

## Sommario

<b>AN004- Esempio di utilizzo e calibrazione del device EANPOS</b> .....	3
Dichiarazione del device nella unit di configurazione .....	3
Corretta parametrizzazione del device .....	3
Movimentazione preliminare .....	5
Parametrizzazione uscita analogica (impostazione parametro maxvel) .....	5
Sviluppo di un applicativo che implementi un posizionatore .....	7

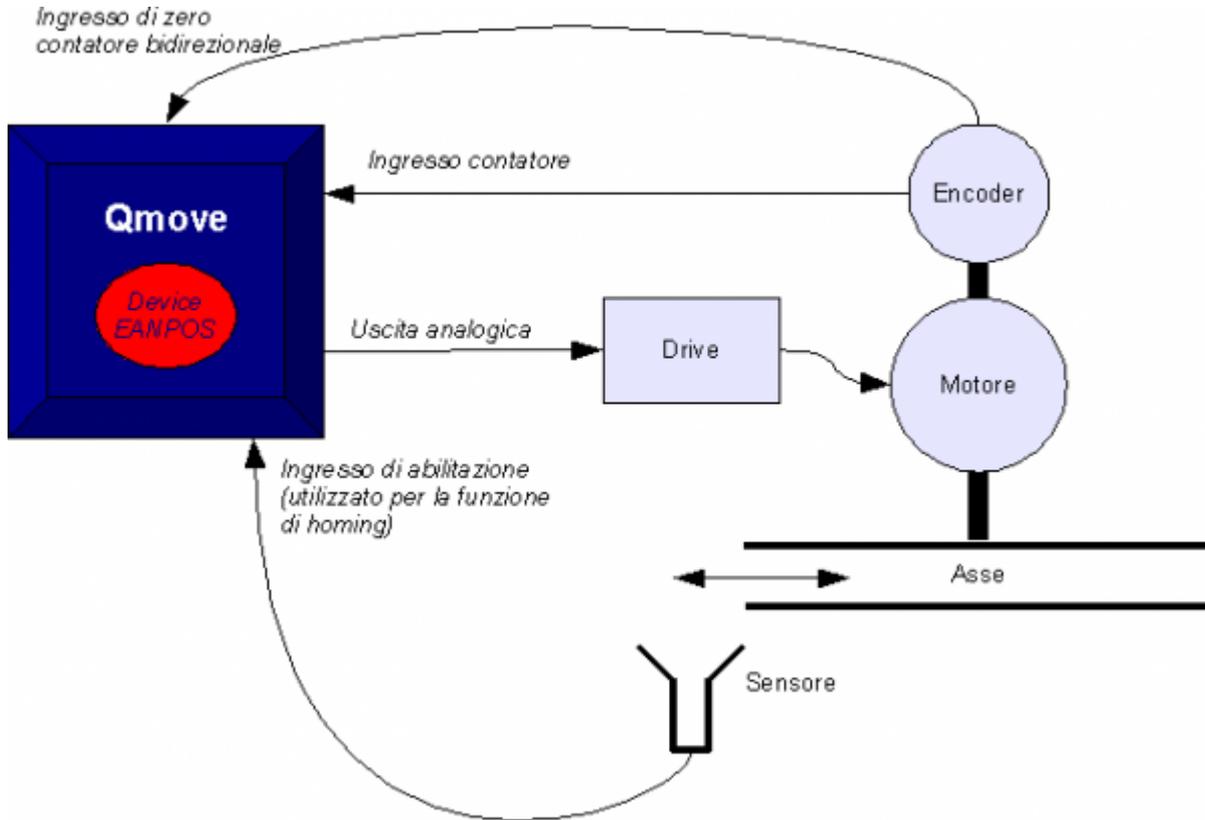


## AN004- Esempio di utilizzo e calibrazione del device EANPOS

In questa sezione si vogliono descrivere le prime operazioni che si troverà a fare l'utente nel suo primo approccio con il device EANPOS. Inoltre si vuole fornire un semplice esempio in cui si utilizzi il device EANPOS per implementare un posizionatore analogico.

