Sommario

DEVICE CAMMING2 ... 3
1. Introduzione ... 3
1.1 Installazione ... 4
1.1.1 Dichiarazione device nel file di configurazione (.CNF) ... 4
1.2 Funzionamento ... 5
1.2.1 Calcolo della risoluzione ... 5
1.2.2 Punto decimale ... 6
1.2.3 Velocità ... 6
1.2.4 Comandi principali ... 6
1.2.5 Cambio velocitá e tempo di rampa in movimento ... 7
1.2.6 Descrizione del movimento epicicloidale ... 8
1.2.7 Riduzione del profilo ... 9
1.2.8 Tipo di stop durante la rampa di accelerazione ... 10
1.2.9 Calibrazione uscita analogica ... 11
1.2.10 Movimentazione ... 12
1.2.11 Taratura PID+FF ... 12
1.2.12 Applicazione di movimentazione ... 15
1.2.13 La struttura dei settori ... 16
1.3 I settori ... 16
1.3.1 Il settore di accelerazione ... 17
1.3.2 Il settore di decelerazione ... 20
1.3.3 Il settore di cambio velocità ... 22
1.3.4 Il settore di Start sincronizzato al Master ... 25
1.3.5 Il settore di fine camma ... 26
1.3.6 Il settore di absolute jump ... 26
1.3.7 Il settore di jump condizionato ... 26
1.3.8 Il settore di loop camma ... 27
1.3.9 Il settore non operativo ... 27
1.3.10 Definizione di settori a campionamento zero ... 27
1.3.11 I settori di aggiornamento conteggio ... 27
1.3.12 Descrizione settori camma ... 28
1.3.13 Basi per la costruzione di una camma per spandifilo ... 29
1.3.14 Basi per la costruzione di una camma per taglio al volo con extravelocità ... 30
1.4 Gestione errori device ... 31
1.5 Gestione warning device ... 32
1.6 Gestione master simulato ... 32
1.6.1 Esempio di programmazione ... 33
1.7 Limitazione rapporto frequenze trasduttore M/S ... 33
1.8 Tabella di configurazione ingressi ... 33
1.9 Tabella di configurazione uscite ... 34
1.10 Tabella comandi, stati e parametri: Simbologia adottata ... 34
1.10.1 Condizioni ... 34
1.10.2 Parametri ... 34
1.10.3 Variabili asse ... 37
1.10.4 Variabili di programma ... 39
1.10.5 Comandi ... 40
1.10.6 Stati ... 42
1.10.7 Limitazioni del device ... 44

DEVICE CAMMING2

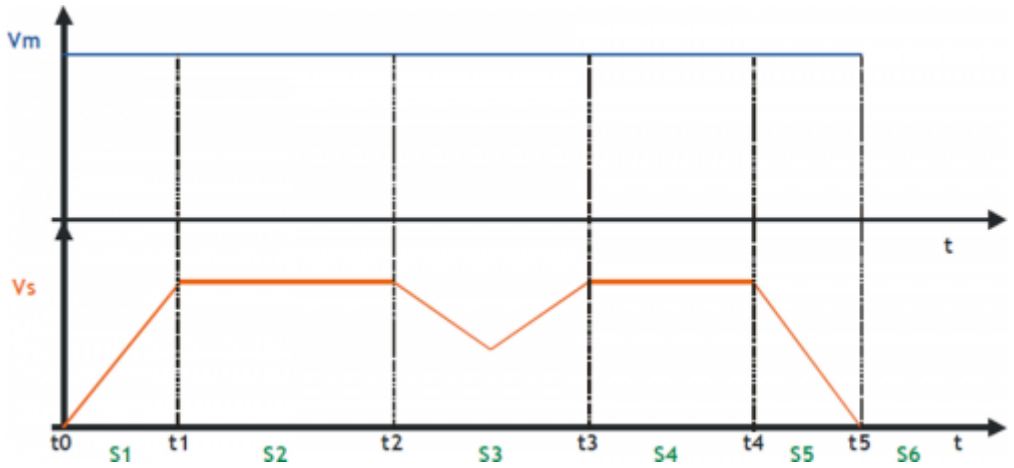
1. Introduzione

Il camming, è una tecnica di controllo del moto applicabile ad assi servoassistiti e consente di risolvere applicazioni in cui uno o più assi “slave” devono percorrere spazi, anche non omogenei, rimanendo sempre in sincronismo rispetto alla posizione di un asse di riferimento chiamato “master”. L’asse master può essere un asse reale o virtuale (master simulato).
Le tipiche applicazioni sono:

- Tagli e lavorazioni al volo, sia lineari che circolari, su plastica, lamiera, cartone.
- Nel confezionamento in sostituzione delle camme meccaniche.
- Nella bobinatura di cavo, filo metallico, reggia ecc. con funzioni di guidafilo.
- Nel tessile e nell’alimentare nelle macchine “affaldatrici” per la stratificazione di tessuti o pasta alimentare.
- Nella stampa serigrafica o flexografica con clichè circolari.
- Nelle linee di “trasporto prodotto” per la spaziatura e/o la sincronizzazione dei materiali movimentati.

La posizione assoluta che deve assumere l’asse slave è sempre espressa in funzione della posizione assoluta dell’asse master e questa associazione viene inserita in una tabella specifica detta “cam table”.
La “cam table” è composta da 40 settori; ogni settore è composto da: **CodeG** = *istruzione operativa* del settore in uso.
CodeQm = posizione *incrementale* del master, in unità di misura; sono accettati incrementi solo positivi.
CodeQs = posizione *incrementale* dello slave , in unità di misura; sono accettati incrementi sia positivi che negativi.
CodeM = codice numerico generale, utilizzabile dalla logica PLC.
CodeQma =quota master ausiliaria utilizzata con le istruzioni operative speciali.
CodeQsa = quota slave ausiliaria utilizzata con le istruzioni operative speciali.

Utilizzando le istruzioni operative codeG associate a ciascun settore della camma si può definire con quale legge di moto (accelerazione, decelerazione, velocità costante…) l’asse slave si deve muovere percorrendo lo spazio stabilito in **codeQs** nello stesso tempo in cui il master percorre lo spazio definito come **codeQm**.
Finchè il master si muove a velocità costante, lo spazio percorso dall’asse master risulta direttamente proporzionale al tempo trascorso ed essendo gli spazi **codeQs** e **codeQm** definiti sempre nello stesso intervallo di tempo anche la legge di moto applicata all’asse slave, all’interno del settore, risulta applicabile in modo direttamente proporzionale allo spazio percorso dal master nel settore; il master e lo slave risultano perciò legati in spazio tra loro.
Se la velocità costante scelta per il master corrisponde alla massima sarà possibile valutare immediatamente anche quali saranno le massime accelerazioni, decelerazioni e velocità a cui verrà sottoposto l’asse slave.
Questo procedimento consente di formulare la legge di moto dell’asse slave in funzione del tempo per valutare le prestazioni dinamiche richieste dall’applicazione e di applicare poi la stessa legge di moto in funzione dello spazio percorso dal master durante l’esecuzione della camma.
Per rendere semplice il calcolo delle posizioni assolute del master e dello slave si assume che il master si muova ad una velocità costante per cui le posizioni degli assi possono essere rappresentate in un diagramma cartesiano Velocità / Tempo. Di seguito si riporta un semplice esempio di compilazione della “cam table”.

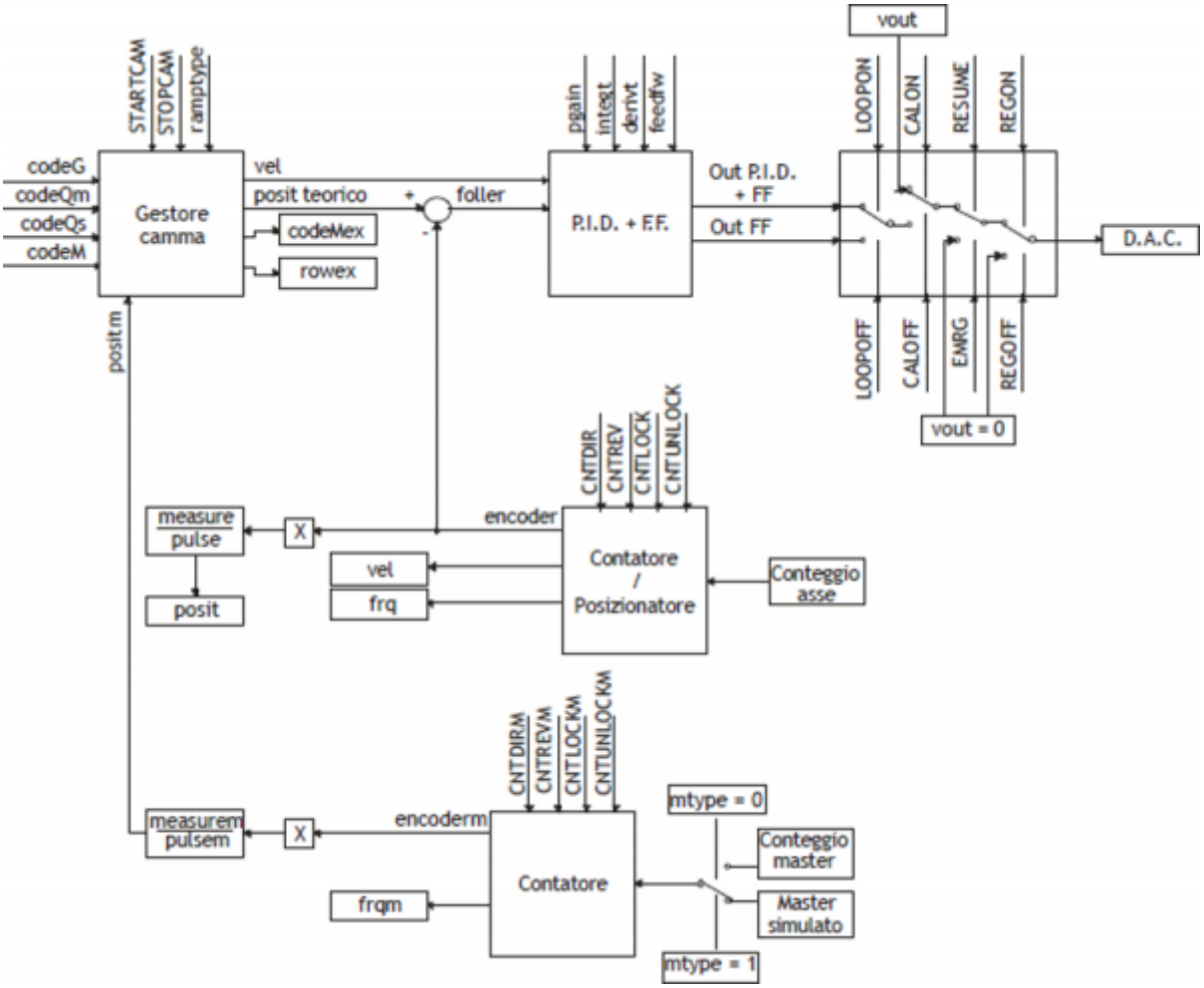


Per poter eseguire una camma come nell’esempio, bisogna compilare la “cam table” nel modo seguente:

Settore	CodeG	CodeQm	CodeQs	
S1	132	100	50	Settore di accelerazione con Vs = Vm alla fine del settore
S2	133	200	200	Settore intermedio a velocità costante
S3	134	160	120	Settore di compensazione con velocità iniziale = vel. finale
S4	133	150	150	Settore intermedio a velocità costante
S5	135	90	45	Settore di decelerazione con Vs = 0 alla fine del settore

Settore	CodeG	CodeQm	CodeQs	
S6	136	-	-	Comando di fine camma

Naturalmente QEM rimane a disposizione per aiutare i clienti nella compilazione della “cam table”.
Il device si può dividere in due parti principali:
Un posizionatore asse slave con rampe trapezoidali o epicicloidali selezionabili. Un gestore camme analogiche.
Lo schema a blocchi di base è il seguente:



1.1 Installazione

1.1.1 Dichiarazione device nel file di configurazione (.CNF)

Nel file di configurazione (.CNF), la sezione BUS deve essere dichiarata in modo tale che siano presenti le risorse hardware necessarie all'implementazione del device CAMMING2. Devono essere presenti almeno due contatori bidirezionali ed una uscita analogica con risoluzione 16 bit. Nella sezione INTDEVICE del file .CNF deve essere aggiunta la seguente definizione:

```
;-----------------------------
; Dichiarazione device interni
;-----------------------------
  INTDEVICE
        <NOME_DEVICE> CAMMING2 TCAMP COUNTS COUNTM INTL IAZERO INTLM
                            IAZEROM ING INGINT IOUTA OUT
```

Dove:

INTDEVICE	È la parola chiave che indica l'inizio della definizione di device interni.
nome_device	È il nome del device.
CAMMING2	È la parola chiave che identifica il device descritto in questo documento.
TCamp	Tempo di campionamento device (1÷250 ms).
CountS	Indirizzo contatore bidirezionale Slave
CountM	Indirizzo contatore bidirezionale Master
IntL	Numero della linea di interrupt dedicata per l'impulso di zero dell'encoder Slavedurante la fase di ricerca di preset. Valori ammessi: 1÷8 (per evitare che il device utilizzi questa risorsa, inserire il carattere X).
IAZero	Ingresso abilitazione impulso di zero slave (per evitare che il device utilizzi questa risorsa, inserire il carattere X.X)

IntLM	Numero della linea di interrupt dedicata per l'impulso di zero dell'encoder Master durante la fase di ricerca di preset. Valori ammessi: 1÷8 (per evitare che il device utilizzi questa risorsa, inserire il carattere X).
IAZeroM	Ingresso abilitazione impulso di zero master (per evitare che il device utilizzi questa risorsa, inserire il carattere X.X)
InG	Ingresso per funzione generica come descritto nel paragrafo di tabella configurazione ingressi (per evitare che il device utilizzi questa risorsa, inserire il carattere X.X)
InGInt	Numero della linea di interrupt dedicata ad una funzione generica come descritto nel paragrafo di tabella configurazione ingressi . Valori ammessi: 1÷8 (per evitare che il device utilizzi questa risorsa, inserire il carattere X).
IoutA	Indirizzo hardware del componente DAC dell'uscita analogica Slave.
Out	Uscita per funzione generica come descritto nel paragrafo di tabella configurazione uscite (per evitare che il device utilizzi questa risorsa, inserire il carattere X.X)

1.1.1.1 Esempio applicativo

Si prende come esempio un device CAMMING2 configurato come nello START UP e con la parametrizzazione dell'asse (set-up) già scritto.
Nel task viene prima inizializzato il device e poi gestito un ingresso in interruzione il quale riporta il suo stato su un'uscita.
Il task verrà così svolto:

```
;-------------------------------------
; Gestione del device CAMMING2
;-------------------------------------
INIT AsseX                          : Inizializza l'asse
WAIT AsseX:st_init                  ; Attendi che l'asse sia inizializzato
LOOPON AsseX                        ; Aggancia il loop di regolazione
WAIT AsseX:st_loopon                ; Attendi che l'asse abbia agganciato il
                                    ; loop di regolazione
CALOFF AsseX                        ; Esci dall'eventuale calibrazione
                                    ; dell'asse
WAIT NOT AsseX:st_cal               ; Attendi che il device non sia in
                                    ; calibrazione
CNTUNLOCK AsseX                     ; Sblocca il contatore master
WAIT NOT AsseX:st_cntlock           ; Attendi che il contatore master sia
                                    ; sbloccato
CNTDIR AsseX                        ; Imposta il giusto senso di incremento
                                    ; del contatore slave
WAIT NOT AsseX:st_cntrev            ; Attendi che il contatore slave sia
                                    ; impostato nel senso di incremento
CNTUNLOCKM AsseX                    ; Sblocca il contatore master
WAIT NOT AsseX:st_cntlockm          ; Attendi che il contatore master sia
                                    ; sbloccato
CNTDIRM AsseX                       ; Imposta il giusto senso di incremento
                                    ; del contatore master
WAIT NOT AsseX:st_cntrevm           ; Attendi che il contatore master sia
                                    ; impostato nel senso di incremento
REGON AsseX                         ; Sblocca la regolazione
WAIT NOT AsseX:st_regoff            ; Attendi lo sblocco della regolazione

MAIN:

IF AsseX:st_int                     ; Se la linea di interrupt è attiva
  AsseX:funOut = 2                  ; attiva l'uscita
ELSE
  AsseX:funOut = 1                  ; disattiva l'uscita
ENDIF

ENDIF                               ; FINE

WAIT 1
JUMP MAIN
END
```

1.2 Funzionamento

1.2.1 Calcolo della risoluzione

Il device CAMMING2 non ha al suo interno il parametro *cntratio*, ma lascia all'installatore la possibilità di lavorare con risoluzioni encoder non finite impostando i dati come spazio percorso in un giro encoder (*measure*) e numero di impulsi giro dell'encoder (*pulse*).
Il rapporto tra *measure* e *pulse* è la risoluzione dell'encoder e deve avere valori compresi tra 1 e 0.000935.

1.2.1.1 Definizioni:

- Il parametro *measure* viene inserito in unità di misura senza punti decimali (ad esempio 100.0 millimetri viene inserito 1000 decimi di millimetro).
- Il parametro *pulse* viene inserito in bit encoder per 4 (ad esempio se ho collegato un encoder da 1024 impulsi giro, viene inserito 4096, se il parametro measure viene calcolato su un giro di encoder).

1.2.1.2 Esempio:

Si deve controllare una tavola rotante che abbia la precisione di 0,1° avente un encoder da 1024 impulsi giro calettato direttamente; si imposteranno i seguenti valori:
measure = 3600

pulse = 4096

1.2.2 Punto decimale

Se per l'unità di misura scelta é prevista anche la presenza di un punto decimale, le posizioni devono essere rappresentate sempre come valore intero e rappresentare lo spazio nell'unità di misura senza punto decimale. La risoluzione deve quindi essere calcolata con lo stesso metodo e nel parametro *measure* la grandezza senza punto decimale. Il punto decimale verrà poi inserito nei visualizzatori in fase di rappresentazione del valore (es. come proprietà nel terminale operatore). Questo parametro può assumere valori 0÷3.

1.2.3 Velocità

Le velocità sono sempre espresse in unità di misura intere nell'unità di tempo scelta. Da questo si ricava che il device deve comunque conoscere la posizione del punto decimale dell'unità di misura e questo viene fatto con il parametro *decpt*.

1.2.4 Comandi principali

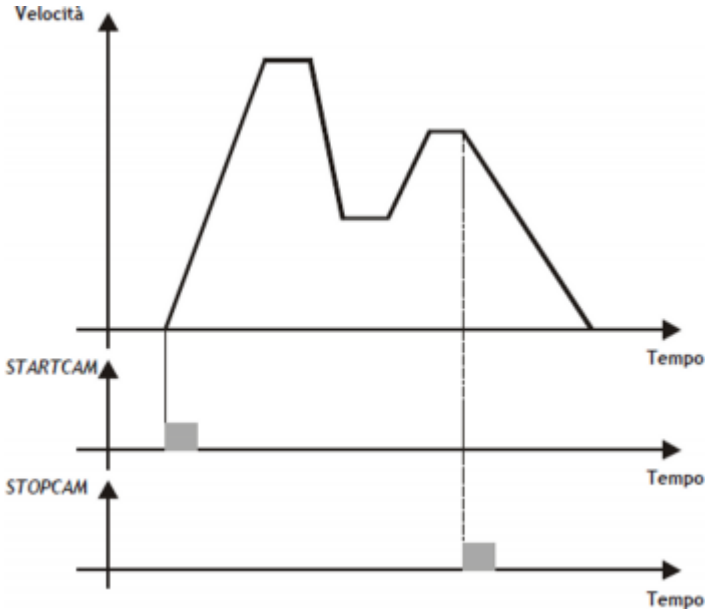
In questo paragrafo viene descritto solamente l'utilizzo di alcuni di comandi; per le descrizioni relative a tutto il set di comandi si rimanda ai capitoli seguenti.
I due comandi principali sono quello che danno inizio ed interrompono l'esecuzione della camma: *STARTCAM* e *STOPCAM*.
Esistono poi una serie di comandi dediti al controllo dell'emergenza, il loop di reazione, lo START e lo STOP all'asse.

1.2.4.1 STARTCAM

Al comando *STARTCAM*, l'asse slave si aggancia al master e seguirà l'andamento descritto nella camma partendo sempre dal primo settore. Non è possibile dare uno *STARTCAM* durante l'esecuzione della camma (*st_camex* = 1); tale controllo viene lasciato al programmatore.
La camma si sgancerà automaticamente se incontrerà un istruzione di *END* oppure sarà possibile fermarla in rampa utilizzando il comando di *STOPCAM*.

