggiornato o eseguito du- rante il mov.	Pagina	Valori di Ingresso/Uscita			
					Simbolo	Tipo	Formato	Descrizione / Funzione
FStart	Start Motion (Inizio Movimento)	1ms	si	7-43	--			
FStop	Stop Motion (Arresto Movimento)	2,7ms	si	7-44	MCW	R	Binario	Motion Control Word (Parola di Controllo Movimento)
FMotOff	Motor Off (Disattivazione Motore)	1,8ms	si	7-46	--			
FSSStepF	Single Step Forward (Singolo Pas- so Avanti)	4,5ms	no	7-47	--			
FSSStepB	Single Step Backwards (Singolo Passo Indietro)	4,5ms	no	7-48	--			
FForw	Forward with def. Velocity (Avanti a velocità definita)	4,5ms	si	7-49	--			
FBackw	Backwards with def. Velocity (Indietro a velocità definita)	4,5ms	si	7-50	--			

## Comandi di movimento

FB Flag di Comando	Descrizione / Funzione	Tempo di elabo- razione	Può essere aggiornato o eseguito du- rante il mov.	Pagina	Valori di Ingresso/Uscita			
					Simbolo	Tipo	Formato	Descrizione / Funzione
FLdDR	Load Destination Position Relative (Carica Pos. Relativa di Destinaz.)	4,1ms	si	7-30	DestP MCFac	R R	Intero virgola mobile	Destination Position (Pos. di Dest.) Motion Control Factor (Parola di Controllo Movimento)
FLdDA	Load Destination Position Absolute (Carica Pos. Assoluta di Destinaz.)	4,1ms	si	7-28	DestP MCFac	R R	Intero virgola mobile	Destination Position (Pos. di Dest.) Motion Control Factor (Parola di Controllo Movimento)
FLdVR	Load Velocity Relative (Carica Velocità Relativa)	4,5ms	si	7-34	Veloc MCFac	R R	Intero virgola mobile	Velocity (Velocità) Motion Control Factor (Parola di Controllo Movimento)
FLdVA	Load Velocity Absolute (Carica Velocità Assoluta)	4,5ms	si	7-32	Veloc MCFac	R R	Intero virgola mobile	Velocity (Velocità) Motion Control Factor (Parola di Controllo Movimento)
FLdAR	Load Acceleration Relative (Carica Accelerazione Relativa)	7,2ms	con condiz.	7-38	Accel MCFac	R R	Intero virgola mobile	Acceleration (Accelerazione) Motion Control Factor (Parola di Controllo Movimento)
FLdAA	Load Acceleration Absolute (Carica Accelerazione Assoluta)	7,2ms	con condiz.	7-36	Accel MCFac	R R	Intero virgola mobile	Acceleration (Accelerazione) Motion Control Factor (Parola di Controllo Movimento)
FLdRP	Load Regulator Parameter (Carica Parametri Regolatore)	5,2ms	si	7-40	KProp KInt KDer IntL  Sampl	R R R R  R	Intero Intero Intero Intero  Intero	Proportional Factor (Fatt. proporz.) Integral Factor (Fattore Integrale) Derivative Factor (Fattore Derivativo) Integration Limit (Limite di Integrazio- ne) Sampling Interval Derivative Term (Intervallo di campionamento termi- ne derivativo)
FUpDRP	Up Date Regulator Parameters (Aggiorna i Param. del Regolatore)	0,9ms	si	7-42	--			