Possiamo suddividere nelle seguenti sezioni il procedere dell'operazione:

- dichiarazione del device nella unit di configurazione
- introduzione dei parametri al fine di calibrare correttamente ingressi ed uscite
- sviluppo dell'applicativo secondo le esigenze



### Dichiarazione del device nella unit di configurazione

Come è stato già spiegato nella sezione di descrizione del device EANPOS, è necessario programmare correttamente la unit di configurazione dell'applicativo. È molto importante la porzione di codice in cui vengono dichiarati i device, qui infatti si dovranno indicare le risorse hardware da utilizzare per garantire un corretto funzionamento. Sarà compito del programmatore individuare e scegliere ingressi ed uscite più opportune. Ad esempio con la seguente riga di codice:

```

;-----
; Dichiarazione device interni
;-----
INTDEVICE
AsseX EANPOS 2 2.CNT02 3 3.INP01 3.AN01

```

Si definisce un device EANPOS con nome "AsseX" il cui tempo di campionamento è di 2 ms. Sono state dichiarate inoltre le seguenti risorse hardware: l'ingresso del contatore bidirezionale ha indirizzo 2.CNT02 (dove 2 indica lo slot su cui è installata la scheda mentre CNT02 è il nome mnemonico dell'ingresso), il numero dell'ingresso digitale per interruzione dedicata all'impulso di zero del contatore bidirezionale (numero 3), un ingresso digitale di abilitazione per l'acquisizione dell'impulso di zero (3.INP01) e l'indirizzo dell'uscita analogica 3.AN01 che regola la tensione in uscita per comandare un azionamento.

Un applicativo che abbia soltanto al suo interno la dichiarazione del device nella unit di configurazione ed una unit qcl che non esegua alcuna operazione (salvo quella di WAIT forzato) consente già di compiere le prime operazioni sfruttando le funzionalità del device. Infatti dopo avere trasferito l'applicativo nel sistema e avendolo messo in stato di RUN, sarà già possibile modificare i parametri, osservare gli stati o dare comandi al device tramite il monitor apposito da QView. Questo risulta molto comodo nelle prime fasi della programmazione quando si vogliono solo verificare alcuni funzionamenti oppure in fase di debug di un applicativo.

### Corretta parametrizzazione del device

Una volta dichiarate correttamente le risorse hardware da utilizzare è necessario impostare alcuni parametri in base alle componenti che sono collegate al prodotto Qmove.

## Introduzione di measure e pulse

Prendiamo in considerazione il caso (piuttosto diffuso) in cui il contatore bidirezionale sia un encoder digitale. Supponiamo che questo sia direttamente calettato su un motore che debba movimentare un asse. Sarà necessario impostare correttamente i parametri *measure* e *pulse* del device EANPOS in modo che quest'ultimo possa interpretare gli impulsi del contatore e calcolare la posizione dell'asse.

L'introduzione di *measure* e *pulse* consente di stabilire una corrispondenza tra uno spazio in una unità di misura prescelta ed un determinato numero di impulsi. Nel caso in cui l'utilizzatore conosca già lo spazio percorso in un giro encoder allora potrà procedere direttamente all'inserimento dei valori.

Chiariamo questo concetto con un esempio: se l'encoder genera 1000 impulsi giro e si è a conoscenza che l'asse si muove di 5 cm quando l'encoder compie precisamente un giro allora si potranno inserire i seguenti valori:

```
AsseX:measure = 50;
AsseX:pulse = 4000
```

Il valore di *measure* introdotto inoltre implica la scelta di un'unità di misura del millimetro per misurare le posizioni, nel parametro *pulse* si è introdotto un valore pari al numero di impulsi encoder moltiplicato per 4. Si ricorda che il rapporto *measure/pulse* deve avere un valore compreso tra 0.00935 e 1 (per rispetto dei limiti di precisione del device). È importante sottolineare che i valori appena descritti sono presi come riferimento: non è necessario introdurre i parametri prendendo come riferimento un giro encoder come si andrà a descrivere di seguito.