1.2.4.2 STOPCAM

Se la camma è in esecuzione (*st_camex* = 1), una volta ricevuto il comando di *STOPCAM* l'asse slave si sgancia immediatamente dal master, porta la sua velocità a zero seguendo la rampa di decelerazione impostata (parametro *tdec*) e rimanendo in reazione di spazio. La rampa di decelerazione è asincrona rispetto al master.



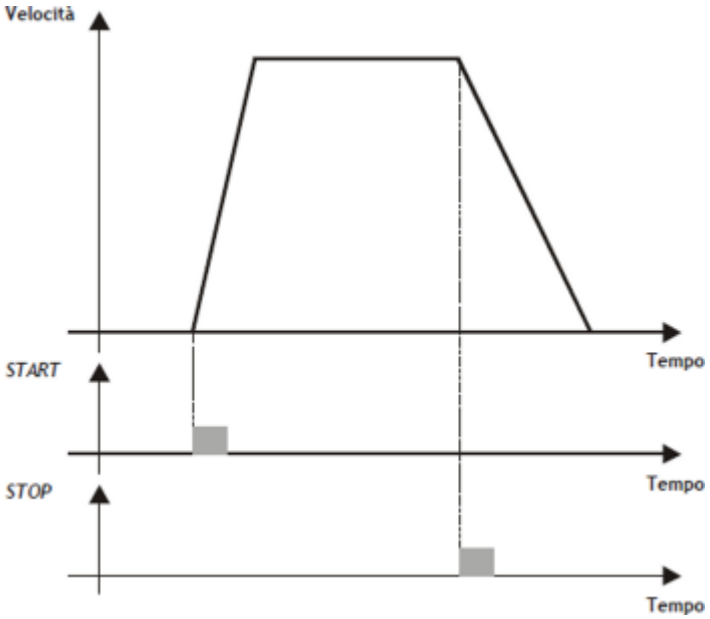
1.2.4.3 START

Al comando di *START*, l'asse slave si posiziona alla quota dichiarata nella variabile *setpos* con la velocità impostata in *setvel*; il posizionamento verrà eseguito utilizzando la rampa di accelerazione impostata nel parametro *tacc* e la rampa di decelerazione

impostata nel parametro *tdec*. Il tipo di rampa utilizzata (trapezoidale o epicicloidale) è inserita nel parametro *ramptype*.

1.2.4.4 STOP

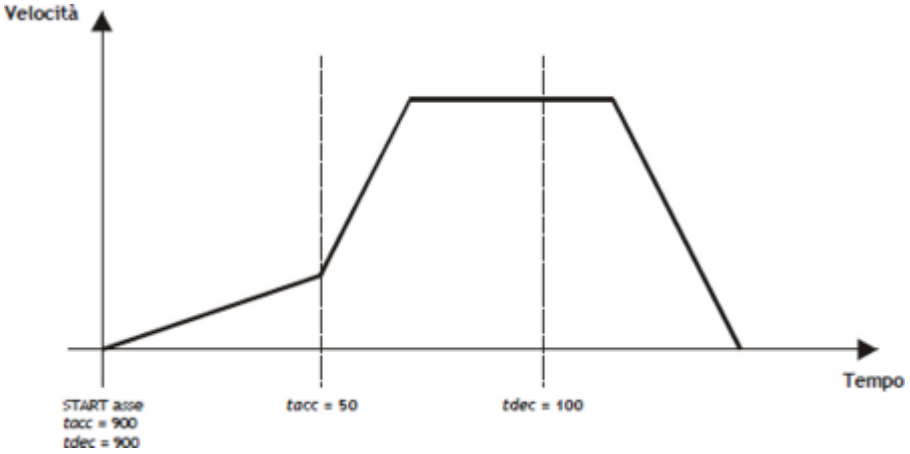
Se durante il posizionamento (non durante l'esecuzione di una camma) è necessario fermare l'asse con una rampa di decelerazione, sarà sufficiente dare il comando di *STOP* e l'asse decelera fino a fermarsi con la rampa impostata nel parametro *tdec*.



1.2.5 Cambio velocitá e tempo di rampa in movimento

Durante il posizionamento é possibile variare la velocitá dell'asse senza influenzare la posizione da raggiungere. Questa operazione può determinare un aumento o una diminuzione della velocitá, anche in più punti dello stesso posizionamento. Questa operazione viene eseguita con nuova scrittura nel parametro *setvel*. Il cambio di velocità è sempre disponibile tranne durante la rampa di decelerazione e uno stato apposito segnala il possibile cambio di velocità (*st_chvel* = 1)

Durante il posizionamento possono essere variati anche i tempi di accelerazione/decelerazione. Per esempio il device può avviare un posizionamento con una rampa molto breve e, una volta raggiunta la velocitá impostata, viene variato il parametro *tacc* ed eseguito un cambio di velocitá con una rampa molto lunga.
Per applicazioni particolari e in presenza di rampe trapezoidali, il tempo di rampa può essere variato anche durante una variazione di velocitá, in questo caso il nuovo tempo viene messo in esecuzione immediatamente.



1.2.5.1 EMRG

Questo comando mette l'asse in condizioni di emergenza; lo stato *st_emrg* viene posto ad uno. Se il comando di emergenza viene inviato all'asse durante un posizionamento, il movimento viene interrotto senza rampa di decelerazione, l'uscita analogica viene impostata a zero volt e viene sganciata la reazione di spazio. Se la camma è attiva (*st_camex* = 1), il movimento viene interrotto senza rampa di decelerazione, l'uscita analogica viene impostata a zero volt, viene sganciata la reazione di spazio e la camma (*st_camex* = 0).
Con *st_emrg* = 1 (condizione di emergenza), non è possibile movimentare l'asse.

1.2.5.2 RESUME

Con questo comando viene resettata la condizione di emergenza; l'asse entra in reazione di spazio ed attende un comando per potersi muovere (non riprende automaticamente il posizionamento interrotto).

1.2.5.3 LOOPOFF

Il comando LOOPOFF toglie la reazione di spazio senza fermare l'asse. Con *st_loopon* = 0 l'asse accetta i comandi di movimentazione asse ma tutti i posizionamenti saranno eseguiti senza reazione di spazio.
Un posizionamento fatto senza loop di reazione è paragonabile ad un posizionamento eseguito senza guadagno proporzionale (non viene garantito l'arrivo in posizione).

1.2.5.4 LOOPON

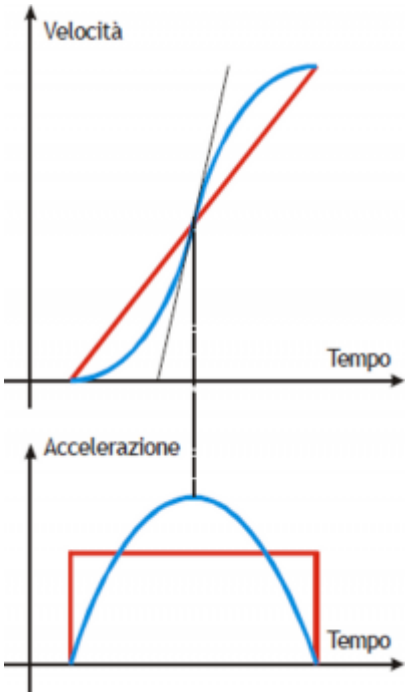
Il comando *LOOPON* chiude l'anello di spazio senza fermare l'asse. Con *st_loopon* = 1 l'asse viene movimentato utilizzando tutte le caratteristiche del controllo P.I.D.

A seguito si riporta una tabella che riassume le condizioni necessarie per avere l'asse in reazione di spazio e per eseguire dei posizionamenti.

Loopon	Emrg	Reazione di spazio	Possibilità di movimento
SI	NO	SI	SI
SI	SI	NO	NO
NO	NO	NO	SI
NO	SI	NO	NO

1.2.6 Descrizione del movimento epicicloidale

Il movimento epicicloidale viene utilizzato per movimentare gli assi senza brusche variazioni di velocità. Il tempo di posizionamento di un asse movimentato con le rampe trapezoidali è lo stesso rispetto allo stesso asse movimentato con le rampe epicicloidali, ma le rampe epicicloidali variano il gradiente di velocità (accelerazione) con un massimo a metà della rampa stessa.
Per confronto viene mostrata la differenza dell'andamento dell'accelerazione nei due casi: con rampa lineare (trapezoidale) e con rampa epicicloidale.

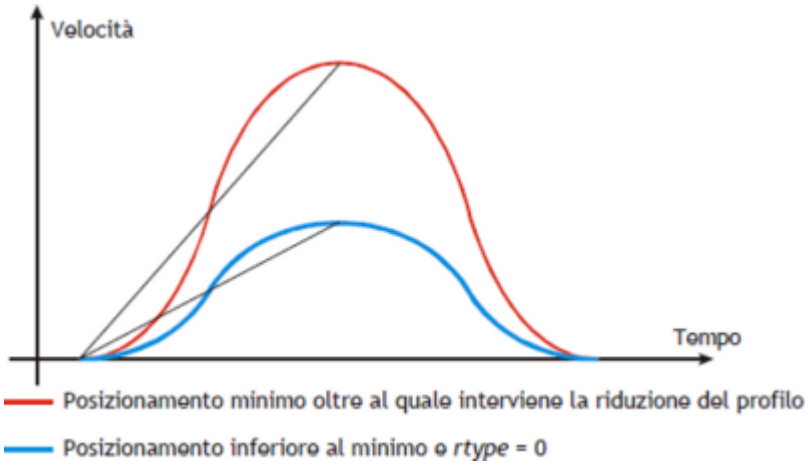


Lo stesso vale anche per la rampa di decelerazione.
Il movimento epicicloidale ha la possibilità di comportarsi in modi diversi nel caso di riduzione di profilo (*rtype*) e nel caso di stop durante la rampa di accelerazione (stopt) se la camma non è in esecuzione (*st_camex* = 0).

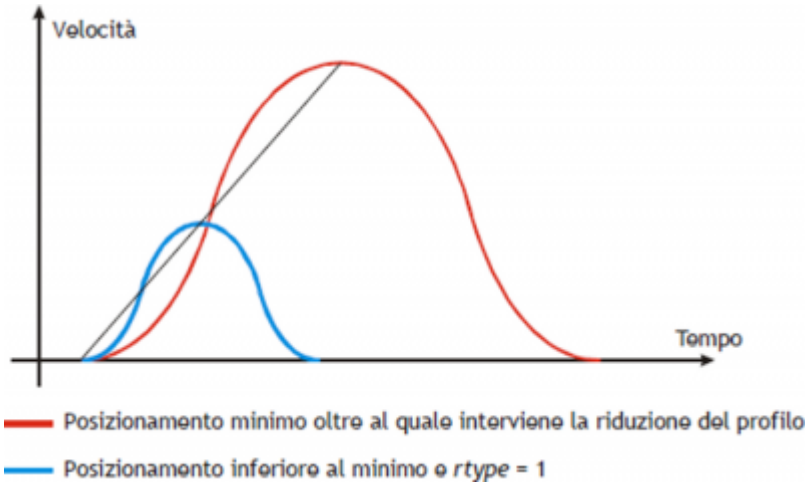
1.2.7 Riduzione del profilo

La riduzione del profilo viene utilizzata solamente se si sta eseguendo un posizionamento e non se si sta eseguendo una camma (*st_camex* = 0).

Nel caso in cui la camma non è in esecuzione (*st_camex* = 0) e lo spazio da percorrere sia minore di quello che consente di raggiungere la velocità impostata eseguendo le rampe di accelerazione e decelerazione, si passa nella fase chiamata "riduzione di profilo".
È possibile mantenere fisso il tempo delle rampe, diminuendo i gradienti delle rampe e la velocità in proporzione (parametro *rtype* impostato a 0).



È inoltre possibile diminuire il tempo delle rampe mantenendo il gradiente di accelerazione costante e diminuire la velocità in proporzione (parametro *rtype* impostato a 1).

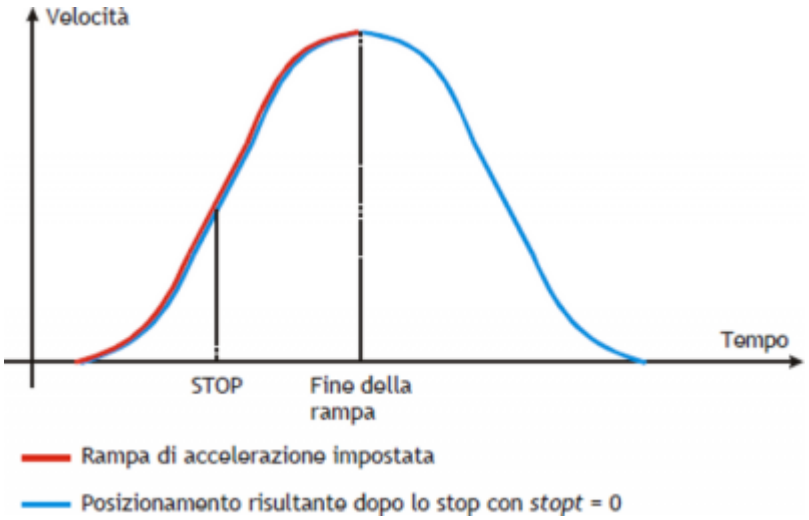


Con il parametro rtype impostato a 0 si allungano notevolmente i tempi necessari ai posizionamenti piccoli con relativa perdita di produttività della macchina, invece impostandolo a 1 si hanno nel caso di posizionamenti brevi tempi ridotti, ma mantenendo il gradiente costante si perde l'effetto benefico dell'epicicloide.

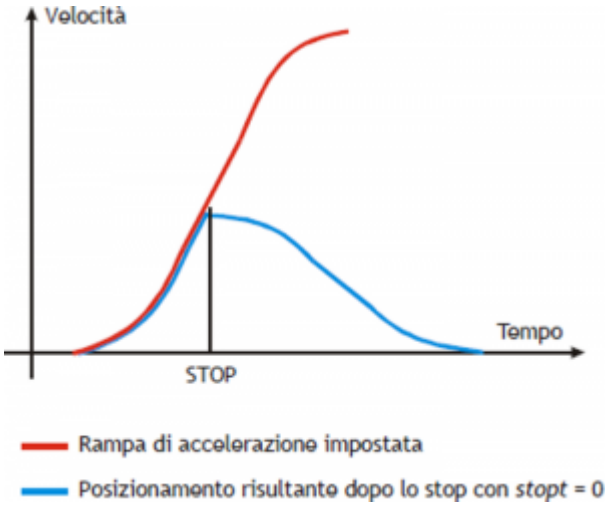
1.2.8 Tipo di stop durante la rampa di accelerazione

Il tipo di stop durante le rampe viene utilizzato solamente se si sta eseguendo un posizionamento e non se si sta eseguendo una camma (*st_camex* = 0).

Nel caso in cui la camma non è in esecuzione (st_camex = 0) e si debba frenare l'asse durante la rampa di accelerazione con il comando di *STOP* si deve scegliere se far completare la rampa oppure se si vuole interrompere la rampa e di conseguenza modificare l'epicicloide.
Nel caso in cui si imposti il parametro *stopt* a 0 viene prima completata la rampa di accelerazione e poi eseguita la rampa di decelerazione.



Nel caso in cui si imposti il parametro stopt a 1 viene interrotta la rampa di accelerazione e iniziata immediatamente la rampa di decelerazione impostata.



Si nota immediatamente che esiste una differenza sostanziale tra il settaggio di *stopt* a 0 o a 1. Per fare la scelta di quale tipo di stop utilizzare, bisogna tener conto che in caso di fermata di emergenza esiste il comando di emergenza che blocca istantaneamente e senza rampa il posizionamento.

1.2.9 Calibrazione uscita analogica

Prima di iniziare dei posizionamenti veri e propri è necessario verificare che collegamenti elettrici ed organi meccanici non siano causa di malfunzionamenti.

Per la gestione dell'asse, il device utilizza un'uscita analogica con range ±10 V e risoluzione 16 bit con segno; con la funzione di calibrazione questa uscita analogica può essere pilotata con un valore costante con lo scopo di verificare collegamenti e funzionalità.

1.2.9.1 Movimentazione preliminare

- Togliere la condizione di emergenza con il comando *RESUME*.
- Lo stato *st_emrg* = 0
- Abilitare lo stato di taratura asse con il comando *CALON*; lo stato st_cal deve quindi assumere il valore 1.
- É ora possibile impostare la tensione analogica con il parametro *vout*; il valore é espresso in decimi di volt (-100 ÷ 100 = -10 ÷ 10 V). Si consiglia di introdurre valori bassi (5, 10, 15… pari a 0.5, 1, 1,5 V).
- Quando l'asse é in movimento il parametro *frq* indica la frequenza in Hz delle fasi del trasduttore.
- Il parametro *posit* che visualizza la posizione, varia indicando lo spazio compiuto dall'asse. Se impostando una tensione positiva il conteggio si decrementa, é necessario invertire le fasi del trasduttore o invertire la direzione nell'azionamento.
- È possibile invertire la direzione del conteggio utilizzando il comando *CNTREV*.
- Se con tensione di uscita uguale a zero l'asse non é fermo, agire sul parametro *offset* per correggere la tensione finché il movimento non si arresta. Il valore introdotto (ogni bit corrisponde a circa 0.3 mV), viene sommato algebricamente al valore dell'uscita analogica; questa operazione permette di compensare l'eventuale deriva propria del componente elettronico, sia in uscita da QMOVE che in ingresso all'azionamento. Il valore é espresso in bit con segno.
  Per un ottimale risultato della taratura l'operazione deve essere eseguita con il sistema a regime di temperatura.
- Per disabilitare lo stato di taratura inviare il comando *CALOFF*.
- Lo stato *st_cal* = 0

1.2.9.2 Parametrizzazione uscita

Il device genera il valore di tensione dell'uscita analogica sulla base di una proporzione tra la velocitá massima dell'asse e la massima tensione di uscita. La proporzionalità è ottenuta con il parametro *maxvel*, rappresentante la velocitá dell'asse relativa alla massima tensione analogica (10 V). Ovviamente l'asse deve avere un comportamento simmetrico rispetto al valore zero di tensione analogica, quindi la velocitá deve essere la stessa sia alla tensione massima positiva che alla massima negativa.
Prima di determinare il valore della velocitá massima, bisogna stabilire l'unità di tempo da utilizzare per la rappresentazione delle velocitá nel device; il parametro *unitvel* definisce l'unità di tempo della velocità (Um/min oppure Um/s).

1.2.9.3 Metodo teorico per la determinazione della velocità massima.

Il metodo teorico é un calcolo eseguito sulla base della velocitá massima del motore. Una volta stabiliti i giri massimi al minuto dichiarati del motore, si ricava la velocitá massima espressa nell'unità di misura sull'unità di tempo scelti.
Introdurre il valore di velocitá massima calcolato nel parametro *maxvel*.

1.2.9.4 Metodo pratico per la determinazione della velocità massima.

Il metodo pratico si basa sulla lettura della velocitá rilevata dal device nel parametro *vel*, fornendo all'azionamento una tensione nota. Per fornire la tensione all'azionamento il device deve essere posto nella condizione di calibratura come descritto nel paragrafo precedente. Se il sistema lo permette, fornire all'azionamento una tensione di 10 V e leggere il valore di velocità nel parametro *vel*. Se, al contrario, viene fornita una porzione della tensione in uscita (1, 2, ... 5 V), calcolare la velocitá massima con una proporzione.
Introdurre il valore trovato di velocitá massima nel parametro *maxvel*.

1.2.10 Movimentazione

Prima di movimentare l'asse, verificare il corretto funzionamento dei dispositivi di emergenza e protezione.