**Comandi di lettura dati**

FB Flag di Comando	Descrizione / Funzione	Tempo di elabo- razione	Può essere aggiornato o eseguito du- rante il mov.	Pagina	Valori di Ingresso/Uscita			
					Simbolo	Tipo	Formato	Descrizione / Funzione
FRdAP	Read Actual Position (Lettura Posizione Attuale)	4,3ms	si	7-51	RActP MCFac	R R	Intero virgola mobile	Actual Position (Posizione Attuale) Motion Control Factor (Parola di Controllo Movimento)
FRdSP	Read Setpoint Position (Lettura Posizione Impostata)	4,3ms	si	7-52	RSetP MCFac	R R	Intero virgola mobile	Setpoint Position (Pos. di Setpoint) Motion Control Factor (Parola di Controllo Movimento)
FRdAV	Read Actual Velocity (Lettura Velocità Attuale)	2,8ms	si	7-53	RActV MCFac	R R	Intero virgola mobile	Actual Velocity (Velocità Effettiva) Motion Control Factor (Parola di Controllo Movimento)
FRdSV	Read Setpoint Velocity (Lettura Velocità Impostata)	3,7ms	si	7-54	RSetV MCFac	R R	Intero virgola mobile	Setpoint Velocity (Vel. di Setpoint) Motion Control Factor (Parola di Controllo Movimento)
FRdITS	Read Integration Term Sum (Lettura Somma di Integraz.)	1,9ms	si	7-55	RIntTS	R	Intero	Integration Term Sum (Somma di Integrazione)
FRdIP	Read Index Position (Lettura Posizione Indice)	4,3ms	si	7-56	RIndP MCFac	R R	Intero virgola mobile	Index Position (Posizione Indice) Motion Control Factor (Parola di Controllo Movimento)
FRdSR	Read Signal Register (Lettura Registro Segnali)	1,8ms	si	7-57	RSigB	R	Binario	Signalisation Bits (Bit Segnalazione)

**Comandi vari**

FB Flag di Comando	Descrizione / Funzione	Tempo di elabo- razione	Può essere aggiornato o eseguito du- rante il mov.	Pagina	Valori di Ingresso/Uscita			
					Simbolo	Tipo	Formato	Descrizione / Funzione
FResSF	Reset Status Flag (Azzera Flag di Stato)	1,8ms	si	7-60	StaFRR	R	Binario	Status Flag Reset Register (Registro Reset Flag di Stato)
FLdBPR	Load Break Position Relative	3,3ms	si	7-64	BrkP	R	Intero	Break Position (Posizione di Break- point)
	(Carica Pos. Relativa di Breakpoint)				MCFac	R	virgola mobile	Motion Control Factor (Fattore di Controllo Movimento)
FLdBPA	Load Break Position Absolute	3,3ms	si	7-62	BrkP	R	Intero	Break Position (Posizione di Break- point)
	(Carica Pos. Assoluta di Breakpoint)				MCFac	R	virgola mobile	Motion Control Factor (Fattore di Controllo Movimento)
FSetIP	Set Index Position (Imposta Posizione Indice)	0,9ms	si	7-66	--			
FSetZP	Set Zero Position (Imposta Posizione Zero)	0,9ms	si	7-67	--			

# Menu generale

**SAIA®PCD per l'automazione industriale**

**Gamma di prestazioni SAIA®PCD**

**Manuali**

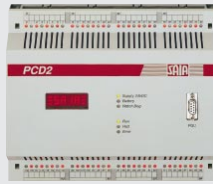
**SAIA®PCD for Building automation**

**Strumenti di sviluppo**

**Informazioni generale**



**Hardware PCD1**



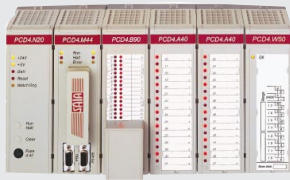
**Hardware PCD2**



**PLC + PC: PCD2.M220**



**I/U remote  
PCD0 e PCD1.RIO**



**Hardware PCD4**



**Hardware PCD6**



**Terminali**



**Serie xx7:  
II PLC compatibile  
con SIMATIC® S7**

# Menù generale

**Gamma di prestazioni dal PCD1 fino al PCD6**



**PCD1:  
un concentrato di potenza**



**PCD2:  
il modulare compatto dalle prestazioni eccezionali**



**Series xx7:  
il PLC compatibile con SIMATIC® S7**



**SAIA®PCD  
per l'automazione industriale**



**PCD4:  
la flessibilità del PLC di media grandezza**



**PCD6:  
il PLC di alta gamma per sistemi multiprocessore**



**PCD2.M250:  
PLC + PC integrati in una unità industriale**



**Dai piccoli terminali di testi ai terminali «Touch Screen»**



**SAIA®PCD  
per la Building automation**

**Strumenti di programmazione**

**Manuali**

**Informazioni generale**