Quando l'utilizzatore non conosce preventivamente i parametri di misura, potrà comunque effettuare la corretta calibrazione seguendo questi passi:

- dare il comando di *INIT* al device, verificare che lo stato *st\_init* commuti a 1
- tramite il "device monitor" di QView visualizzare sul pc il valore del parametro *posit*
- impostare *measure* e *pulse* entrambi al valore 1
- movimentare l'asse manualmente facendogli compiere uno spostamento di una posizione facilmente misurabile
- leggere il valore di *posit*
- a questo punto inserire con la unità di misura desiderata il valore misurato nel parametro *measure* ed il valore del parametro *posit* nel parametro *pulse*.

La risoluzione dell'encoder è ora correttamente impostata.

Un'ulteriore operazione importante da compiere è impostare i parametri *maxpos* e *minpos* che definiscono rispettivamente la massima e la minima posizione raggiungibile dall'asse.

## Scelta dell'unità di misura della velocità

L'unità di misura della velocità istantanea dell'asse viene scelta tramite i parametri *unitvel* e *decpt*. È possibile scegliere l'unità di tempo della velocità con il parametro *unitvel*: se questo è pari a 0 allora la velocità è misurata in Um/min, se è pari a 1 allora è misurata in Um/s. Il parametro *decpt* invece stabilisce se misurare i valori di velocità in multipli dell'unità di misura fondamentale Um. Ad esempio, se l'unità di misura fondamentale Um=mm, ed *unitvel*=1 si ottiene la visualizzazione della velocità nella variabile *vel* in:

mm/s (con *decpt* = 0),  
 cm/s (con *decpt* = 1),  
 dm/s (con *decpt* = 2),  
 m/s (con *decpt* = 3).

In seguito, se necessario, bisognerà configurare correttamente la visualizzazione sul terminale operatore per aggiustare la corretta posizione del punto decimale.

## Calibrazione uscita analogica



Attenzione:  
 prima di effettuare dei posizionamenti veri e propri è necessario verificare che collegamenti ed organi meccanici non siano causa di malfunzionamenti.

Esaminiamo il caso in cui il device EANPOS utilizzi un'uscita analogica implementata con un dispositivo DAC: questo assumerà

in ingresso valori discreti con risoluzione 16 bit (perciò compresi fra -32768 e 32767) per dare in uscita una tensione analogica con range  $\pm 10V$ . Con la funzione di calibrazione questa uscita analogica può essere pilotata con un valore costante con lo scopo di verificare collegamenti e funzionalità.

## Movimentazione preliminare

In questa sezione vengono descritte le operazioni da effettuare per verificare la correttezza dei collegamenti e la funzionalità del sistema che è stato costruito.

- dare il comando di *INIT* al device, verificare che lo stato *st\_init* commuti a 1
- dare il comando *RESUME* per togliere un'eventuale condizione di emergenza (*st\_emrg* = 1)
- abilitare lo stato di calibrazione asse dando il comando *CALON*, lo stato *st\_cal* commuta a 1
- in queste condizioni è possibile impostare la tensione analogica tramite il parametro *vout*: il valore è espresso in decimi di Volt (perciò il range di valori introducibili è  $\pm 100$ ). Si consiglia di introdurre valori bassi (5, 10, 15 ...)
- poiché ora il device è utilizzato come "generatore di tensione" l'asse dovrebbe iniziare a muoversi. Se ciò non accade è opportuno verificare la correttezza dei collegamenti. Quando l'asse è in movimento il parametro *frq* indica il valore della frequenza di uno dei segnali in ingresso al contatore bidirezionale, *vel* indica la velocità dell'asse mentre *posit* la posizione secondo l'unità di misura scelta. Se dando tensione positiva la posizione decrementa è necessario invertire le fasi del trasduttore (o fisicamente spostando i cavi, oppure tramite il comando *CNTREV*) o invertire la direzione dell'azionamento
- se con tensione di uscita pari a 0 V si nota comunque che l'asse è in movimento a causa di tensioni di offset, queste si possono controbilanciare agendo sul parametro *offset*. Per un ottimale esito della taratura, l'operazione deve essere eseguita con il sistema in regime di temperatura
- ora è possibile disabilitare lo stato di calibrazione con il comando *CALOFF* (lo stato *st\_cal* commuta a 0)