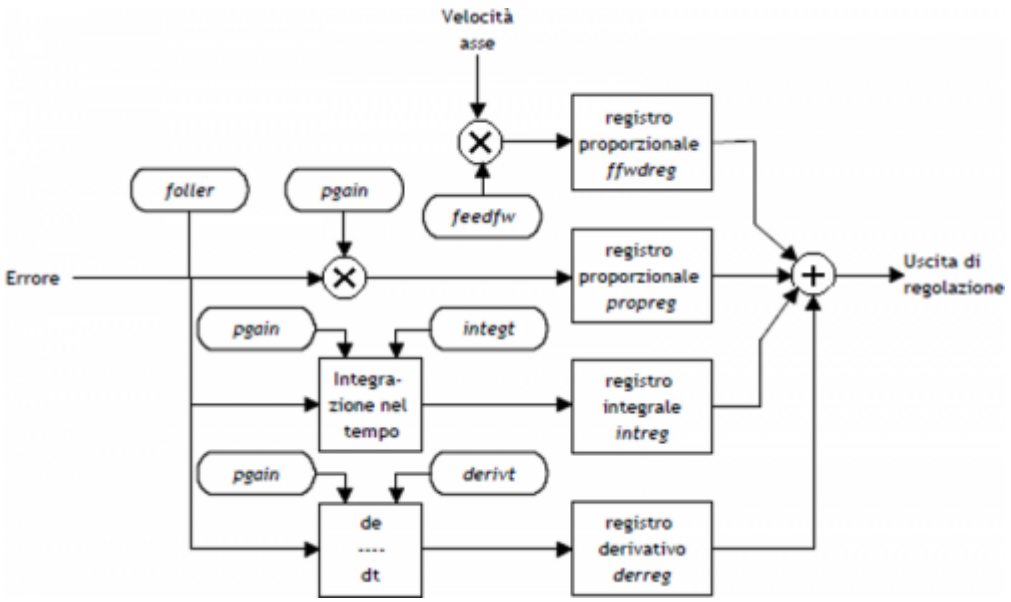
Le procedure fin qui descritte hanno permesso di completare la prima fase di parametrizzazione del device. Ora é possibile eseguire una semplice movimentazione dell'asse.

- Spostare l'asse in una posizione tale per cui possa compiere un determinato spazio senza toccare i finecorsa di quota massima e minima.
- Impostare la posizione attuale dell'asse al valore zero, settando il parametro *posit* = 0.
- Impostare i parametri che definiscono la posizione dei finecorsa software: *minpos* = 0 e *maxpos* al valore della corsa massima dell'asse.
- Impostare il parametro che definisce il tempo impiegato dall'asse per raggiungere la velocitá massima *tacc* = 100. Questo parametro é espresso in centesimi di secondo (100 = 1 sec.)
- Impostare la velocitá di posizionamento con il parametro *setvel*.
- Impostare la quota di destinazione con il parametro *setpos*.
- Impostare il parametro *feedfw* = 1000 (100%)
- Se il device é in stato di emergenza (st_emrg = 1) dare il comando *RESUME*.
- Avviare il posizionamento con il comando *START*. In per arrestare il movimento dare il comando *EMRG*.

Questa prima movimentazione é stata eseguita senza la retroazione di spazio. Il posizionamento potrebbe essere stato eseguito con un certo errore introdotto dalla non linearità dei componenti o da una imperfezione nella taratura della velocitá massima. Successivamente abilitando la retroazione di spazio questo errore scompare.

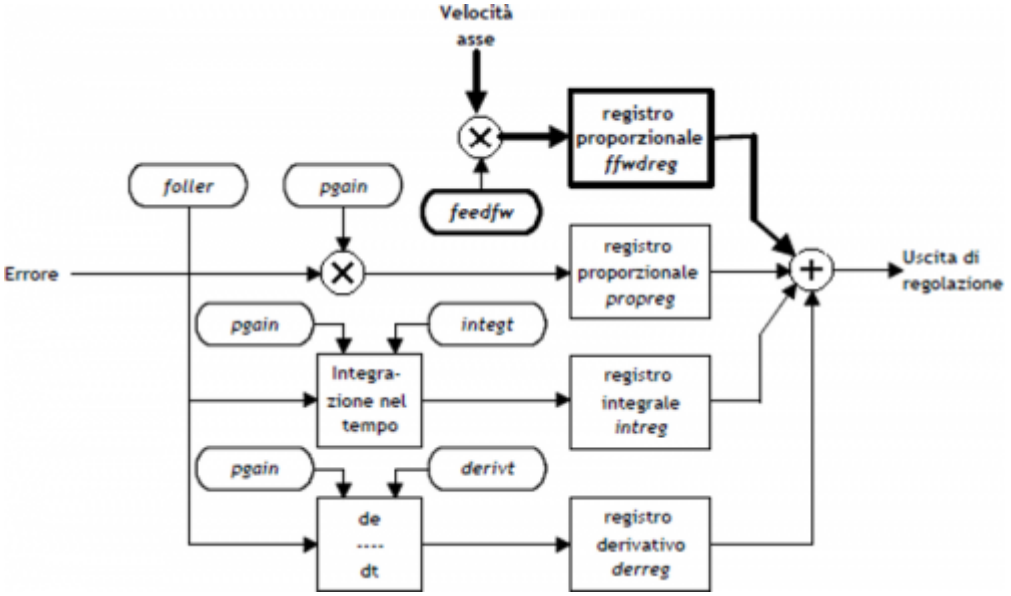
1.2.11 Taratura PID+FF

Il posizionamento eseguito nel paragrafo precedente é stato realizzato senza considerare eventuali errori di posizione.
Per controllare la corretta posizione dell'asse in maniera continua ed automatica, è necessario avere un feed-back sulla posizione; per questo motivo viene introdotto l'algoritmo di regolazione PID+FF comprendente azioni di tipo proporzionale, integrale, derivativo e feed-forward; il valore dell'uscita analogica è dato dalla sommatoria delle azioni feed forward, proporzionale, integrativa e derivativa.
Senza entrare nel merito di una descrizione tecnica della teoria della regolazione, in questo paragrafo vengono descritte una serie di operazioni per regolare i parametri che influenzano questo controllo.
Per realizzare una regolazione soddisfacente è sufficiente utilizzare solamente le azioni feedforward e proporzonale; le azioni integrale e derivativa vengono utilizzate solamente per regolazioni in condizioni particolari.



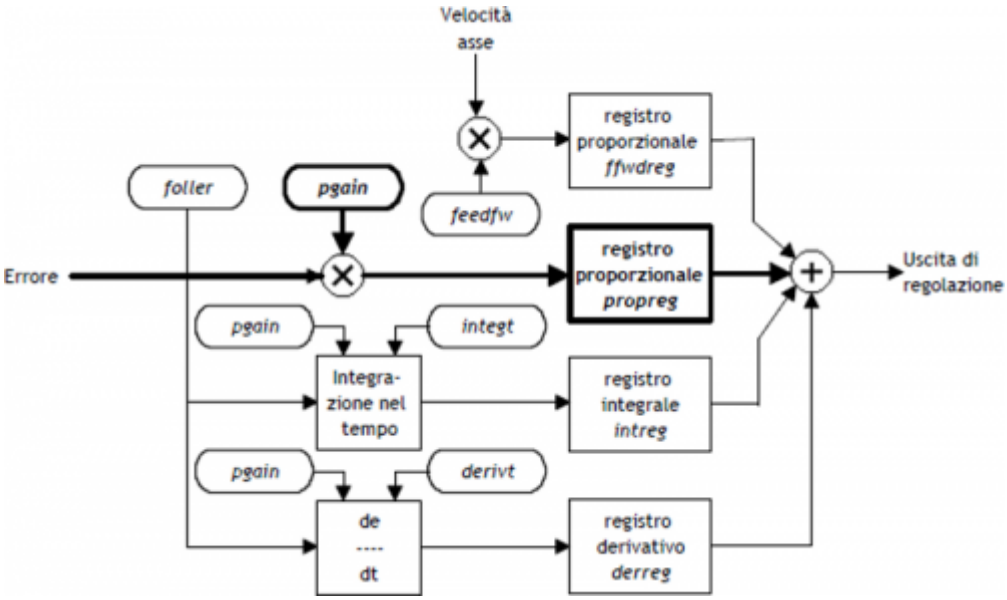
1.2.11.1 Azione feed forward

Il feed-forward contribuisce a rendere il sistema più pronto nei posizionamenti, fornendo all'uscita analogica un valore di tensione proporzionale alla velocitá teorica di posizionamento. In pratica é la componente grazie alla quale sono stati eseguiti i posizionamenti del capitolo precedente.
Può essere regolato il contributo di questa azione mediante il parametro *feedfw*; questo parametro é espresso come porzione millesimale della velocitá teorica; quindi, per introdurre ad esempio 98.5 % è necessario impostare 985 (millesimi).



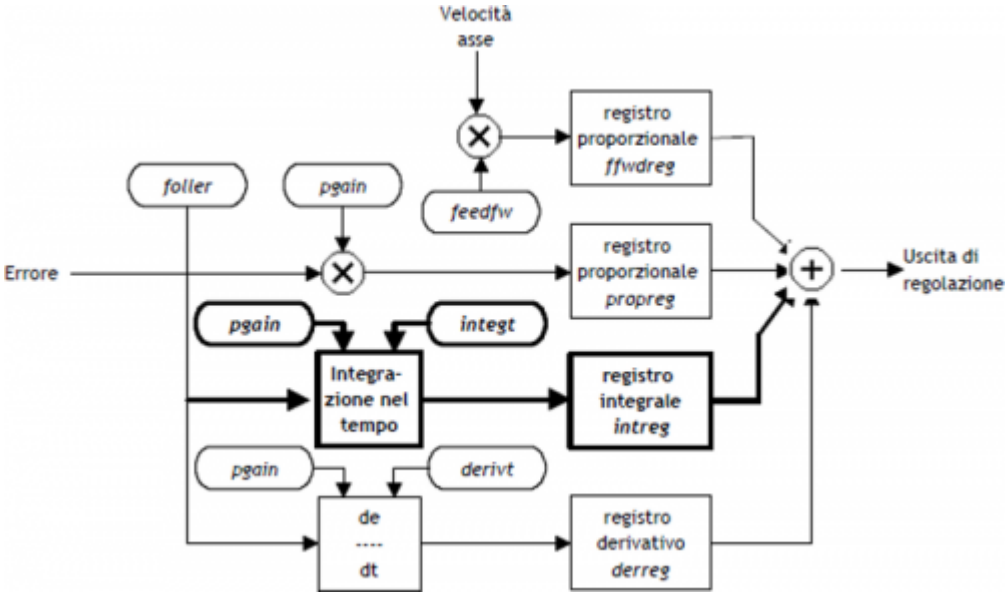
1.2.11.2 Azione proporzionale

Questa azione fornisce un'uscita proporzionale all'errore di posizione istantaneo dell'asse. L'entità dell'azione proporzionale é definita dal parametro *pgain* che definisce la sensibilità del sistema.
Il parametro pgain viene introdotto in millesimi; il valore unitario del guadagno (1000) fornisce un'uscita analogica al massimo valore (10 V) relativamente al massimo errore di velocitá. Per massimo errore di velocitá si intende lo spazio compiuto dall'asse - alla massima velocità - per la durata del tempo di campionamento del device.



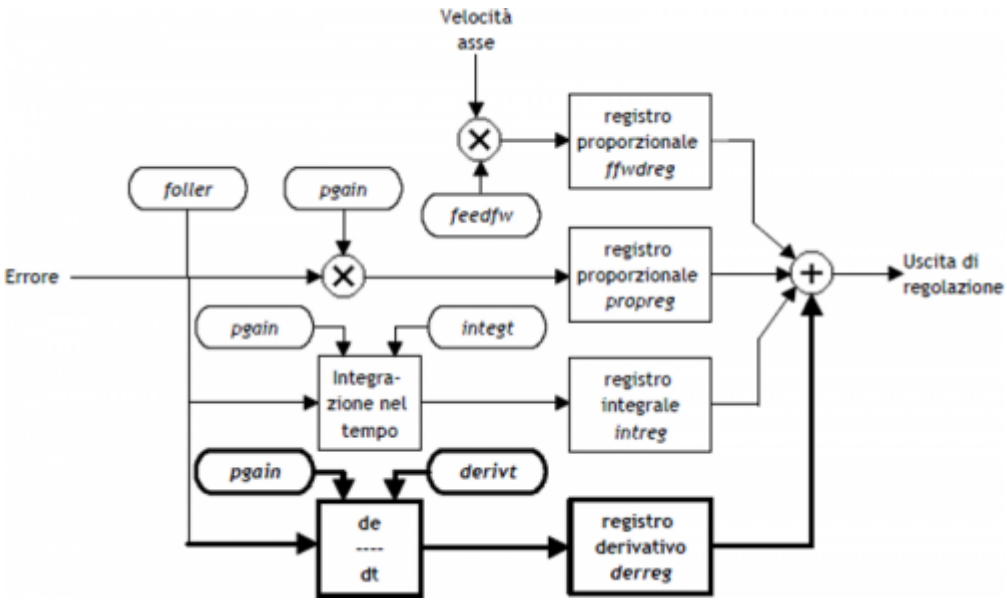
1.2.11.3 Azione integrale

Integra l'errore di posizione del sistema nel tempo impostato nel parametro *integt* aggiornando l'uscita finché l'errore non viene annullato.
Più basso è il tempo di integrazione dell'errore, più veloce è il sistema nel recupero dell'errore, ma il sistema può diventare instabile tendendo ad oscillare.



1.2.11.4 Azione derivativa

Anticipa la variazione del moto del sistema tendendo ad eliminare gli overshoot del posizionamento. L'entità della variazione viene calcolata nel tempo impostato nel parametro *derivt*.
Più alto è il tempo di derivazione dell'errore e più veloce è il sistema nel recupero dell'errore nei transitori, ma se viene inserito un valore troppo alto il sistema diventa instabile tendendo quindi ad oscillare.



1.2.12 Applicazione di movimentazione

Per poter muovere l'asse slave si deve innanzitutto dichiarare la parametrizzazione dell'asse. Una volta eseguita questa fase si ipotizza di voler far muovere l'asse slave con i jog manuali utilizzando gli ingressi Inp01 per movimentare l'asse in avanti e l'ingresso Inp02 per spostarlo indietro.
Come esempio consideriamo un device configurato come nello START UP. Nel task viene prima inizializzato il device e poi gestito il jog manuale.

```
;-----------------------------------------------------
; Gestione del jog manuale
;-----------------------------------------------------

INIT AsseX                          : Inizializza l'asse
WAIT AsseX:st_init                  ; Attendi che l'asse sia inizializzato
LOOPON AsseX                        ; Aggancia il loop di regolazione
WAIT AsseX:st_loopon                ; Attendi che l'asse abbia agganciato il
                                    ; loop di regolazione
CALOFF AsseX                        ; Esci dall'eventuale calibrazione
                                    ; dell'asse
WAIT NOT AsseX:st_cal               ; Attendi che il device non sia in
                                    ; calibrazione
CNTUNLOCK AsseX                     ; Sblocca il contatore master
WAIT NOT AsseX:st_cntlock           ; Attendi che il contatore master sia
                                    ; sbloccato
CNTDIR AsseX                        ; Imposta il giusto senso di incremento del
                                    ; contatore slave
WAIT NOT AsseX:st_cntrev            ; Attendi che il contatore slave sia
                                    ; impostato nel senso di incremento
CNTUNLOCKM AsseX                    ; Sblocca il contatore master
WAIT NOT AsseX:st_cntlockm          ; Attendi che il contatore master sia
                                    ; sbloccato
CNTDIRM AsseX                       ; Imposta il giusto senso di incremento del
                                    ; contatore master
WAIT NOT AsseX:st_cntrevm           ; Attendi che il contatore master sia
                                    ; impostato nel senso di incremento
REGON AsseX                         ; Sblocca la regolazione
WAIT NOT AsseX:st_regoff            ; Attendi lo sblocco della regolazione
MAIN:

IF Inp01 AND Inp02                  ; Se gli ingressi Inp01 e
                                    ; Inp02 sono attivi
  IF NOT AsseX:st_still             ; Se l'asse non è fermo
    STOP AsseX                      ; Ferma l'asse
  ENDIF
ENDIF

IF Inp01 AND NOT Inp02              ; Se l'ingresso Inp01 è
                                    ; attivo e l'ingresso
                                    ; Inp02 è disattivo
  IF AsseX:st_still                 ; Se l'asse è fermo
    AsseX:setvel=AsseX:maxvel/10    ; Imposto la velocità di
                                    ; movimento manuale
    MANFW AsseX                     ; Avanti manuale
  ENDIF
ELSE
  IF NOT Inp02                      ; Se l'ingresso Inp02
                                    ; è disattivo
    IF NOT AsseX:st_still           ; Se l'asse non è fermo
       STOP AsseX                   ; Ferma l'asse
    ENDIF
  ENDIF
ENDIF

IF Inp02 AND NOT Inp01              ; Se l'ingresso Inp02
                                    ; è attivo e l'ingresso
                                    ; Inp01 è disattivo
  IF AsseX:st_still                 ; Se l'asse è fermo
    AsseX:setvel=AsseX:maxvel/10    ; Imposto la velocità di
                                    ; movimento manuale
    MANBW AsseX                     ; Indietro manuale
  ENDIF
ELSE                                ; Altrimenti
  IF NOT Inp01                      ; Se l'ingresso Inp01 è
                                    ; disattivo
  IF NOT AsseX:st_still             ; Se l'asse non è fermo
     STOP AsseX                     ; Ferma l'asse
  ENDIF
ENDIF                               ; FINE
```

```
WAIT 1
JUMP MAIN
END
```

1.2.13 La struttura dei settori

Il device non ha al suo interno datagroup o array dati dove è possibile contenere vari tipi di camme, percui, se si devono gestire camme diverse in base al tipo di lavorazione, ci si deve appoggiare ai tool della CPU e scaricare i dati sul device ogni volta che ve ne è la necessità.

Esempio:
Con questo esempio viene gestita la programmazione della camma con i dati inseriti nel secondo programma di un datagroup.
Il device è configurato come descritto nello startup.

```
;-----------------------------------------------
; File di configurazione
;-----------------------------------------------

;-----------------------------------------------
; Variabili Globali
;-----------------------------------------------

GLOBAL
gfProgram F ;Abilitazione programmazione camma

;-----------------------------------------------
; Variabili System
;-----------------------------------------------

SYSTEM
sbPuntProg B ;Numero del programma da porre in esecuzione

;-----------------------------------------------
; Variabili Datagroup
;-----------------------------------------------

DATAGROUP
dgCamma

DATAPROGRAM 10 ;10 programmi disponibili
ddlCode L ;codice del programma

STEP 128 ;128 passi di programma disponibili
ddbCodeG B ;Codice G
ddlCodeQs L ;Codice Qs
ddlCodeQs L ;Codice Qm
ddlCodeM L ;Codice M
ddlCodeQma L ;Codice Qm ausiliario
ddlCodeQsa L ;Cocice Qs ausiliario

;-----------------------------------------------
; Task di programmazione camma
;-----------------------------------------------
MAIN:
   .
   .
   sbPuntProg = 2 ;Imposto il puntatore di programma
   .
   .

;-----------------------------------------------
; Programmazione del device CAMMING2
;-----------------------------------------------
IF gfProgram
   AsseX:codeG1     = ddbCodeG [sbPuntProg , 1] ;Settore 1
   AsseX:codeQm1    = ddlCodeQm [sbPuntProg , 1] ;Settore 1
   AsseX:codeQs1    = ddlCodeQs [sbPuntProg , 1] ;Settore 1
   AsseX:codeQma1   = ddlCodeQma [sbPuntProg , 1] ;Settore 1
   AsseX:codeQsa1   = ddlCodeQsa [sbPuntProg , 1] ;Settore 1
   AsseX:codeM1     = ddlCodeM [sbPuntProg , 1] ;Settore 1
   AsseX:codeG2     = ddbCodeG [sbPuntProg , 2] ;Settore 2
   AsseX:codeQm2    = ddlCodeQm [sbPuntProg , 2] ;Settore 2
   AsseX:codeQs2    = ddlCodeQs [sbPuntProg , 2] ;Settore 2
   AsseX:codeQma2   = ddlCodeQma [sbPuntProg , 2] ;Settore 2
   AsseX:codeQsa2   = ddlCodeQsa [sbPuntProg , 2] ;Settore 2
   AsseX:codeM2     = ddlCodeM [sbPuntProg , 2] ;Settore 2

   .
   .

   AsseX:codeG128 = ddbCodeG [sbPuntProg , 128] ;Settore 128
   AsseX:codeQm128 = ddlCodeQm [sbPuntProg , 128] ;Settore 128
   AsseX:codeQs128 = ddlCodeQs [sbPuntProg , 128] ;Settore 128
   AsseX:codeQma128 = ddlCodeQma [sbPuntProg , 128] ;Settore 128
   AsseX:codeQsa128 = ddlCodeQsa [sbPuntProg , 128] ;Settore 128
   AsseX:codeM128 = ddlCodeM [sbPuntProg , 128] ;Settore 128

   gfProgram = 0
ENDIF
```

1.3 I settori

Il device CAMMING2 gestisce dei settori di camma programmati in incrementale, all interno dei quali vengono riportati lo spazio da percorrere dal master e lo spazio che deve percorrere lo slave. Una camma è composta da più settori i quali possono essere di accelerazione, di decelerazione, di cambio velocità o dedicati ad operazioni particolari come, ad esempio, il rifasamento conteggi o loop camma.
Ogni settore della camma deve contenere delle informazioni relative a:

- codeG tipo di settore
- codeQm quota master (ATTENZIONE: inserire valori solo positivi)
- codeQs quota slave

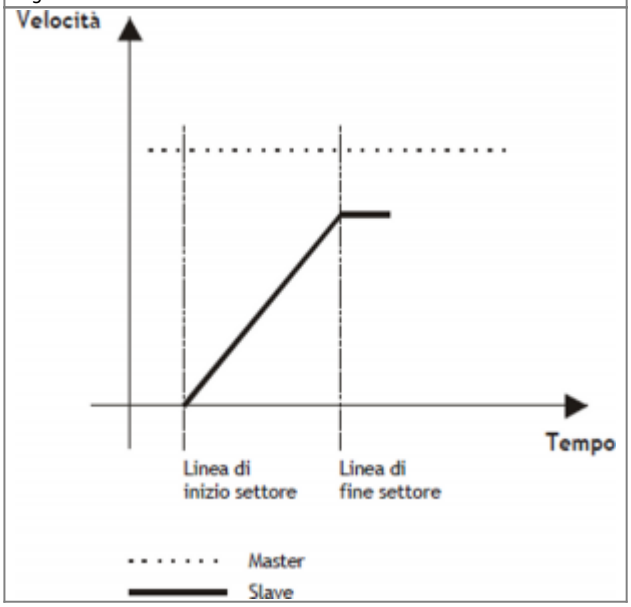
- codeQma quota master ausiliaria (ATTENZIONE: inserire valori solo positivi)
- codeQsa quota slave ausiliaria
- codeM codice di utilizzo generico, il quale viene visualizzato attraverso la variabile codeMex. In genere contiene lo stato degli utensili, gli stati particolari della camma, ecc.

1.3.1 Il settore di accelerazione

Il settore di accelerazione viene utilizzato con asse slave fermo (velocità slave uguale a zero, indipendentemente dalla velocità del master); alla fine del settore la velocità dello slave è uguale a quella del master.
I casi tipici di accelerazione sono riportati nelle figure A, B, C e D.
Nell esempio di figura A, alla fine del settore la velocità dello slave sarà uguale a quella del master; la legge che lega lo spazio master e quello slave è:
Spazio slave = 1/2 Spazio master
Più piccolo è lo spazio master che si considera e maggiore sarà il gradiente di accelerazione dello slave, il quale lo possiamo ricavare dalla formula:
Tempo acc. slave = Spazio master nel settore di acc. / Velocità massima master

Nel caso in cui ci si trovi di fronte a questo tipo di accelerazione si consiglia l'utilizzo del codice codeG = 132.

Figura A



Esempio di programmazione:

- codeG 132
- codeQm Spazio Master
- codeQs Spazio Slave
- codeQma Non utilizzato
- codeQsa Non utilizzato
- codeM codice generico

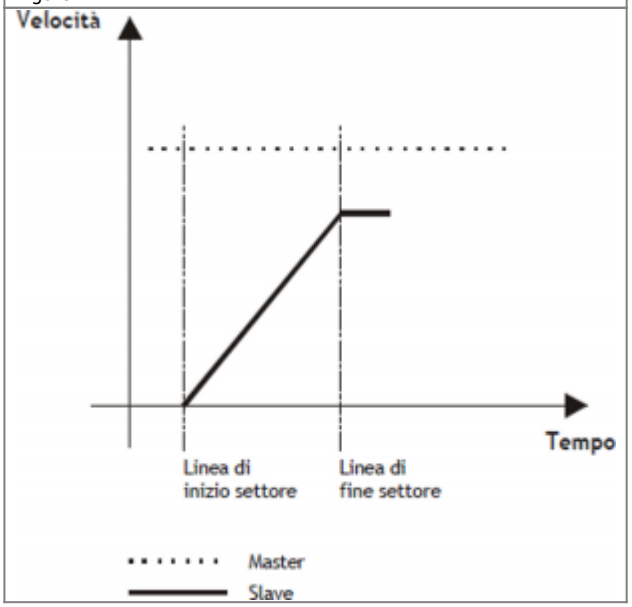
Nel caso si volessero utilizzare le rampe epicicloidali, si consiglia l'utilizzo del codice codeG = 232.

Nel caso in cui si vogliano utilizzare le rampe epicicloidali per accelerare rispettando lo stesso funzionamento descritto per il settore 132, è sufficiente programmare il settore come descritto sopra e programmando il *codeG* = 232.

Nell'esempio di figura B, alla fine del settore la velocità dello slave è in proporzione alla velocità del master (la proporzione verrà chiamata K), la legge che lega lo spazio master e lo spazio slave è:
Spazio slave = K/2 Spazio master
Più piccolo è lo spazio master che si considera e maggiore sarà il gradiente di accelerazione dello slave, il quale lo possiamo ricavare dalla formula:
Tempo di acc. slave = Spazio master nel settore di acc. / Velocità massima master

Nel caso in cui ci si trovi di fronte a questo tipo di accelerazione è obbligatorio l'utilizzo del codice codeG = 131.

Figura B



Esempio di programmazione:

- codeG 131
- codeQm Spazio Master
- codeQs Spazio Slave
- codeQma Non utilizzato
- codeQsa Non utilizzato
- codeM codice generico

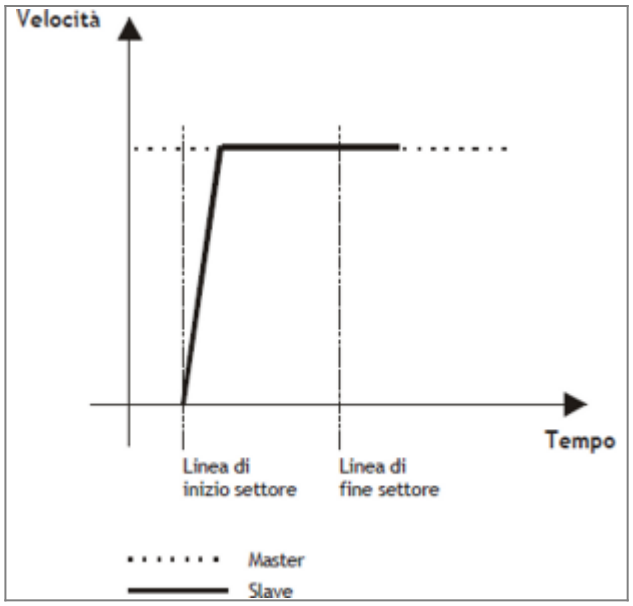
Nel caso si volessero utilizzare le rampe epicicloidali, si consiglia l'utilizzo del codice codeG = 231.

Nel caso in cui si vogliano utilizzare le rampe epicicloidali per accelerare rispettando lo stesso funzionamento descritto per il settore 131, è sufficiente programmare il settore come descritto sopra e programmando il *codeG* = 231.
Nell'esempio di figura C, si vogliono delle accelerazioni spinte, e non è possibile impostare delle quote Master/Slave di valore finito. Il settore 150 è in pratica la somma di due settori: 131 e 133. Tale settore è utilizzato quando si conoscono gli spazi successivi al settore di accelerazione e si vuole uno spazio slave accelerativo molto piccolo, anche inferiore all'unità di misura.
Il settore 150 si avvale dei seguenti parametri:

- codeG : codice settore (150)
- codeQma : indica lo spazio master entro il quale lo slave si deve portare a una certa velocità, che chiameremo di sincronizzazione.
- codeQm e codeQs : la cui divisione indica il rapporto tra lo spazio slave e master (rapporto di sincronizzazione) . Questi spazi saranno effettuati dopo la sezione accelerativa.
- **codeQsa : indica lo spazio in impulsi encoder che deve percorrere lo slave nella fase di accelerazione per raggiungere la velocità di sincronizzazione.**
  Più piccolo è lo spazio master che si considera e maggiore sarà il gradiente di accelerazione dello slave, il quale lo possiamo ricavare dalla formula:
  Tempo di acc. Slave = Spazio master nel settore di acc. / Velocità massima master

Nel caso in cui ci si trovi di fronte a questo tipo di accelerazione è obbligatorio l'utilizzo del codice codeG = 150.

Figura C



Esempio di programmazione:

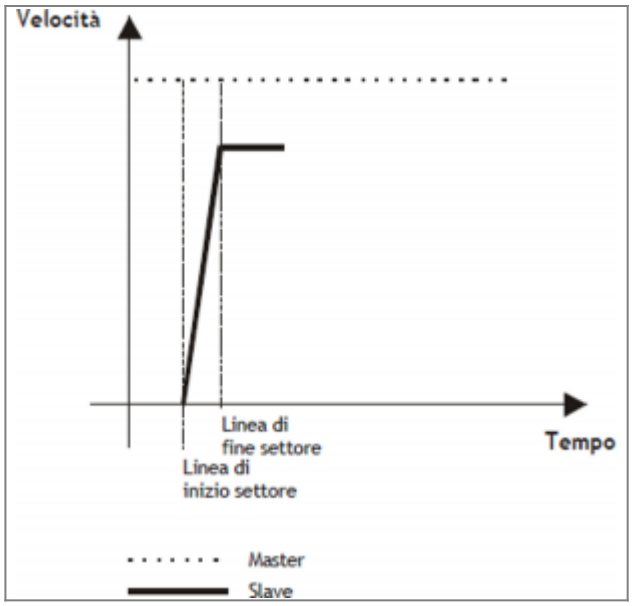
- codeG 150
- codeQm Spazio Master
- codeQs Spazio Slave
- codeQma Spazio Master in accelerazione
- codeQsa Spazio Slave in accelerazione (bit * 4)
- codeM codice generico

Nell'esempio di figura D, si vogliono delle accelerazioni spinte, e non è possibile impostare delle quote Master/Slave di valore finito. Il settore 152 è in pratica come il settore 131. Tale settore è utilizzato quando si conosce il rapporto di sincronizzazione e si vuole uno spazio slave accelerativo molto piccolo, anche inferiore all'unità di misura.
Il settore 152 si avvale dei seguenti parametri:

- codeG : codice settore (152)
- codeQma : indica lo spazio master entro il quale lo slave si deve portare a una certa velocità, che chiameremo di sincronizzazione.
- codeQm e codeQs : la cui divisione indica il rapporto tra lo spazio slave e master (rapporto di sincronizzazione).
- codeQsa : indica lo spazio in impulsi encoder che deve percorrere lo slave nella fase di accelerazione per raggiungere la velocità di sincronizzazione.
  Più piccolo è lo spazio master che si considera e maggiore sarà il gradiente di accelerazione dello slave, il quale lo possiamo ricavare dalla formula:
  Tempo di acc. Slave = Spazio master nel settore di acc. / Velocità massima master

Nel caso in cui ci si trovi di fronte a questo tipo di accelerazione è obbligatorio l'utilizzo del codice codeG = 152.

Figura D



Esempio di programmazione:

- codeG 152
- codeQm Coefficente Master
- codeQs Coefficente Slave
- codeQma Spazio Master in accelerazione
- codeQsa Spazio Slave in accelerazione (bit * 4)
- codeM codice generico

Nel caso si volessero utilizzare le rampe epicicloidali, si consiglia l'utilizzo del codice codeG = 252.

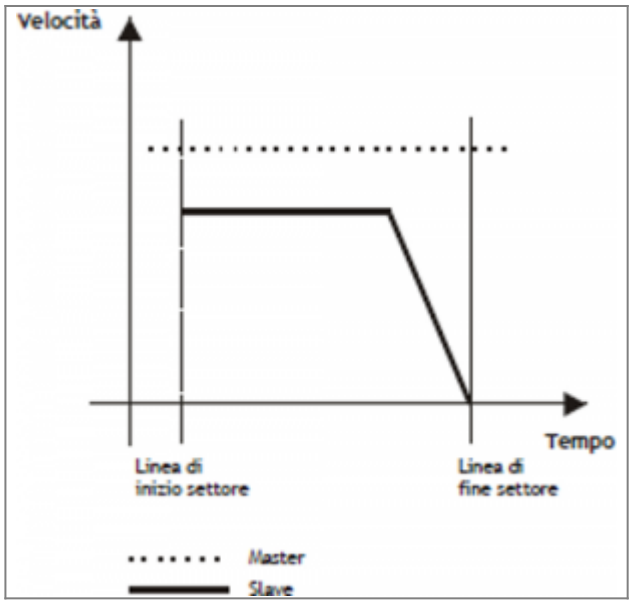
Nel caso in cui si vogliano utilizzare le rampe epicicloidali per accelerare rispettando lo stesso funzionamento descritto per il settore 152, è sufficiente programmare il settore come descritto sopra e programmando il *codeG* = 252.

1.3.2 Il settore di decelerazione

Nel caso in cui sia necessario fermare l'asse slave (indipendentemente dalla sua velocità), rimanendo agganciati con la camma (velocità zero indipendentemente dalla velocità del master), può essere utilizzato il settore di decelerazione.
Nell'esempio di figura E, alla fine del settore, la velocità dello slave sarà uguale a zero; la legge che lega lo spazio master e quello slave (la proporzione tra la velocità master e quella slave verrà chiamata K) è:
Spazio slave = K/2 Spazio master
Più piccolo è lo spazio master che si considera e maggiore sarà il gradiente di decelerazione dello slave, che è possibile ricavare da:
Tempo di dec. Slave = Spazio master nel settore di dec. / Velocità massima master

Nel caso in cui ci si trovi di fronte ad una decelerazione è obbligatorio l'utilizzo del codice codeG = 135.

Figura E



Esempio di programmazione

- codeG 135
- codeQm Spazio Master
- codeQs Spazio Slave
- codeQma Non utilizzato
- codeQsa Non utilizzato
- codeM codice generico

Nel caso si volessero utilizzare le rampe epicicloidali, si consiglia l'utilizzo del codice codeG = 235.

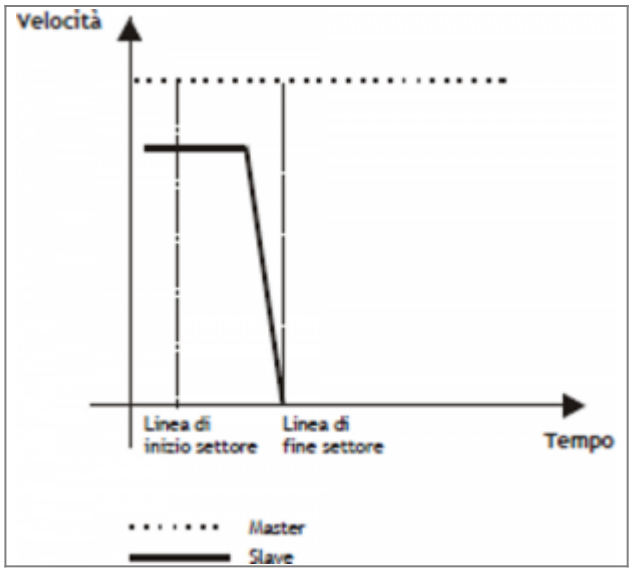
Nel caso in cui si vogliano utilizzare le rampe epicicloidali per accelerare rispettando lo stesso funzionamento descritto per il settore 135, è sufficiente programmare il settore come descritto sopra e programmando il *codeG* = 235.

Nell'esempio di figura F, si vogliono delle decelerazioni spinte, e non è possibile impostare delle quote Master/Slave di valore finito. Il settore 151 è in pratica la somma di due settori: 133 e 135. Tale settore è utilizzato quando si conoscono gli spazi precedenti al settore di decelerazione e si vuole uno spazio slave decelerativo molto piccolo, anche inferiore all'unità di misura. Il settore 151 si avvale dei seguenti parametri:

- codeG : codice settore (151)
- codeQma : indica lo spazio master entro il quale lo slave si deve portare da una certa velocità, che chiameremo di sincronizzazione a velocità zero.
- codeQm e codeQs : la cui divisione indica il rapporto tra lo spazio slave e master (rapporto di sincronizzazione) . Questi spazi sono effettuati prima della sezione decelerativa.
- codeQsa : indica lo spazio in impulsi encoder che deve percorrere lo slave nella fase di decelerazione.

Più piccolo è lo spazio master che si considera e maggiore sarà il gradiente di decelerazione dello slave, il quale lo possiamo ricavare dalla formula:
Tempo di dec. Slave = Spazio master nel settore di dec. / Velocità massima master

Figura F



Esempio di programmazione

- codeG 151
- codeQm Spazio Master
- codeQs Spazio Slave
- codeQma Spazio Master in decelerazione
- codeQsa Spazio Slave in decelerazione (bit * 4)
- codeM codice generico

1.3.3 Il settore di cambio velocità

Per poter effettuare queste operazioni esistono due tipi di codici (codeG = 133 e codeG = 134) i quali si differenziano solamente per la scelta della velocità che si vuole dare allo slave alla fine del settore di cambio velocità.

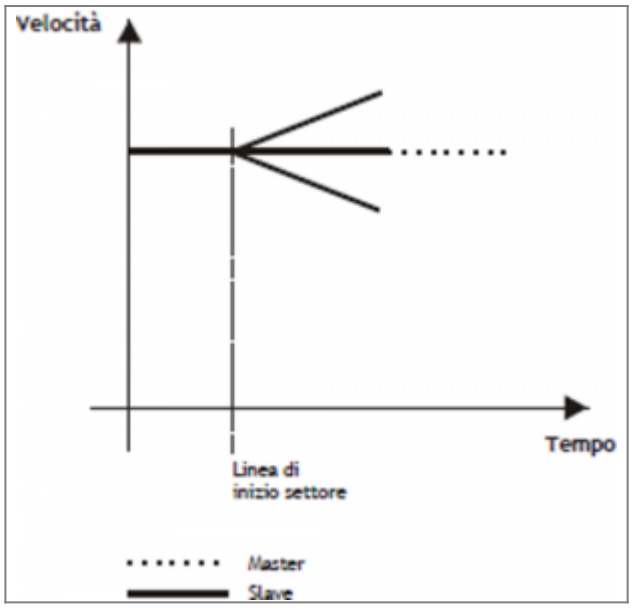
Il settore di cambio velocità può essere utilizzato:

- Ogni volta che l'asse slave deve raggiungere una velocità (diversa da zero), partendo da un diverso valore di velocità (anch'esso diverso da zero).
- Ogni volta che l'asse slave deve mantenere una velocità costante.

Nell'esempio la velocità dello slave è uguale a quella del master (all'inizio del settore di cambio velocità). Nel caso in cui la velocità sia diversa è necessario considerare, nelle formule a seguire, la costante del rapporto delle velocità master e slave all'inizio del settore.
Il codeG = 133 prevede che la velocità dello slave alla fine del settore possa essere diversa da quella iniziale e la velocità finale dello slave (di fine settore), dipenderà esclusivamente dal rapporto degli spazi master/slave (vedi figura G).
Ci si trova infatti di fronte a tre casi:
1)Rapporto master/slave < 1 Velocità dello slave a fine settore > della velocità del master
2)Rapporto master/slave = 1 Velocità dello slave a fine settore = della velocità del master
3)Rapporto master/slave > 1 Velocità dello slave a fine settore < della velocità del master

La velocità alla fine del settore sarà data dalla formula:
Vel. Slave = Vel. Master + { [2 (Spazio Slave - Spazio Master) / Spazio Master] x 100 } %

Figura G



Esempio di programmazione:

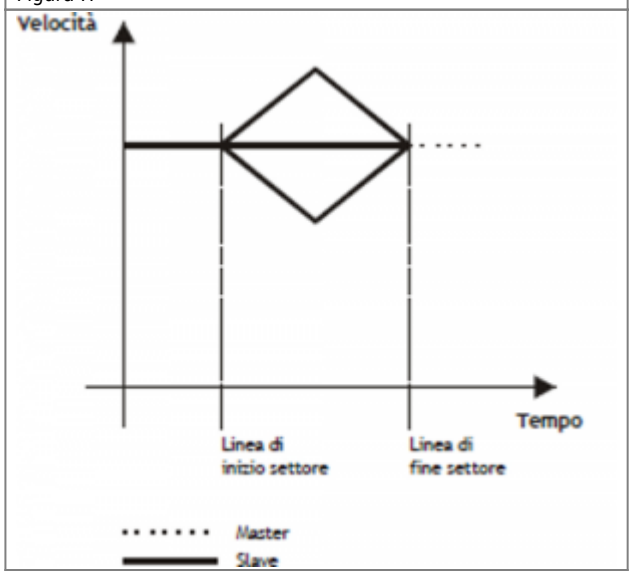
- codeG 133
- codeQm Spazio Master
- codeQs Spazio Slave
- codeQma Non utilizzato
- codeQsa Non utilizzato
- codeM codice generico

Nel caso si volessero utilizzare le rampe epicicloidali, si consiglia l'utilizzo del codice codeG = 233.