## Parametrizzazione uscita analogica (impostazione parametro maxvel)

Il device EANPOS genera il valore di tensione dell'uscita analogica sulla base di una proporzione tra la velocità massima dell'asse e la massima tensione di uscita. Per fare ciò è necessario impostare il parametro *maxvel*, ovvero la velocità con cui si muove l'asse quando viene data tensione massima all'azionamento. Ovviamente l'asse deve avere un comportamento simmetrico rispetto al valore zero di tensione analogica, quindi la velocità deve essere la stessa (in modulo) alla tensione massima sia positiva che negativa.

Per conoscere la velocità massima esistono due vie: il "metodo teorico" presuppone di conoscere la velocità massima impostata nell'azionamento da cui si può ricavare facilmente la velocità lineare.

Se non si è a conoscenza della velocità massima dichiarata del motore si deve procedere in questo modo:

- entrare nella modalità di calibrazione (come descritto precedentemente)
- se il sistema lo permette fornire all'azionamento tensione massima e leggere il valore del parametro *vel*
- è possibile anche fornire una tensione minore e calcolare la velocità massima con la proporzione *vout* : 10  

$$V = vel : maxvel$$

Ora è quindi possibile introdurre il valore della velocità massima nel parametro *maxvel*.

## Prima movimentazione



Attenzione:  
prima di movimentare l'asse, verificare il corretto funzionamento dei dispositivi d'emergenza e protezione.

Le procedure fin qui descritte hanno permesso di completare la prima fase di parametrizzazione del device. Ora è possibile eseguire una semplice movimentazione dell'asse. Seguire ad esempio i seguenti passi:

- dare il comando di *INIT* al device, verificare che lo stato *st\_init* commuti a 1
- spostare l'asse in una posizione tale per cui si possa compiere un determinato spazio senza incontrare fincorsa
- impostare il parametro che definisce il tempo impiegato dall'asse a raggiungere la velocità massima da 0 (e viceversa) *taccdec*=50

- impostare la velocità di posizionamento con il parametro *setvel*
- impostare la quota di posizionamento con il parametro *setpos*
- resettare l'eventuale stato di emergenza con il comando *RESUME*
- avviare il posizionamento con il comando *START*, per arrestare il movimento dare il comando *STOP* (oppure *EMRG*).

Questa prima movimentazione è stata eseguita senza attivare l'anello di reazione di spazio, quindi qualsiasi errore introdotto da valori di tensione di offset o da agenti esterni non viene corretto.

## Prima impostazione dei parametri del controllore PID

Fino ad adesso è stato spiegato come effettuare dei primi posizionamenti con l'anello di spazio aperto, vediamo ora come effettuare una prima calibrazione dei parametri PID.

Seguire questa procedura:

- dare il comando *LOOPON* al device, verificare che lo stato *st\_loopon* commuti a 1
- in questo momento l'anello di spazio sarà chiuso
- impostare il parametro *feedfw* a 1000 (ovvero al 100%)
- dare un valore arbitrario al parametro *pgain*, ad esempio 20
- aumentare il valore del parametro *pgain* fino che non si nota che l'asse collegato al motore che si sta controllando inizia a vibrare
- a questo punto iniziare a diminuire il valore di *pgain* fino a quando questo fenomeno di vibrazione cessa
- il valore dei parametri *integt* e *derivt* possono essere lasciati a 0. Nel caso serva seguire la procedure di calibrazione qui sotto per dare loro un valore.