Nel caso in cui si vogliano utilizzare le rampe epicicloidali per accelerare rispettando lo stesso funzionamento descritto per il settore 133, è sufficiente programmare il settore come descritto sopra e programmando il *codeG* = 233.
Il codeG = 134 prevede che la velocità dello slave alla fine del settore sia uguale a quella iniziale e la velocità a metà settore dello slave dipenderà esclusivamente dal rapporto degli spazi master/ slave (vedi figura H). Ci si trova infatti di fronte a tre casi:
1 Rapporto master/slave < 1 Velocità dello slave al centro del settore > della velocità del master
2 Rapporto master/slave = 1 Velocità dello slave al centro del settore = della velocità del master
3 Rapporto master/slave > 1 Velocità dello slave al centro del settore < della velocità del master

La velocità al centro del settore sarà data dalla formula:
Vel. slave = Vel. master + { [2 (Spazio slave - Spazio master) / Spazio master] x 100 } % x (Vel. master)

Figura H



Esempio di programmazione:

- codeG 134
- codeQm Spazio Master
- codeQs Spazio Slave
- codeQma Non utilizzato
- codeQsa Non utilizzato
- codeM codice generico

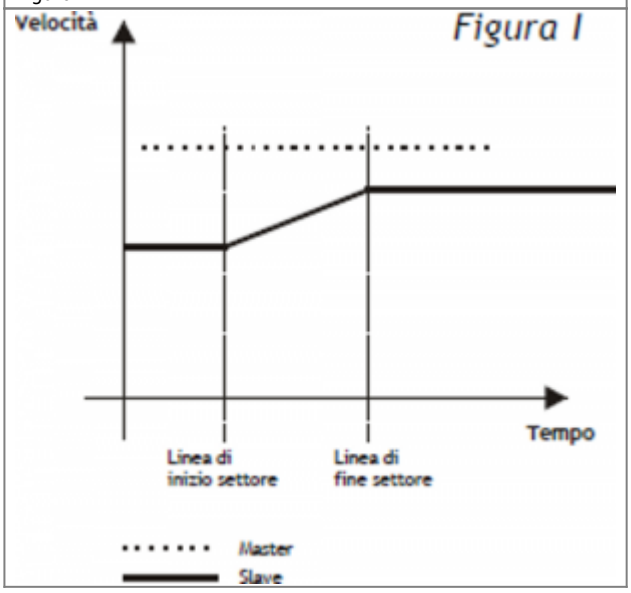
Nel caso si volessero utilizzare le rampe epicicloidali, si consiglia l'utilizzo del codice codeG = 234.

Nel caso in cui si vogliano utilizzare le rampe epicicloidali per accelerare rispettando lo stesso funzionamento descritto per il settore 134, è sufficiente programmare il settore come descritto sopra e programmando il *codeG* = 234.
Se viene programmato un settore 133, 134, 233 o 234 con spazio master e slave a 0, viene considerato come un settore non operativo (codeG = 130).
Nell'esempio di figura I, si vuole cambiare velocità allo slave, e non è possibile impostare un rapporto Master/Slave di valore finito. Il settore 153 è in pratica come il settore 133. Tale settore è utilizzato quando si conosce il rapporto di sincronizzazione e si vuole uno spazio slave accelerativo molto piccolo, a volte anche inferiore all'unitá di misura.
Il settore 153 si avvale dei seguenti parametri:

- codeG : codice settore (153)
- codeQma : indica lo spazio master entro il quale lo slave si deve portare a una certa velocità, che chiameremo di sincronizzazione.
- codeQm e codeQs : la cui divisione indica il rapporto tra lo spazio slave e master (rapporto di sincronizzazione).
- codeQsa : il device indica lo spazio in impulsi encoder che ha percorso lo slave per raggiungere la velocità di sincronizzazione dopo la fase di accelerazione.

Nel caso in cui ci si trovi di fronte a questo tipo di cambio velocità, è consigliato l'utilizzo del codice codeG = 153.

Figura I



Esempio di programmazione:

- codeG 153
- codeQm Coefficente Master
- codeQs Coefficente Slave
- codeQma Spazio Master in accelerazione
- codeQsa Spazio Slave in accelerazione (bit * 4)
- codeM codice generico

Nel caso si volessero utilizzare le rampe epicicloidali, si consiglia l'utilizzo del codice codeG = 253.

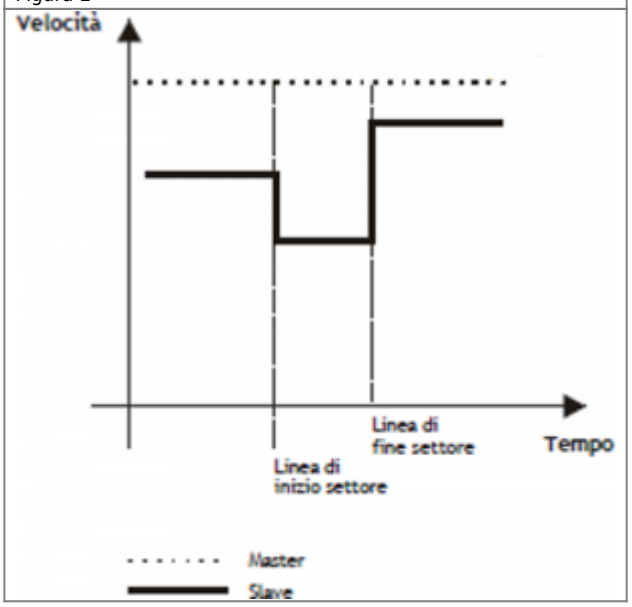
Nel caso in cui si vogliano utilizzare le rampe epicicloidali per accelerare rispettando lo stesso funzionamento descritto per il settore 153, è sufficiente programmare il settore come descritto sopra e programmando il *codeG* = 253.

Nell'esempio di figura L, si vuole portare lo slave ad una velocità senza dovere eseguire una rampa di raccordo. Il settore 154 a differenza di tutti gli altri, impone la velocità iniziale uguale alla velocità finale mantenendo la velocità costante tra i due punti. Questo settore può essere utilizzato come settore di partenza della camma (partenza senza accelerazione), come settore intermedio oppure come ultimo settore (fermata senza rampa).
Il settore 154 si avvale dei seguenti parametri:

- codeG : codice settore (154)
- codeQma : Tipo di addolcimento settore
- codeQm e codeQs : la cui divisione indica il rapporto tra lo spazio slave e master (rapporto di sincronizzazione). Questi spazi vengono eseguiti durante il settore.
- codeQsa : Se impostato a 0 indica che il settore successivo è un settore di movimento, se viene impostato a 1 indica che il settore successivo non prevede il movimento (decelerazione con rampa zero).

Nel caso in cui ci si trovi di fronte a questo tipo di movimento è obbligatorio l'utilizzo del codice codeG = 154.

Figura L



Esempio di programmazione:

- codeG 154
- codeQm Spazio Master
- codeQs Spazio Slave
- codeQma Tipo di addolcimento settore
- codeQsa
  0 = settore sucessivo di movimento
  1 = asse fermo nel settore successivo
  2 = Albero elettrico
- codeM codice generico

1.3.4 Il settore di Start sincronizzato al Master

Molte volte esiste la necessità di far partire lo slave su un punto del master noto, ma non esiste la possibilità di collegarsi ad un sensore di prossimità. L'unico vincolo è che il settore contenente il codeG 160 deve essere il primo settore di movimento della camma e non può essere messo in ciclo. Al comando di STARTCAM, lo stato st_camex va a 1 ed Il movimento dell'asse Slave inizia solo al superamento della quota Master (espressa in unità di misura) impostata nel settore 160 e da li seguirà l'andamento descritto nei settori successivi.
Se lo STARTCAM é dato con il conteggio master superiore alla quota impostata, viene settato il warning 9; in queste condizioni il conteggio Master deve divenire minore della quota impostata per poter trovarsi nella giusta situazione di partenza del sistema.
Non è possibile entrare in un settore con codeG = 160 provenendo da un jump o da un loop camma (errore 7).

Esempio di programmazione:

- codeG 160
- codeQm Quota di STARTCAMMA espressa in unità di misura
- codeQs Non utilizzato
- codeQma Non utilizzato
- codeQsa Non utilizzato
- codeM Non utilizzato

1.3.5 Il settore di fine camma

Il settore di cambio fine camma (codeG = 136), viene utilizzato ogni volta che si deve concludere la camma (sganciare la camma) fermando l'asse slave in reazione di spazio sull'ultimo punto della camma. Naturalmente l'asse slave deve essere fermo al momento dello sgancio della camma, percui si presume che il settore precedente contenga il codice di decelerazione (codeG = 135).
Dopo aver eseguito questo settore la camma è sganciata e, per riagganciarla, bisogna inviare il comando di *STARTCAM*.

Esempio di programmazione:

- codeG 136
- codeQm Non utilizzato
- codeQs Non utilizzato
- codeQma Non utilizzato
- codeQsa Non utilizzato
- codeM Non utilizzato

1.3.6 Il settore di absolute jump

Il settore di absolute jump (codeG = 137), viene utilizzato per fare un salto ad un settore (definito nel codeQm) per poter modificare al volo l'andamento della camma in base a delle condizioni stabilite dal programmatore.
La situazione più comune per l'utilizzo di questa funzione è quella di una parte della camma che deve essere ripetuta parecchie volte.
Bisogna fare attenzione al fatto che i conteggi non vengono aggiornati e quindi a lungo andare possono andare in overflow. Si consiglia quindi di utilizzare i settori di aggiornamento conteggio nel settore che precede quello contenente il codeG = 137.

Esempio di programmazione:

- codeG 137
- codeQm Numero del settore a cui saltare
- codeQs Non utilizzato
- codeQma Non utilizzato
- codeQsa Non utilizzato
- codeM Non utilizzato

1.3.7 Il settore di jump condizionato

Il settore di jump condizionato (codeG = 190), viene utilizzato per fare un salto ad un settore (definito nel codeQm) per un certo numero di volte (definito nel codeQs) dopo di che si passa al settore successivo. Il conteggio del numero di salti eseguiti è disponibile nel codeQma.
Bisogna fare attenzione al fatto che i conteggi non vengono aggiornati e quindi a lungo andare possono andare in overflow. Si consiglia quindi di utilizzare i settori di aggiornamento conteggio nel settore che precede quello contenente il codeG = 190.

Esempio di programmazione:

- codeG 190
- codeQm Numero del settore a cui saltare
- codeQs Numero di volte
- codeQma Visualizzazione numero salti effettuati
- codeQsa Non utilizzato
- codeM Non utilizzato

1.3.8 Il settore di loop camma

Il settore di loop camma (codeG = 138), viene utilizzato per ripetere la camma in esecuzione dal settore numero uno, azzerando per sottrazione sia i conteggi master che slave.
Si consiglia l'utilizzo di questo codice nelle camme ripetute all'infinito che non hanno problemi di sottrazione dei conteggi.

Esempio di programmazione:

- codeG 138
- codeQm Non utilizzato
- codeQs Non utilizzato
- codeQma Non utilizzato
- codeQsa Non utilizzato
- codeM Non utilizzato

1.3.9 Il settore non operativo

Il settore non operativo (codeG = 130), viene utilizzato per riservare dei settori a delle funzioni da eseguire solamente in condizioni particolari definite dal programmatore.
Per esempio si può considerare una camma per il taglio al volo, nella quale è necessario riservare dei settori da utilizzare nel caso in cui, meccanicamente, non si riesca a fare il taglio nello spazio master riservato a tale operazione.

Esempio di programmazione:

- codeG 130
- codeQm Non utilizzato
- codeQs Non utilizzato
- codeQma Non utilizzato
- codeQsa Non utilizzato
- codeM Non utilizzato

1.3.10 Definizione di settori a campionamento zero

Tutti i settori che non necessitano spazio master per essere processati sono definiti “a campionamento zero”; nello specifico sono tutti i settori di NOP, JUMP, LOOP ed END.
Un settore a campionamento zero è considerato anche il codeG = 133 se programmato come:

- codeG = 133
- codeQm = 0
- codeQs = 0

Per come è strutturato il device, non è possibile mettere in sequenza più di 9 settori a campionamento zero.

1.3.11 I settori di aggiornamento conteggio

Il settore di aggiornamento conteggio si utilizza per fare un cambio del conteggio, portandolo a valori che possano indicare la reale posizione fisica dell'asse. Il caso più tipico è l'asse circolare (da 0° a 360°): ogni volta che si raggiungono i 360° si deve sottrarre un angolo giro. Per fare un aggiornamento conteggio esistono molteplici codici di sottrazione o di impostazione conteggio, sia in bit encoder che in unità di misura. Per come è strutturato il device, non è possibile mettere in sequenza più di 4 settori di aggiornamento conteggio. Si riporta a seguito una tabella contenente la descrizione delle operazioni eseguite durante l'aggiornamento conteggio in base al codice utilizzato.

codeG Operazioni eseguite
139 Sottrazione dal conteggio master del valore contenuto in codeQm (espresso in unità di misura).
Sottrazione dal conteggio slave del valore contenuto in codeQs (espresso in unità di misura).
140 Forzatura del conteggio master al valore contenuto in codeQm (espresso in unità di misura).
141 Forzatura del conteggio slave al valore contenuto in codeQs (espresso in unità di misura).
142 Forzatura del conteggio master al valore contenuto in codeQm (espresso in unità di misura).
Forzatura del conteggio slave al valore contenuto in codeQs (espresso in unità di misura).
143 Sottrazione del conteggio master del valore contenuto in codeQm)espresso in bit encoder moltiplicati per 4).
Sottrazione del conteggio slave del valore contenuto in codeQs (espresso in bit encoder moltiplicati per 4).
144 Forzatura del conteggio master al valore contenuto in codeQm (espresso in bit encoder moltiplicati per 4).
145 Forzatura del conteggio slave al valore contenuto in codeQs (espresso in bit encoder moltiplicati per 4).
146 Forzatura del conteggio master al valore contenuto in codeQm (espresso in bit encoder moltiplicati per 4).

Forzatura del conteggio slave al valore contenuto in codeQs (espresso in bit encoder moltiplicati per 4).

1.3.12 Descrizione settori camma

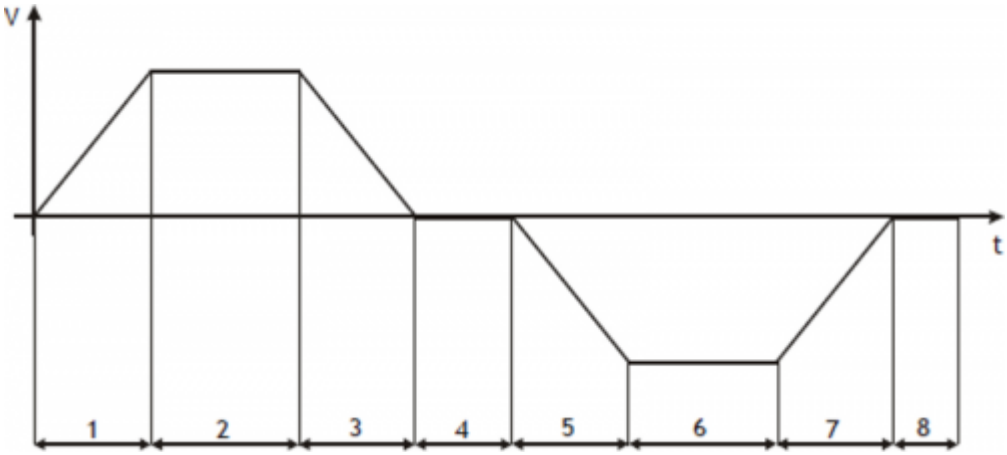
CodeG	codeQm	codeQs	codeQma	codeQsa	codeM	Descrizione
130	n.u.	n.u.	n.u.	n.u.	n.u.	NOP: Settore disabilitato (non operativo)
131	Incremento Master (Um)	Incremento Slave (Um)	n.u.	n.u.	c.u.	AZL: Settore d'accelerazione con velocità iniziale zero e velocità finale calcolata in funzione dello spazio slave da percorrere. Velocità finale slave = f (spazio slave).
132	Incremento Master (Um)	Incremento Slave (Um)	n.u.	n.u.	c.u.	AZM: Settore d'accelerazione con velocità iniziale zero e velocità finale pari a quella del master (velocità finale slave = velocità master), variando il gradiente d'accelerazione.
133	Incremento Master (Um)	Incremento Slave (Um)	n.u.	n.u.	c.u.	RSC: Settore intermedio (raccordo senza compensazione) con velocità iniziale uguale alla velocità finale del settore precedente e velocità finale calcolata in funzione dello spazio slave da percorrere.
134	Incremento Master (Um)	Incremento Slave (Um)	n.u.	n.u.	c.u.	RCC: Settore intermedio (raccordo con compensazione) con velocità iniziale e finale uguale alla velocità finale del settore precedente: questo viene ottenuto eseguendo una compensazione dello spazio slave, dividendo in due fasi (accelerazione e decelerazione) l'esecuzione del settore.
135	Incremento Master (Um)	Incremento Slave (Um)	n.u.	n.u.	c.u.	DZC: Settore di decelerazione con velocità iniziale uguale alla velocità finale del settore precedente e velocità finale uguale a zero: questo viene ottenuto eseguendo anche una compensazione dello spazio slave, dividendo in due fasi l'esecuzione dei settore.
136	n.u.	n.u.	n.u.	n.u.	n.u.	END: Fine camma. Il sistema sgancia la camma in esecuzione e rimane in reazione di spazio con lo slave sull'ultima posizione elaborata dal settore precedente a questo.
137	Numero del settore al quale saltare	n.u.	n.u.	n.u.	n.u.	ABJ: Absolute jump. Il sistema mantiene la posizione e la velocità dell'ultimo settore processato. I conteggi non variano. Il numero della camma a cui si salta va indicato in codeQm e deve essere compreso tra 1 e 128.
138	n.u.	n.u.	n.u.	n.u.	n.u.	LOOP: Loop camma. Quando viene incontrata quest'istruzione viene ripresa l'elaborazione dei settori a partire dal primo, mantenendo come velocità quella dell'ultimo settore processato e sottraendo il conteggio della quantità di spazio eseguita fino a quel momento.
139	Valore di sottrazione conteggio Master (Um)	Valore di sottrazione conteggio Slave (Um)	n.u.	n.u.	n.u.	SMS: Sottrai conteggi in unità di misura. Viene sottratto al conteggio del Master il valore contenuto in codeQm ed al conteggio dello Slave il valore contenuto in codeQs (sottrazione conteggio Master e Slave in unità di misura)
140	Nuovo conteggio Master (Um)	n.u.	n.u.	n.u.	n.u.	NCM: Cambia conteggio master. Viene scritto il valore contenuto in codeQm nel conteggio del Master. L'aggiornamento del conteggio viene eseguito per sottrazione (aggiorna il conteggio Master in unita di misura)
141	n.u.	Nuovo conteggio Slave (Um)	n.u.	n.u.	n.u.	NCS: Cambia conteggio Slave. Viene scritto il valore contenuto in codeQs nel conteggio dello Slave. L'aggiornamento del conteggio viene eseguito per sottrazione (aggiorna il conteggio Slave in unita di misura)
142	Nuovo conteggio Master (Um)	Nuovo conteggio Slave (Um)	n.u.	n.u.	n.u.	NMS: Cambia conteggi. Vengono scritti i conteggi Master e Slave con i valori contenuti rispettivamente in codeQm e codeQs (aggiorna i conteggi master e slave in unita di misura).
143	Valore di sottrazione conteggio Master (bit*4)	Valore di sottrazione conteggio Slave (bit*4)	n.u.	n.u.	n.u.	SBMS: Sottrai conteggi Master e Slave in bit. Viene sottratto al conteggio del Master il valore contenuto in codeQm ed al conteggio dello Slave il valore contenuto in codeQs (sottrazione conteggio master e slave in bit x 4).
144	Nuovo conteggio Master (bit*4)	n.u.	n.u.	n.u.	n.u.	NBM: Cambia conteggio Master in bit. Quest'istruzione scrive il valore contenuto in codeQm nel conteggio del Master. L'aggiornamento del conteggio viene eseguito per sottrazione (aggiorna il conteggio master in bit x 4).
145	n.u.	Nuovo conteggio Slave (bit*4)	n.u.	n.u.	n.u.	NBS: Cambia conteggio Slave in bit. Quest'istruzione scrive il valore contenuto in codeQs nel conteggio dello Slave. L'aggiornamento del conteggio viene eseguito per sottrazione (aggiorna il conteggio slave in bit x 4).
146	Nuovo conteggio Master (bit*4)	Nuovo conteggio Slave (bit*4)	n.u.	n.u.	n.u.	NBMS: Cambia conteggi Master e Slave in bit. Questa istruzione aggiorna i conteggi Master e Slave con i valori contenuti rispettivamente in codeQm e codeQs (aggiorna i conteggi master e slave in bit x 4)

CodeG	codeQm	codeQs	codeQma	codeQsa	codeM	Descrizione
150	Incremento Master (Um)	Incremento Slave (Um)	Spazio Master in accelerazione (Um)	Spazio Slave in accelerazione (bit*4)	c.u.	AZMC: Settore d'accelerazione con velocità iniziale zero e velocità finale calcolata in funzione dello spazio Master e Slave indicato in codeQm e codeQs. L'accelerazione viene eseguita nello spazio indicato in codeQma e codeQsa. Vengono eseguiti gli spazi indicati in codeQm e codeQs con la legge descritta nel codeG 133.
151	Incremento Master (Um)	Incremento Slave (Um)	Spazio Master in decelerazione (Um)	Spazio Slave in decelerazione (bit*4)	c.u.	DZMC: Settore di decelerazione con velocità iniziale uguale alla velocità finale del settore precedente e velocità finale uguale a zero. La decelerazione viene eseguita nello spazio indicato in codeQma e codeQsa. Vengono eseguiti gli spazi indicati in codeQm e codeQs con la legge descritta nel codeG 133.
152	Coefficiente Master	Coefficiente Slave	Spazio Master in accelerazione (Um)	Spazio Slave in accelerazione (bit*4)	c.u.	AZMS: Settore d'accelerazione con velocità iniziale zero e velocità finale calcolata in funzione dei coefficienti Master e Slave indicati in codeQm e codeQs. L'accelerazione viene eseguita nello spazio indicato in codeQma e codeQsa. Non vengono eseguiti gli spazi indicati in codeQm e codeQs
153	Coefficiente Master	Coefficiente Slave	Spazio Master in cambio velocità (Um)	Spazio Slave in cambio velocità (bit*4)	c.u.	NVSR: Cambio velocità in rampa: l'asse Slave passa dalla velocità attuale alla velocità calcolata in funzione dei coefficienti Master e Slave indicati in codeQm e codeQs. Il cambio di velocità viene eseguito nello spazio indicato in codeQma e codeQsa. Non vengono eseguiti gli spazi indicati in codeQm e codeQs
154	Incremento Master (Um)	Incremento Slave (Um)	Tipo di addolcimento	Tipo di settore	c.u.	NVS: Cambio di velocità senza rampa. L'asse Slave passa dalla velocità attuale alla velocità calcolata in funzione degli spazi master e slave indicati in codeQm e codeQs senza rampa (esegue un gradino). Nel codeQsa viene indicato se si tratta dell'ultimo settore (impostando 1 si indica che al successivo settore l'asse slave è fermo) oppure se il movimento continua (impostando 0 si indica che al successivo settore l'asse slave è di movimento) impostando a 2 il codiceQsa si può utilizzare l'asse come ALBERO ELETTRICO. Una volta impostato il codeQm e il codeQs in modo da ottenere il rapporto di velocità MASTER/SLAVE. Il nuovo R.V. viene ottenuto senza rampa quindi se si vuole una variazione graduale bisogna variare gradualmente il codeQs. Modificando il codeQs (riportandolo a 0 o a 1) si passa al settore successivo (nel caso non si sia programmato nessun settore successivo per fermare il device è sufficiente dare uno Stopcam altrimenti si incorre in un errore). N.B. Durante quest'ultima funzionalità i parametri posit e positm perdono il significato dato che rimangono fissi ad un valore corrispondente a metà degli spazi programmati in codeQm e codeQs.
160	Quota Master (Um)	n.u.	n.u.	n.u.	c.u.	STS: Start sincronizzato Allo STARTCAM si attende che l'asse Master superi la quota indicata in codeQm per passare al settore successivo. I settori precedenti a questo non devono essere di movimento e questo codice non può essere messo in loop camma.

1.3.13 Basi per la costruzione di una camma per spandifilo

Come esempio consideriamo un semplice spandifilo:

- Partenza con rampa di accelerazione.
- Raggiungimento di una velocità proporzionale a quella del master.
- Mantenimento della velocità raggiunta per tutto il percorso.
- Fermata con rampa di decelerazione.
- Stop l asse per un certo spazio del master.
- Ritorno al punto di partenza con le stesse modalità del tratto di andata.



Settore 1 Acceleazione, con partenza da velocità zero e spostamento slave positivo (codeG = 131). È importante calcolare il rapporto dello spazio master/slave di questo tratto in modo che la velocità di uscita sia quella che poi verrà mantenuta dall asse slave nel tratto a velocità costante.

Settore 2 Intermedio con velocità costante e spostamento slave positivo (codeG = 133).

Settore 3 Decelerazione con velocità finale zero, con una possibile compensazione della velocità di frenata nella prima metà del tratto e spostamento slave positivo (codeG = 135). Potrebbe avere gli stessi valori impostati nel settore 1.

Settore 4 Fermata lavorazione con spostamento slave uguale a zero (codeG = 133). Si programma lo spazio master mentre quello slave viene impostato a 0.

Settore 5 Accelerazione, con partenza da velocità zero e spostamento slave negativo (codeG = 131). È importante calcolare il rapporto dello spazio master/slave di questo tratto in modo che la velocità di uscita sia quella che poi verrà mantenuta dall asse slave nel tratto a velocità costante. Teoricamente si possono impostare gli stessi valori inseriti nel settore 1 cambiando di segno la quota slave.

Settore 6 Intermedio con velocità costante e spostamento slave negativo (codeG = 133).

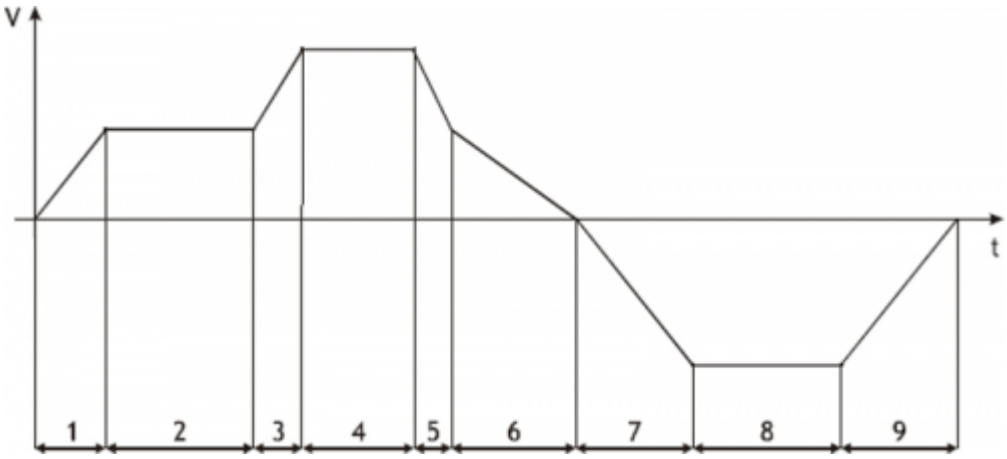
Settore 7 Decelerazione con velocità finale zero, con una possibile compensazione della velocità di frenata nella prima metà del tratto e spostamento slave negativo (codeG = 135). Potrebbe avere gli stessi valori impostati nel settore 5.

Settore 8 Fermata lavorazione con spostamento slave uguale a zero (codeG = 133). Si programma lo spazio master mentre quello slave viene impostato a 0. Dopo avere eseguito il settore 8, ci dovranno essere delle funzioni che eseguono il rifasamento dei conteggi Master e Slave sottraendo lo spazio percorso fino a fine settore; successivamente si dovrà avere la riesecuzione automatica della stessa camma dal settore 1 (JUMP o loop camma).

1.3.14 Basi per la costruzione di una camma per taglio al volo con extravelocità

Come esempio consideriamo un semplice taglio al volo:

- Partenza asse slave con rampa di accelerazione.
- Raggiungimento della velocità master.
- Mantenimento della velocità raggiunta per tutto il taglio.
- Concluso il taglio l asse slave deve accelerare per portarsi ad una extravelocità, mantenendola per un certo spazio.
- Stop asse slave con rampa di decelerazione.
- Ritorno dell asse slave al punto di partenza (home), senza tempo di inversione ed eseguendo le rampe di accelerazione e decelerazione.



Settore 1 Accelerazione, con partenza da velocità zero e spostamento slave positivo (codeG = 132). Alla fine di questo settore lo slave avrà la stessa velocità del master.

Settore 2 Intermedio con velocità costante e spostamento slave positivo (codeG = 133). In questo settore lo spazio percorso dal master sarà uguale a quello percorso dallo slave.

Settore 3 Accelerazione e spostamento slave positivo (codeG = 133). Il codice impostato non è di accelerazione ma, per far accelerare lo slave rispetto al master, viene impostato uno spazio slave maggiore di quello master.

Settore 4 Intermedio con velocità costante e spostamento slave positivo (codeG = 133). In questo settore lo spazio percorso dallo slave sarà proporzione a quello percorso dal master.

Settore 5 Decelerazione e spostamento slave positivo (codeG = 133). In questo settore si porta lo slave alla stessa velocità del master.

Settore 6 Decelerazione con velocità finale zero, con una possibile compensazione della velocità di frenata nella prima metà del tratto e spostamento slave positivo (codeG = 135).

Settore 7 Accelerazione, con partenza da velocità zero e spostamento slave negativo (codeG = 131). In questo settore la velocità di uscita dello slave può essere diversa da quella del master.

Settore 8 Intermedio con velocità costante e spostamento slave negativo (codeG = 133).

Settore 9 Decelerazione con velocità finale zero, con una possibile compensazione della velocità di frenata nella prima metà del tratto e spostamento slave negativo (codeG = 135).

Dopo avere eseguito il settore 9, ci dovrà essere una funzione che esegua il rifasamento del conteggio del Master, sottraendo lo spazio percorso fino a fine settore e, successivamente, la riesecuzione automatica della stessa camma (JUMP o loop camma).

1.4 Gestione errori device

La presenza di un errore nel sistema camming viene segnalato dallo stato *st_error*.
Essendo causato da un evento grave e non essendo garantita in questa situazione la gestione dell'asse slave, si è deciso in modo arbitrario di bloccare l'asse senza rampe come fosse avvenuta un'emergenza.
Quando *st_error* è uguale a 1, troviamo presente sulla variabile *errcode* il tipo di errore intervenuto (vedi tabella) e nella variabile *errvalue* una indicazione sulla causa dell'errore.

Codice	Priorità	Descrizione
1	0	Troppi settori a campionamento nullo consecutivi
2	0	JUMP da un settore con velocità finale diversa da zero su un settore con velocità iniziale uguale a zero (codice di accelerazione).
3	0	Codice G del settore non valido.
4	0	Spazio master del settore camma troppo piccolo, quindi il settore non è calcolato.
5	0	Tentato di scrivere nel settore in esecuzione.
6	0	Nel codice di JUMP, è stato richiesto di andare ad una riga non compresa tra 1 e 128.
7	0	Settore con codeG = 160 non eseguito all'inizio della camma.

Se il device va in errore, per poter riprendere la lavorazione bisogna cancellare lo stato *st_error* attraverso il comando *RSERR* e fare la consueta routine di ripristino da emergenza (*RESUME* asse).

NOTA: L'errore 4 è dovuto al fatto che il settore viene eseguito in un tempo inferiore al tempo di campionamento del device, percui non può essere processato. Se ci si trova in questa situazione bisogna aumentare la quota del master nel settore oppure calare la velocità del master.

1.5 Gestione warning device

La presenza di un warning nel sistema camming viene segnalato dallo stato *st_warning*.
Essendo causato da un evento non grave ed essendo garantita in questa situazione la gestione dell'asse slave, l'asse slave continua il suo lavoro.
Quando *st_warning* è uguale a 1, troviamo presente sulla variabile *wrncode* il tipo di warning intervenuto (vedi tabella) e nella variabile *wrnvalue* il numero del settore della camma che ha provocato il warning.

Codice	Priorità	Descrizione
1	6	Costante di accelerazione settore maggiore di quella programmata.
2	7	Costante di decelerazione settore maggiore di quella programmata.
3	4	Saturazione dell'analogica slave a + 10V (con autoritenuta)
4	5	Saturazione dell'analogica slave a - 10V (con autoritenuta)
5	9	Velocità finale di segno opposto a quella iniziale.
6	2	Incontrato un settore di accelerazione quando la camma proviene da un settore con velocità finale diversa da zero.
7	8	Velocità intermedia di segno opposto a quella iniziale.
8	0	Evento catturato da ingresso in interrupt ma non elaborato immediatamente per sovraccarico nei calcoli del device.
9	1	Quota di partenza asse Slave con codeG = 160 già superata

La priorità più alta è contrassegnata da 0, la più bassa con 8.
Per cancellare lo stato st_warning bisogna inviare il comando RSWRN.\ NOTA: In caso di warning 8, la funzione sará ritardata per un tempo sufficiente da consentire alla CPU di terminare dei calcoli interni. Nel caso di start camma da ingresso di interruzione, la posizione di avvio camma può non essere quella del momento dell'interrupt, ma quella dopo la fine dei calcoli.
Il tempo di esecuzione dei calcoli (espressi in tempo di campionamento del device), è riportata nella tabella seguente:

Parametri che comportano ricalcoli	N.ro campionamenti in cui sono distribuiti i conseguenti ricalcoli
codeG, codeQs, codeQm,codeQsa, codeQma, maxpos, minpos,prspos, prsposm,toll, tacc, tdec, taccmax, tdecmax, syncrange, pgain, feedfw, integt, derivt	1
tbfm	2
tbf	3
maxvel	5
decpt, unitvel	6
pulsem, measurem	42
pulse, measure	51

1.6 Gestione master simulato

L'encoder master del device CAMMING2 non è in alcun modo legato all'encoder del device EANPOS.

Il device CAMMING2 può gestire due tipi di master:

- Entrambi possono essere provenienti da un encoder meccanicamente collegato al sistema master ed elettricamente collegato al sistema QMOVE oppure encoder simulati. Viene inoltre accettata la soluzione mista (uno collegato elettricamente ed uno simulato.) Lo scambio tra i due encoder viene fatto attraverso il parametro mtype senza nessun vincolo, in modo che, anche nell'esecuzione di una camma, sia possibile fare lo scambio tra i dispositivi. Nel sistema utilizzante il device CAMMING2 può essere dichiarato un encoder simulato utilizzando un device di movimento (ad esempio un EANPOS) dichiarato con il contatore sullo slot 1 (normalmente riservato alla CPU del sistema) e tutte le altre periferiche disabilitate:

```
;-----------------------------
; Dichiarazione device interni
;-----------------------------

INTDEVICE
<nome_device>   EANPOS   TCamp   ICont   IntL   IAZero   IOutA
Master          EANPOS   2       1.CNT01 X      X.X      X.X
```

dove:

<nome device>	Nome assegnato al device.
EANPOS	Parola chiave che identifica il device posizionatore analogico.
TCamp	Tempo di campionamento device (1÷255 ms).
ICont	Ingresso contatore bidirezionale.
IntL	Numero della linea di interrupt dedicata per l'impulso di zero dell'encoder durante la fase di ricerca di preset.
IAZero	Ingresso di abilitazione per l'acquisizione dell'impulso di zero del trasduttore durante la fase di ricerca di preset.
IOutA	Indirizzo hardware del componente DAC dell'uscita analogica (obbligatoriamente dichiarata come X.X).

Il device così configurato viene considerato come un master simulato e viene parametrizzato e utilizzato come fosse un device normale tenendo presente che il loop di regolazione deve essere aperto (*st_loopon* = 0) e di conseguenza non serve parametrizzare il P.I.D. ma è sufficiente impostare il feedforward al 100% (*feedfw* = 1000).

1.6.1 Esempio di programmazione

Si ipotizza di utilizzare il device EANPOS configurato come nell esempio precedente e di voler dare il set di velocità (setvel) espresso in Hz. Si ipotizza inoltre che il master simulato debba continuare il suo movimento all infinito.
Il flag sf01 esegue lo start e lo stop del device simulato.

```
;-----------------------------------------------------
; Gestione del master simulato
;-----------------------------------------------------

Master:measure = 1000
Master:pulse = 4000
Master:decpt = 0
Master:unitvel = 1
Master:maxvel = 1000
Master:taccdec = 100
Master:maxpos = 999999
Master:minpos = -999999
INIT Master
WAIT Master:st_init
LOOPOFF Master
WAIT NOT Master:st_loopon
RESUME Master
WAIT NOT Master:st_emrg

MAIN:
IF sf01
  IF Master: st_still
    Master:posit = 0
    Master:setvel = 500
    Master:setpos = 999999
    START Master
  ENDIF
  IF Master:posit GE 500000
    Master:posit = 0
  ENDIF
ELSE
  IF NOT Master:st_still
    STOP Master
  ENDIF
ENDIF

WAIT 1
JUMP MAIN
END
```

1.7 Limitazione rapporto frequenze trasduttore M/S

Per avere un corretto funzionamento durante la fase di sincronismo, si richiede che gli impulsi nel tempo (frequenza) generati dal trasduttore Master siano maggiori o uguali a quelli dell asse Slave. In ogni caso si richiede di rispettare la condizione

Frequenza slave = 1,5 × Frequenza master

Nel caso di un non rispetto di questa condizione si hanno dei problemi nella taratura dell asse Slave in sincronismo a causa di una rugosità nel movimento.

1.8 Tabella di configurazione ingressi

Il device ha la possibilità di gestire un ingresso normale ed un ingresso in interrupt per eseguire comandi o eseguire azioni. L indirizzo degli ingressi è configurabile nel file di configurazione (InG ed InGInt). Per far eseguire una funzione specifica all ingresso, è sufficiente assegnare alla variabile funInp (se si tratta di ingresso normale) oppure funInt (se si tratta di ingresso in interrupt) il codice riportato nella tabella seguente.

Codice	Funzione ingresso
00	Ingresso disabilitato
01	STOPCAM
02	STARTCAM
03	Scrive il valore della variabile encoder nella variabile delta1
04	Scrive il valore della variabile encoderm nella variabile delta2
05	Incrementa di 1 la variabile delta1
06	Incrementa di 1 la variabile delta2
07	Scrive il contenuto della variabile delta1 in encoder
08	Scrive il contenuto della variabile delta2 in encoderm
09	Scrive il valore della variabile encoder nella variabile delta1 + STARTCAM

Tutte le funzioni degli ingressi possono essere gestiti sia su ingressi normali che su ingressi in interrupt.
Per avere un corretto funzionamento degli ingressi, bisogna che essi siano attivati rispettando le condizioni richieste nella descrizione del comando o dell azione descritta.

1.9 Tabella di configurazione uscite

Il device ha la possibilità di gestire un uscita per segnalare alcuni stati. L'indirizzo dell'uscita è configurabile nel file di configurazione (Out). Per far eseguire una funzione specifica all'uscita, è sufficiente assegnare alla variabile funOut il codice riportato nella tabella seguente.

Codice	Funzione uscita
00	Uscita disabilitata
01	Disattivazione uscita
02	Attivazione uscita
03	st_toll
04	st_tpos
05	st_sync
06	Si attiva l'uscita solamente se codeMex é uguale al valore 1000
07	Si attiva l'uscita solamente se codeMex é uguale al valore 1000 e st_sync é attivo
08	Si attiva l'uscita solamente se codeMex é uguale al valore 1001
09	Si attiva l'uscita solamente se codeMex é uguale al valore 1002

1.10 Tabella comandi, stati e parametri: Simbologia adottata

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

=**R**\\=

Indica se il relativo parametro o stato è ritentivo (al momento dell'inizializzazione del device mantiene lo stato precedentemente definito), oppure lo stato che assume al momento dell'inizializzazione del device.
R = Ritentivo
0 = Al momento dell'inizializzazione del device il valore viene forzato a zero.
1 = Al momento dell'inizializzazione del device il valore viene forzato a uno.

=**D**\\=

Indica la dimensione del parametro.
F = Flag
B = Byte
W = Word
L = Long

1.10.1 Condizioni

Vengono descritte tutte le condizioni necessarie affinché il parametro sia considerato corretto o perché il comando venga accettato.
In alcuni casi vengono specificati dei valori limite per l'accettazione del parametro: se vengono introdotti dei valori esterni ai limiti impostati, il dato viene comunque accettato; pertanto devono essere previsti opportuni controlli interni tali da garantire il corretto funzionamento. Per l'esecuzione di un comando, tutte le relative condizioni devono necessariamente essere soddisfatte; in caso contrario il comando non viene eseguito.

A

Indica la modalità di accesso.
R = Read (lettura).
W = Write (scrittura).

1.10.2 Parametri

Nome	D	Condiz. scritt.	R	A	Descrizione
decpt	B	st_still = 1 st_camex = 0 st_prson = 0	R	RdWr	**Decimal point** (0÷3) Definisce la precisione con la quale si intendono impostare le preselezioni e visualizzare i conteggi relativamente all'asse slave.
measure	L	st_still = 1 st_camex = 0 st_prson = 0	R	RdWr	**Measure** (1÷999999) Indica lo spazio, in unità di misura, percorso dall'asse slave per ottenere gli impulsi encoder impostati nel parametro *pulse*. Questo parametro è utilizzato per il calcolo della risoluzione dell'asse con la formula: Risoluzione = *measure** 4 / *pulse* La risoluzione deve avere un valore compreso tra 0.00374 e 4.00000

Nome	D	Condiz. scritt.	R	A	Descrizione
pulse	L	st_still = 1 st_camex = 0 st_prson = 0	R	RdWr	**Pulse encoder** (1÷999999) Indica gli impulsi moltiplicato 4 forniti dall'encoder slave per ottenere lo spazio impostato nel parametro *measure*. Questo parametro è utilizzato per il calcolo della risoluzione dell'asse con la formula: Risoluzione = *measure** 4 / *pulse* La risoluzione deve avere un valore compreso tra 0.00374 e 4.00000
measurem	L	st_still = 1 st_camex = 0 st_prson = 0	R	RdWr	**Measure of master** (1÷999999) Indica lo spazio, in unità di misura, percorso dall'asse master per ottenere gli impulsi encoder impostati nel parametro *pulsem*. Questo parametro è utilizzato per il calcolo della risoluzione dell'asse con la formula: Risoluzione = *measurem* * 4 / *pulsem* La risoluzione deve avere un valore compreso tra 0.00374 e 4.00000
pulsem	L	st_still = 1 st_camex = 0 st_prson = 0	R	RdWr	**Pulse encoder of master** (1÷999999) Indica gli impulsi moltiplicato 4 forniti dall'encoder master per ottenere lo spazio impostato nel parametro *measurem*. Questo parametro è utilizzato per il calcolo della risoluzione dell'asse con la formula: Risoluzione = *measurem** 4 / *pulsem* La risoluzione deve avere un valore compreso tra 0.00374 e 4.00000
unitvel	B	st_still = 1 st_camex = 0 st_prson = 0	R	RdWr	**Velocity unit** (0÷1) Definisce se l'unità di tempo della velocità dello slave è espressa in minuti o secondi. 0 = Um/min 1 = Um/sec
maxvel	L	st_still = 1 st_camex = 0 st_prson = 0	R	RdWr	**Max velocity** (0÷999999) Definisce la massima velocità dell'asse slave (relativa al riferimento analogico di +/-10V). Il valore introdotto è nell'unità di tempo della velocità impostata nel parametro unitvel.
prsvel	L	st_prson = 0	R	RdWr	**Preset velocity** (0÷maxvel) Definisce la velocità dell'asse slave durante la procedura di ricerca di preset. Il valore introdotto è nell'unità di tempo della velocità impostata nel parametro unitvel.
sprsvel	L	st_prson = 0	R	RdWr	**Preset velocity** (0÷maxvel) Definisce la velocità dell'asse slave durante la procedura di ricerca di preset. Il valore introdotto è nell'unità di tempo della velocità impostata nel parametro unitvel.
taccmax	W	st_prson = 0	R	RdWr	**Search preset velocity** (0÷prsvel) Nella procedura di ricerca di preset dello slave, definisce la velocità dell'asse nella fase di acquisizione dell'impulso di zero. Il valore introdotto è nell'unità di tempo della velocità impostata nel parametro unitvel.
tdecmax	W	st_prson = 0	R	RdWr	**Max deceleration time** (0÷999) Usato durante l'esecuzione della camma per eseguire le comparazioni sul gradiente di decelerazione massimo. Definisce il tempo minimo di decelerazione con cui l'asse slave può portarsi da velocità massima ad asse fermo (velocità uguale a zero). Il valore introdotto è espresso in centesimi di secondo.
tacc	W	Nessuna condizione	R	RdWr	**Acceleration time** (0÷999) Definisce il tempo impiegato dall'asse slave per portarsi da fermo alla velocità massima. Il valore introdotto è espresso in centesimi di secondo. Se l'asse si sta muovendo (st_still = 0) si possono cambiare i gradienti della rampa solamente se i nuovi valori consentono di raggiungere la quota impostata.
tdec	W	Nessuna condizione	R	RdWr	**Deceleration time** (0÷999) Definisce il tempo necessario all'asse slave per decelerare dalla velocità massima a zero (condizione di asse fermo). Il valore introdotto è espresso in centesimi di secondo. Se l'asse si sta muovendo (st_still = 0) si possono cambiare i gradienti della rampa solamente se i nuovi valori consentono di raggiungere la quota impostata.
maxpos	L	st_still = 0	R	RdWr	**Max position** (-999999÷999999) Definisce la massima quota raggiungibile dall'asse slave. Tale limite non è controllato durante l'esecuzione della camma.
minpos	L	st_still = 0	R	RdWr	**Min position** (-999999÷999999) Definisce la minima quota raggiungibile dall'asse slave. Tale limite non è controllato durante l'esecuzione della camma.
prspos	L	st_still = 0	R	RdWr	**Preset position** (minpos÷maxpos) Definisce il valore che viene caricato sul conteggio slave con la procedura di ricerca di preset.

Nome	D	Condiz. scritt.	R	A	Descrizione
prsposm	L	st_prsonm = 0	R	RdWr	**Preset position of master** (-999999 ÷999999) Definisce il valore che viene caricato sul conteggio master con la procedura di ricerca di preset.
toll	L	st_still = 0	R	RdWr	**Tolerance** (0÷999999) Definisce una fascia di conteggio intorno alle quote di posizionamento dell asse slave. Se il posizionamento (non l arrivo in camma) si conclude entro tale fascia, è da considerarsi corretto e viene segnalato attraverso lo stato st_toll.
maxfollerr	L	Nessuna condizione	R	RdWr	**Maximum following error** (0÷2 31-1) Definisce il massimo scostamento accettabile tra la posizione teorica e la posizione reale dell asse slave. Il valore introdotto è espresso in bit trasduttore per 4.
syncrange	L	Nessuna condizione	R	RdWr	**Synchronism range** (0÷999999) È il valore espresso in unità di misura entro il quale viene segnalato il sincronismo slave (st_sync = 1) rispetto al master durante l esecuzione della camma.
prsmode	B	st_prson = 0	R	RdWr	**Preset mode** (0÷2) Definisce il tipo di ricerca di preset dello slave: **0** = Per la ricerca dell abilitazione impulso di zero, l asse inizia il movimento in veloce, incontra il segnale di abilitazione, inverte la direzione rallentando e, sul fronte di discesa relativo al segnale di abilitazione dell asse slave, carica la quota di preset. **1** = Per la ricerca dell abilitazione impulso di zero, l asse inizia il movimento in veloce, incontra il segnale di abilitazione, inverte la direzione ed in lento acquisisce il primo impulso di zero (dopo la disattivazione del segnale di abilitazione dell asse slave). **2** =Non viene attivata la procedura di ricerca preset (st_prson = 0). Il conteggio viene aggiornato alla quota di preset all attivazione dell abilitazione impulso di zero dell asse slave.
prsmodem	B	st_prsonm = 0	R	RdWr	**Preset mode of master** (0÷2) Definisce il tipo di ricerca di preset del master: **0** = Se st_prsonm = 1, il conteggio viene aggiornato alla quota di preset alla disattivazione dell abilitazione impulso di zero dell asse master. **1** = Se st_prsonm = 1, il conteggio viene aggiornato alla quota di preset all attivazione dell impulso di zero dopo la disattivazione dell abilitazione impulso di zero dell asse master. **2** =Non viene attivata la procedura di ricerca preset (st_prsonm = 0). Il conteggio viene aggiornato alla quota di preset all attivazione dell abilitazione impulso di zero dell asse master.
prsdir	B	st_prson = 0	R	RdWr	**Preset search direction** (0÷1) Definisce la direzione del movimento asse per la ricerca del finecorsa di abilitazione impulso di zero dell asse slave. **0** = l asse si dirige in avanti. **1** = l asse si dirige indietro.
mtype	B	Nessuna condizione	R	RdWr	**Master type** (0÷1) Definisce l indirizzo del master utilizzato: **0** = Il master è l encoder avente indirizzo “A”. **1** = Il master è l encoder avente indirizzo “B”. (Vedi capitolo “Gestione master simulato”)
ramptype	B	st_still = 0	R	RdWr	**Ramp type of slave** (0÷1) Definisce il tipo di rampe dello slave utilizzate nei normali posizionamenti; nell esecuzione della camma i raccordi saranno sempre eseguiti con rampe trapezoidali: **0** = rampe trapezoidali. **1** = rampe epicicloidali. (Vedi capitolo “Descrizione movimento trapezoidale”)
rtype	B	Nessuna condizione	R	RdWr	**Riduction profile type** (0÷1) Definisce il tipo di riduzione del profilo di posizionamento dell asse slave se sono state selezionate le rampe di tipo epicicloidale (ramptype = 1). **0** = I tempi di accelerazione e di decelerazione rimangono quelli d e l l a velocità impostata e viene diminuita proporzionalmente la velocità. **1** = Vengono diminuiti i tempi di accelerazione e di decelerazione (mantenendo il gradiente di accelerazione e di decelerazione impostato) e anche la velocità stessa. (Vedi capitolo “Descrizione movimento trapezoidale”)
stopt	B	Nessuna condizione	R	RdWr	**Stop type** (0÷1) Definisce il tipo di frenata che viene utilizzata in caso di stop posizionamento dell asse slave se sono state selezionate le rampe di tipo epicicloidale (ramptype = 1). **0** = Quando si esegue una frenata in rampa viene prima completata le rampa di accelerazione e poi viene eseguita la rampa di decelerazione. **1** = Quando viene eseguita una frenata in rampa viene immediatamente eseguita la rampa di decelerazione. (Vedi capitolo “Descrizione movimento trapezoidale”)

Nome	D	Condiz. scritt.	R	A	Descrizione
pgain	W	Nessuna condizione	R	RdWr	**Proportional gain** (0÷32767) Impostando il valore 1000, il coefficente è 1.000 È il coefficente che moltiplicato per l'errore di inseguimento genera la parte proporzionale dell'uscita di regolazione dell'asse slave. (Vedi capitolo dedicato)
feedfw	W	Nessuna condizione	R	RdWr	**Feed forward** (0÷32767) Impostando il valore 1000, la percentuale è del 100%. È il coefficente percentuale che, moltiplicato per la velocità istantanea, genera la parte feed-forward dell'uscita di regolazione dell'asse slave. (Vedi capitolo dedicato)
integt	W	Nessuna condizione	R	RdWr	**Integral time** (0÷32767) È il tempo, espresso in millisecondi, che produce il coefficente di integrazione dell'errore di inseguimento. L'integrazione di tale errore moltiplicata per tale coefficente genera la parte integrale dell'uscita di regolazione dell'asse slave. (Vedi capitolo dedicato)
derivt	W	Nessuna condizione	R	RdWr	**Derivation time** (0÷32767) È il tempo, espresso in millisecondi, che produce il coefficente derivativo dell'errore di inseguimento. La derivazione di tale errore moltiplicata per tale coefficente genera la parte integrale dell'uscita di regolazione dell'asse slave. (Vedi capitolo dedicato)
offset	W	Nessuna condizione	R	RdWr	**Offset output** (-32767÷32767) Offset uscita DAC asse slave espressa in bit. Definisce il valore in bit della correzione relativa all'uscita analogica dell'asse slave in modo da compensare l'eventuale deriva del sistema.
tbfm	W	Nessuna condizione	R	RdWr	**Time base frequency-meter master** (0÷3) Definisce il tempo di campionamento del frequenzimetro relativo all'asse master. **0** = 240 ms **1** = 480 ms **2** = 24 ms **3** = 120 ms N.B. Minore è il tempo di campionamento, più veloce è l'acquisizione della frequenza, ma maggiore è l'errore alle basse frequenze.
tbf	W	Nessuna condizione	R	RdWr	**Time base frequency-meter slave** (0÷3) Definisce il tempo di campionamento del frequenzimetro relativo all'asse slave. **0** = 240 ms **1** = 480 ms **2** = 24 ms **3** = 120 ms N.B. Minore è il tempo di campionamento, più veloce è l'acquisizione della frequenza, ma maggiore è l'errore alle basse frequenze.

1.10.3 Variabili asse

Nome	D	Condiz. scritt.	R	A	Descrizione
frqm	L	Nessuna condizione	0	RdWr	**Actual frequency of master** Indica la frequenza del trasduttore relativo all'asse master. Per modificare la precisione riferirsi al parametro tbfm. I valore è espresso in Hz
positm	L	st_init = 1 st_camex = 0	R	RdWr	**Actual position of master** (-999999 ÷ +999999) Indica la posizione attuale dell'asse master. Il valore è espresso in unità di misura.
encoderm	L	st_init = 1 st_camex = 0	R	RdWr	**Encoder value of master** Indica la posizione attuale dell'asse master. Il valore è espresso in bit encoder per 4.
vout	B	st_init = 1 st_cal = 1	0	RdWr	**Output voltage** (-100÷100) Impostando il valore 100, la percentuale è del 100%. Consente l'impostazione o la visualizzazione (in questo caso senza nessuna condizione) della tensione di uscita relativa all'uscita analogica dell'asse slave. Il dato è espresso in decimi di Volt.
follerr	L	Nessuna condizione	0	Rd	**Following error** Indica l'errore tra la posizione teorica e la posizione reale dell'asse slave in valore assoluto. Il valore è espresso in bit trasduttore per 4.
vel	L	Nessuna condizione	0	Rd	**Actual velocity** Indica la velocità attuale dell'asse slave. Il valore letto è espresso nell'unità di tempo della velocità impostata (Velocity unit).

Nome	D	Condiz. scritt.	R	A	Descrizione
frq	L	Nessuna condizione	0	Rd	**Actual frequency** Indica la frequenza del trasduttore relativo all'asse slave. Il valore letto è espresso in Hz
posit	L	st_init = 1 st_camex = 0	R	RdWr	**Actual position** (-999999 ÷ +999999) Indica la posizione attuale dell'asse slave. Il valore introdotto o letto è espresso in unità di misura.
encoder	L	st_init = 1 st_camex = 0	R	RdWr	**Encoder value** ($-2^{31} \div 2^{31}-1$) Indica la posizione attuale dell'asse slave. Il valore letto è espresso in bit trasduttore per 4.
delta1	L	Nessuna condizione	R	RdWr	**Delta 1** ($-2^{31} \div 2^{31}-1$) Variabile d'uso generico. Utilizzata come registro per scambio dati.
delta2	L	Nessuna condizione	R	RdWr	**Delta 2** ($-2^{31} \div 2^{31}-1$) Variabile d'uso generico. Utilizzata come registro per scambio dati.
setvel	L	Nessuna condizione	R	RdWr	**Set velocity** (0÷maxvel) Definisce la velocità dell'asse slave nei posizionamenti. Il valore introdotto è nell'unità di tempo della velocità impostata (Velocity unit). Se l'asse si sta muovendo (st_still = 0) si può cambiare il setpoint di velocità solamente se il nuovo valore consente di raggiungere la quota impostata.
setpos	L	Nessuna condizione	R	RdWr	**Set position** (minpos÷maxpos) Definisce la quota di posizionamento raggiungibile dall'asse slave alla velocità setvel.
rowex	W	Nessuna condizione	0	Rd	**Row in use** (0÷40) Definisce il numero del settore in secuzione.
ffwdreg	L	Nessuna condizione	0	Rd	**Feed-forward register** ($-2^{31} \div 2^{31}-1$) È il valore istantaneo del registro di feed-forward espresso in bit.
propreg	L	Nessuna condizione	0	Rd	**Proportional register** ($-2^{31} \div 2^{31}-1$) È il valore istantaneo del registro di proporzionale espresso in bit.
intreg	L	Nessuna condizione	0	Rd	**Integral register** ($-2^{31} \div 2^{31}-1$) Registro integrale asse slave.
derreg	L	Nessuna condizione	0	Rd	**Derivate register** ($-2^{31} \div 2^{31}-1$) Registro derivata asse slave.
codeMex	L	Nessuna condizione	0	Rd	**Code M in execution** ($-2^{31} \div 2^{31}-1$) Consente la lettura del codice M del settore in esecuzione.
funInp	B	Nessuna condizione	R	RdWr	**Programmable function of input** (0÷99) Consente di configurare il funzionamento dell'ingresso normale come da tabella configurazione ingressi. (Vedi capitolo dedicato)
funInt	B	Nessuna condizione	R	RdWr	**Programmable function of interrupt input** (0÷99) Consente di configurare il funzionamento dell'ingresso in interrupt come da tabella configurazione ingressi. (Vedi capitolo dedicato)
funOut	B	Nessuna condizione	R	RdWr	**Programmable function of output** (0÷99) Consente di configurare il funzionamento dell'uscita come da tabella configurazione uscite. (Vedi capitolo dedicato)
inpcapt	B	Nessuna condizione	0	RdWr	**Capture mode** (0÷2) Definisce il modo di cattura della funzione dell'ingresso per funzione generica (vedi file di configurazione). **0** = Disabilitato. **1** = Singola cattura sul fronte di discesa. **2** = Singola cattura sul fronte di salita. La cattura è abilitata se lo stato st_enbl = 1.
intcapt	B	Nessuna condizione	0	RdWr	**Interrupt capture mode** (0÷2) Definisce il modo di cattura della funzione dell'ingresso in interrupt (vedi file di configurazione). **0** = Disabilitato. **1** = Singola cattura sul fronte di discesa. **2** = Singola cattura sul fronte di salita. La cattura è abilitata se lo stato vt_intenbl = 1.

Nome	D	Condiz. scritt.	R	A	Descrizione
errcode	B	Nessuna condizione	0	Rd	**Error code** (0÷100) Indica il tipo di errore intervenuto nel sistema. Il codice è valido solo se st_error = 1 (Vedi capitolo dedicato)
errvalue	B	Nessuna condizione	0	Rd	**Error value** (0÷100) Specifica il settore che ha causato l errore nel sistema. Il valore è valido solo se st_error = 1 (Vedi capitolo dedicato)
wrncode	B	Nessuna condizione	0	Rd	**Warning code** (0÷100) Indica il tipo di warning intervenuto nel sistema. Il codice è valido solo se st_warning = 1 (Vedi capitolo dedicato)
wrnvalue	B	Nessuna condizione	0	Rd	**Warning value** (0÷100) Specifica il settore che ha causato il warning nel sistema. Il valore è valido solo se st_warning = 1 (Vedi capitolo dedicato)

1.10.4 Variabili di programma

Nome	D	Condizioni di scrittura	R	A	Descrizione
codeG1	W	rowex ?1	R	RdWr	**Code G1** Indica il valore che assume il codice G nel passo 1. Vedi descrizione codici G.
codeG2	W	rowex ?2	R	RdWr	**Code G2** Indica il valore che assume il codice G nel passo 2. Vedi descrizione codici G.
codeG40	W	rowex ?40	R	RdWr	**Code G40** Indica il valore che assume il codice G nel passo 40. Vedi descrizione codici G.
codeQm1	L	rowex ?1	R	RdWr	**Code Q1 master** (0÷999999) Indica la quota incrementale master del settore 1. Il valore introdotto è in unità di misura.
codeQm2	L	rowex ?2	R	RdWr	**Code Q2 master** (0÷999999) Indica la quota incrementale master del settore 2. Il valore introdotto è in unità di misura.
codeQm128	L	rowex ?128	R	RdWr	**Code Q128 master** (0÷999999) Indica la quota incrementale master del settore 128. Il valore introdotto è in unità di misura.
codeQs1	L	rowex ?1	R	RdWr	**Code Q1 slave** (-999999÷999999) Indica la quota incrementale slave del settore 1. Il valore introdotto è in unità di misura.
codeQs2	L	rowex ?2	R	RdWr	**Code Q2 slave** (-999999÷999999) Indica la quota incrementale slave del settore 2. Il valore introdotto è in unità di misura.
codeQs128	L	rowex ?128	R	RdWr	**Code Q128 slave** (-999999÷999999) Indica la quota incrementale slave del settore 128. Il valore introdotto è in unità di misura.
codeQma1	L	rowex ?1	R	RdWr	**Code Q1 auxiliary master** (0÷999999) Indica la quota ausiliaria incrementale master del settore 1. Il valore introdotto è in unità di misura.
codeQma2	L	rowex ?2	R	RdWr	**Code Q2 auxiliary master** (0÷999999) Indica la quota ausiliaria incrementale master del settore. Il valore introdotto è in unità di misura.
codeQma40	L	rowex ?128	R	RdWr	**Code Q40 auxiliary master** (0÷999999) Indica la quota ausiliaria incrementale master del settore 40. Il valore introdotto è in unità di misura.
codeQsa1	L	rowex ?1	R	RdWr	**Code Q1 auxiliary slave** (-999999÷999999) Indica la quota ausiliaria incrementale slave del settore 1. Il valore introdotto è in unità di misura.
codeQsa2	L	rowex ?2	R	RdWr	**Code Q2 auxiliary slave** (-999999÷999999) Indica la quota ausiliaria incrementale slave del settore 2. Il valore introdotto è in unità di misura.

Nome	D	Condizioni di scrittura	R	A	Descrizione
codeQsa40	L	rowex ?128	R	RdWr	**Code Q40 auxiliary slave** (-999999÷999999) Indica la quota ausiliaria incrementale slave del settore 128. Il valore introdotto è in unità di misura.
codeM1	L	rowex ?1	R	RdWr	**Code M1** Introduce un codice non inerente al posizionamento, ma che identifica una variabile che potrà essere poi elaborata dal programma applicativo (codice utensile, tipo lavorazione, numero pezzi ...).
codeM2	L	rowex ?2	R	RdWr	**Code M2** Introduce un codice non inerente al posizionamento, ma che identifica una variabile che potrà essere poi elaborata dal programma applicativo (codice utensile, tipo lavorazione, numero pezzi ...).
codeM40	L	rowex ?128	R	RdWr	**Code M40** Introduce un codice non inerente al posizionamento, ma che identifica una variabile che potrà essere poi elaborata dal programma applicativo (codice utensile, tipo lavorazione, numero pezzi ...).

1.10.5 Comandi

Nome	Condizioni	Descrizone
INIT	st_init = 0	**Init** Comando di inizializzazione device. Se il device non é inizializzato non vengono eseguiti i calcoli relativi all asse e quindi rimane inattivo. Con comando INIT l asse verrà inizializzato, eseguendo i calcoli una sola volta. Attiva lo stato st_init.
EMRG	st_init = 1	**Emergency** Pone in emergenza l asse slave interrompendo, senza rampa di decelerazione, l eventuale movimento in corso. Viene inoltre disabilitata la reazione di spazio dell asse.
RESUME	st_init = 1 st_emrg = 1	**Resume** Ripristino della condizione di emergenza dell asse slave; viene riabilitata la reazione di spazio. All acquisizione dello start, l asse riprende il posizionamento.
STOP	st_init = 1 st_regoff = 0 st_emrg = 0 st_cal = 0 st_still = 0 st_camex = 0	**Stop** Interrompe l eventuale posizionamento in corso dell asse slave. La fermata dell asse avviene seguendo la rampa di decelerazione impostata nel parametro tdec. L asse rimane in reazione di spazio.
START	st_init = 1 st_regoff = 0 st_emrg = 0 st_cal = 0 st_still = 0 st_camex = 0 st_prson = 0	**Start** L asse slave inizia il posizionamento alla quota setpos con velocitá impostata in setvel.
PRESET	st_init = 1 st_regoff = 0 st_emrg = 0 st_cal = 0 st_still = 0 st_camex = 0	**Preset** Start ricerca preset asse slave. Viene dato inizio alla procedura di ricerca di preset con le modalità impostate con i parametri prsmode e prsdir. Se la ricerca di preset è già in esecuzione, il comando esegue l inversione del senso di ricerca.
RSPRSOK	st_init = 1 st_prson = 0	**Reset stato st_prsok** Azzera lo stato st_prsok
PRESETM	st_init = 1 st_camex = 0 st_prson = 0	**Master preset** Start ricerca preset asse master. Viene dato inizio alla procedura di ricerca di preset con le modalità impostate con il parametro prsmodem.
RSPRSM	st_init = 0 st_prson = 0	**Reset preset of master** Azzera lo stato st_prsokm se il preset del master è concluso. Se il preset del master è in corso (st_prsonm = 1) viene bloccato
RSERR	st_init = 1	**Reset status st_error** Azzera lo stato st_error ed il relativo codice di errore errcode ed errvalue.
RSWRN	st_init = 1	**Reset status st_warning** Azzera lo stato st_warning ed il relativo codice di warning wrncode ed wrnvalue.
LOOPON	st_init = 1 st_loopon = 1	**Loop on** Abilita la reazione di spazio dell asse slave. L uscita analogica contrasta ogni azione esterna che tenti di spostare l asse dalla posizione raggiunta (deriva, operatore, ...). Questa operazione azzera l eventuale errore di inseguimento follerr.
LOOPOFF	st_init = 1 st_loopon = 1	**Loop off** Disabilita la reazione di spazio dell asse slave. L asse può essere spostato dalla sua posizione senza che l uscita analogica contrasti il movimento.

Nome	Condizioni	Descrizone
MANFW	st_init = 1 st_regoff = 0 st_prson = 0 st_camex = 0 st_cal = 0 st_still = 1 st_emrg = 0	**Forward** Movimento manuale asse slave in avanti. Comanda il movimento manuale in avanti dell'asse alla velocità impostata con setvel. Il movimento viene fermato con il comando di STOP.
MANBW	st_init = 1 st_regoff = 0 st_prson = 0 st_camex = 0 st_cal = 0 st_still = 1 st_emrg = 0	**Backward** Movimento manuale asse slave in indietro. Comanda il movimento manuale indietro dell'asse alla velocità impostata con setvel. Il movimento viene interrotto con il comando di STOP.
CALON	st_init = 1	**Volt generator on** L'uscita analogica dell'asse slave viene impiegata come generatore di tensione; in questo caso non è possibile usarla per posizionare l'asse. Il valore in uscita è settabile a piacere tramite la variabile vout.
CALOFF	st_init = 1 st_cal = 0	**Volt generator off** L'uscita analogica dell'asse slave non viene gestita come generatore di tensione, pertanto può essere nuovamente usata per la gestione dei posizionamenti.
CNTLOCK	st_init = 1	**Lock counter** Blocca l'acquisizione del conteggio dell'asse slave anche se il trasduttore continua ad inviare i segnali. In questa fase l'eventuale spostamento dell'asse non viene rilevato.
CNTUNLOCK	st_init = 1	**Unlock counter** Sblocca il conteggio dell'asse slave. Viene ripresa la lettura dei segnali inviati dal trasduttore e, di conseguenza, l'aggiornamento del conteggio.
CNTREV	st_init = 1	**Reverse counter** Consente di invertire le fasi del trasduttore slave all'interno del device. Viene quindi invertito il senso del conteggio (Incremento/decremento).
CNTDIR	st_init = 1	**Direct counter** Ripristina la direzione del conteggio del trasduttore dell'asse slave.
CNTLOCKM	st_init = 1	**Lock counter master** Blocca l'acquisizione del conteggio asse master anche se il trasduttore continua ad inviare i segnali. In questa fase l'eventuale spostamento dell'asse non viene rilevato.
CNTUNLOCKM	st_init = 1	**Unlock counter master** Sblocca il conteggio dell'asse master. Viene ripresa la lettura dei segnali inviati dal trasduttore e, di conseguenza, l'aggiornamento del conteggio.
CNTREVM	st_init = 1	**Reverse counter master** Consente di invertire le fasi del trasduttore master all'interno del device. Viene quindi invertito il senso del conteggio (Incremento/decremento).
CNTDIRM	st_init = 1	**Direct counter master** Ripristina la direzione del conteggio del trasduttore dell'asse master.
STOPCAM	st_init = 1 st_camex = 1	**Stop cam** Interrompe la camma in corso. La fermata dell'asse avviene seguendo una rampa di decelerazione asincrona, secondo il parametro tdec. L'asse rimane in reazione di spazio.
STARTCAM	st_init = 1 st_still = 1 st_camex = 1 st_prson = 0 st_emrg = 0 st_regoff = 0	**Start cam** L'asse inizia il posizionamento dell'asse slave partendo con l'elaborazione del settore 1 della camma introdotta ed eseguendo il codice descritto.
REGOFF	st_init = 1 st_still = 1 st_camex = 0 st_prson = 0	**Regulation OFF** Disabilita la regolazione e l'aggiornamento del DAC dell'asse slave, nonché tutti i comandi di movimento.
REGON	st_init = 1 st_still = 1 st_regoff = 1 st_emrg = 0	**Regulation ON** Riabilita la regolazione e l'aggiornamento del DAC dell'asse slave, nonché tutti i comandi di movimento.
ENBL	st_init = 1	**Reverse counter** Consente di invertire le fasi del trasduttore slave all'interno del device. Viene quindi invertito il senso del conteggio (Incremento/ decremento).
INTENBL	st_init = 1	**Direct counter** Ripristina la direzione del conteggio del trasduttore dell'asse slave.
DSBL	st_init = 1	**Lock counter master** Blocca l'acquisizione del conteggio asse master anche se il trasduttore continua ad inviare i segnali. In questa fase l'eventuale spostamento dell'asse non viene rilevato.
INTDSBL	st_init = 1	**Interrupt disable** Disabilita la funzione dell'ingresso in interrupt inserita nel parametro funInt. Disattiva lo stato st_intenbl.
RSCAPT	st_init = 1 st_capt = 1	**Reset status of capture input** Disattiva lo stato di st_capt.
RSINTCAPT	st_init = 1 st_intcapt = 1	**Reset status of capture interrupt input** Disattiva lo stato di st_intcapt.

Nome	Condizioni	Descrizone
DELCNT	st_init = 1 st_still = 1 st_camex = 0 st_prson = 0 st_cal = 0 st_regoff = 0	**Delta counter** Il conteggio dell'asse slave (posizione dell'asse) viene modificato sommandogli algebricamente il valore specificato nel parametro delta1 (posit = posit + delta1).
DELCNTM	st_init = 1 st_prsonm = 0 st_camex = 0	**Delta counter of master** Il conteggio dell'asse master (posizione dell'asse) viene modificato sommandogli algebricamente il valore specificato nel parametro delta2 (positm = positm + delta2).

1.10.6 Stati

Nome	Dim.	Condiz. scritt.	Accesso	Descrizione
st_init	F	Nessuna condizione	Rd	**Init** Segnalazione di device inizializzato. **0** = device non inizializzato **1** = device inizializzato All'accensione per default viene caricato il valore zero.
st_chvel	F	Nessuna condizione	Rd	**Status of enable velocity change** Segnala che il device può accettare un setpoint di velocità dell'asse slave diverso da quello in esecuzione e porlo in esecuzione eseguendo la procedura di cambio velocità. La procedura di cambio velocità è disponibile solamente durante i posizionamenti (non durante l'esecuzione della camma). All'accensione per default viene caricato il valore zero.
st_emrg	F	Nessuna condizione	Rd	**Emergency** (0÷1) Segnalazione di asse slave in emergenza. **0** = asse non in emergenza **1** = asse in emergenza All'accensione per default viene caricato il valore zero.
st_toll	F	Nessuna condizione	Rd	**Tolerance** (0÷1) Segnalazione di asse slave in tolleranza rispetto alla quota posta in esecuzione dal comando di START. **0** = asse non in tolleranza **1** = asse in tolleranza All'accensione per default viene caricato il valore zero.
st_tpos	F	Nessuna condizione	Rd	**Tolerance of set position** (0÷1) Indica che il conteggio dell'asse slave è all'interno della fascia di tolleranza rispetto alla quota presente nella variabile setpos indipendentemente dal fatto che sia stato dato uno START o no. **0** = asse non in tolleranza **1** = asse in tolleranza All'accensione per default viene caricato il valore zero.
st_prson	F	Nessuna condizione	Rd	**Preset ON** (0÷1) Segnalazione di ricerca di preset asse slave conclusa correttamente **0** = ricerca di preset non ancora conclusa o non eseguita **1** = ricerca di preset conclusa correttamente All'accensione per default viene caricato il valore zero.
st_prsok	F	Nessuna condizione	Rd	**Preset ok** (0÷1) Segnalazione di ricerca di preset asse slave conclusa correttamente. **0** = ricerca di preset non ancora conclusa o non eseguita **1** = ricerca di preset conclusa correttamente All'accensione per default viene caricato il valore zero.
st_prsonm	F	Nessuna condizione	Rd	**Preset of master ON** (0÷1) Segnalazione di ricerca di preset asse masterin corso. **0** = ricerca di preset non in corso **1** = ricerca di preset in corso All'accensione per default viene caricato il valore zero.
st_prsokm	F	Nessuna condizione	Rd	**Preset ok of master** (0÷1) Segnalazione di ricerca di preset asse master conclusa correttamente. **0** = ricerca di preset non ancora conclusa o non eseguita **1** = ricerca di preset conclusa correttamente All'accensione per default viene caricato il valore zero.

Nome	Dim.	Condiz. scritt.	Accesso	Descrizione
st_still	F	Nessuna condizione	**Rd Still** (0÷1) Segnalazione di asse slave fermo. Durante l'esecuzione della camma questo stato è uguale ad 1. **0** = asse in movimento **1** = asse fermo All'accensione per default viene caricato il valore 1.	
st_camex	F	Nessuna condizione	Rd	**Cam to execution** (0÷1) Segnalazione di camma in esecuzione. **0** = camma non in esecuzione **1** = camma in esecuzione All'accensione per default viene caricato il valore zero.
st_movdir	F	Nessuna condizione	Rd	**Direction BW** (0÷1) Segnalazione della direzione del movimento dell'asse slave solamente se non si sta eseguendo una camma (st_camex = 0). **0** = avanti **1** = indietro All'accensione per default viene caricato il valore zero.
st_loopon	F	Nessuna condizione	Rd	**Loop ON** (0÷1) Segnalazione di asse slave in reazione di spazio. **0** = asse non in reazione di spazio **1** = asse in reazione di spazio All'accensione per default viene caricato il valore zero.
st_foller	F	Nessuna condizione	Rd	**Following error** (0÷1) Segnalazione di asse slave in errore di inseguimento (ritenuta 500 ms) **0** = asse non in errore di inseguimento **1** = asse in errore di inseguimento All'accensione per default viene caricato il valore zero.
st_sync	F	Nessuna condizione	Rd	**Syncronism** (0÷1) Segnalazione di asse slave in sincronismo durante l'esecuzione della camma: **0** = asse non in sincronismo **1** = asse in sincronismo All'accensione per default viene caricato il valore 0.
st_cal	F	Nessuna condizione	Rd	**Calibration** (0÷1) Segnalazione di asse slave come generatore di tensione. **0** = generatore di tensione asse disattivo **1** = generatore di tensione asse attivo All'accensione per default viene caricato il valore zero.
st_cntlock	F	Nessuna condizione	**Rd Locked** (0÷1) Segnalazione di conteggio asse slave bloccato. **0** = Conteggio asse sbloccato **1** = Conteggio asse bloccato All'accensione viene mantenuto lo stato presente allo spegnimento.	
st_cntrev	F	Nessuna condizione	Rd	**Reversed** (0÷1) Segnalazione di conteggio asse slave invertito. **0** = Conteggio asse sbloccato **1** = Conteggio asse bloccato All'accensione viene mantenuto lo stato presente allo spegnimento.
st_cntlockm	F	Nessuna condizione	Rd	**Master locked** (0÷1) Segnalazione di conteggio asse master bloccato. **0** = Conteggio asse sbloccato **1** = Conteggio asse bloccato All'accensione viene mantenuto lo stato presente allo spegnimento.

Nome	Dim.	Condiz. scritt.	Accesso	Descrizione
st_cntrevm	F	Nessuna condizione	Rd	**Master reversed** (0÷1) Segnalazione di conteggio asse master invertito. **0** = Conteggio asse non invertito **1** = Conteggio asse invertito. All accensione viene mantenuto lo stato presente allo spegnimento.
st_regoff	F	Nessuna condizione	Rd	**Regulation OFF** (0÷1) Segnalazione di regolazione asse slave é disabilitata e aggiornamento DAC non effettuato. **0** = regolazione sbloccata **1** = regolazione bloccata All accensione per default viene caricato il valore zero.
st_enbl	F	Nessuna condizione	Rd	**Normal input enabled** (0÷1) Segnala l abilitazione della funzione dell ingresso normale inserita nel parametro funInp. Viene attivato dal comando ENBL e disattivato dal comando DSBL. Viene disattivato automaticamente a cattura avvenuta **0** = Cattura del conteggio non è abilitata **1** = Cattura del conteggio abilitata. All accensione per default viene caricato il valore zero.
st_intenbl	F	Nessuna condizione	Rd	**Interrupt input enabled** (0÷1) Segnala l abilitazione della funzione dell ingresso in interrupt inserita nel parametro funInt. Viene attivato dal comando INTENBL e disattivato dal comando INTDSBL. Viene disattivato automaticamente a cattura avvenuta. **0** = Cattura del conteggio non è abilitata **1** = Cattura del conteggio abilitata. All accensione per default viene caricato il valore zero.
st_capt	F	Nessuna condizione	Rd	**Capture of normal input** (0÷1) Viene attivato alla cattura della funzione impostata in funInp; viene resettato dal comando RSCAPT. **0** = Cattura non eseguita. **1** = Eseguita cattura. All accensione per default viene caricato il valore zero.
st_intcapt	F	Nessuna condizione	Rd	**Capture of interrupt input** (0÷1) Viene attivato alla cattura della funzione impostata in funInt; viene resettato dal comando RSINTCAPT **0** = Cattura non eseguita. **1** = Eseguita cattura. All accensione per default viene caricato il valore zero.
st_int	F	Nessuna condizione	Rd	**Status of interrupt line** (0÷1) Indica lo stato della linea di interrupt di uso generico. **0** = Ingresso in interrupt disattivo. **1** = Ingresso in interrupt attivo. All accensione per default viene caricato il valore zero.
st_error	F	Nessuna condizione	Rd	**Status of camming device error** (0÷1) Indica lo stato di errore nel device CAMMING3. Per la decodifica dell errore si deve fare riferimento alla variabile errcode ed errvalue. **0** = Errore non presente. **1** = Errore presente. All accensione per default viene caricato il valore zero.
st_warning	F	Nessuna condizione	Rd	**Status of camming device warning** (0÷1) Indica lo stato di warning nel device CAMMING3. Per la decodifica del warning si deve fare riferimento alla variabile wrncode ed wrnvalue. **0** = Warning non presente. **1** = Warning presente. All accensione per default viene caricato il valore zero.

1.10.7 Limitazioni del device

1. Non è possibile mettere in sequenza più di 7 settori a campionamento zero.
2. Non è possibile mettere in sequenza più di 3 settori di aggiornamento conteggio.
3. Con i parametri:
   pulse = 999999
   measure = 934

maxvel = 999999
unitvel = 0
decpt = 3
Si stabiliscono le condizioni per creare degli overflow nei calcoli dei settori 150, 151, 152 e 153.

4. Durante l'esecuzione della camma (st_camex = 1), non è possibile cambiare il settore in esecuzione e quello eseguito successivamente.
5. Il device è stato creato per funzionare con il master che si incrementa. È possibile eseguire la camma con il master che decrementa sottostando alle seguenti condizioni:
   La camma si ferma rimane in reazione di spazio sul settore precedente se incontra i settori 130, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 160. Può eseguire il settore precedente solamente se è già stato eseguito almeno una volta (potrebbe non essere stato eseguito a causa di un jump).

Documento generato automaticamente da **Qem Wiki** - https://wiki.qem.it/
Il contenuto wiki è costantemente aggiornato dal team di sviluppo, è quindi possibile che la versione online contenga informazioni più recenti di questo documento.