## Utilizzo del device RECDATA per calibrare correttamente il controllore PID

Quando il programmatore intende regolare finemente i parametri del regolatore PID, egli può servirsi di un device interno (residente sul QMove) che permette di registrare dati del tipo: posizione in impulsi primari, posizioni virtuali asse, uscite analogiche, errori di inseguimento, stati di ingressi e uscite. Questo tool si chiama RECDATA e sfrutta la RAM della CPU del sistema QMove, perciò, quando viene impiegato, l'utilizzo della memoria dei dati non deve superare il 50% di quella totale.

Poiché gli effetti della regolazione PID producono manifestazioni molto difficili da apprezzare ad occhio nudo, il device RECDATA diventa molto utile quando si vogliono tarare parametri quali *pgain*, *integt* o *derivt*. Infatti fenomeni come l'overshoot o un rallentamento dovuto ad un valore eccessivo del tempo integrale devono essere tenuti in considerazione nella fase di parametrizzazione.

L'utilizzo di un device RECDATA risulta molto semplice, è sufficiente infatti che esso sia dichiarato correttamente nella unit di configurazione con una riga del tipo:

```

-----
; Dichiarazione device interni
-----
INTDEVICE
Recorder RECDATA TCamp QCTL1 QCTL2 IOutA1 IOutA2 IntL1 IntL2 Ing1 Ing2 Out1 Out2

```

Il device è particolarmente indicato a monitorare le regolazioni di un device EANPOS infatti consente di visualizzare: i valori di posizione in impulsi primari (prendendo direttamente i dati provenienti dal contatore bidirezionale), i valori di posizione teorica che il generatore di profilo di velocità ha calcolato, i valori di una uscita analogica. L'utente può quindi comodamente accorgersi se i parametri che ha inserito sono corretti per le sue esigenze.

Il device RECDATA può registrare informazioni riguardanti due assi, per il nostro scopo è possibile dichiarare solo gli indirizzi relativi al primo asse lasciando a X.X gli altri.

Una descrizione più approfondita del device RECDATA si può trovare nella sezione apposita, qui ci limitiamo ad elencare i passi da seguire per utilizzarlo:

- dichiarazione del device nella unit di configurazione inserendo come parametri ingressi ed uscite utilizzati anche dal device EANPOS
- aprire il pannello dal menu a tendina Monitor... e decidere che parametri registrare
- dare il comando *STARTR* nel momento in cui si vuole iniziare la registrazione, e *STOPR* quando la si vuole arrestare
- ad operazioni completate dal menu "Monitor" di QView selezionare "Device Panels", e da lì il valore relativo al device RECDATA
- nella finestra che si aprirà sono disponibili diverse opzioni, l'operazione più importante da compiere è dare

il comando "Start data acquiring" tramite l'apposito icona

- sugli assi cartesiani, a fine caricamento, compariranno i valori dei parametri che sono stati monitorati.



Nota:

il device RECDATA registra una notevole quantità di dati all'interno della RAM del prodotto QMove, al momento dell'acquisizione da parte del pc il trasferimento avviene via seriale. Questa operazione potrebbe risultare molto lunga a causa della bassa velocità intrinseca del protocollo di comunicazione seriale. L'utente quindi può adottare alcuni accorgimenti come:

- prestare attenzione a registrare solo le variabili che necessitano effettivamente di essere monitorate
- aumentare il tempo di campionamento del device RECDATA

## Sviluppo di un applicativo che implementi un posizionatore

Nella sezione precedente è stato spiegato quali sono i primi passi da seguire per una procedura di movimentazione di un asse con un posizionatore analogico. Tale esempio impiega solo un ristretto spettro dei parametri impostabili del device, in questa sezione inseriamo un codice di esempio, dettagliatamente commentato, da cui l'utilizzatore può prendere spunto per sviluppare un applicativo. Il modo in cui il device va dichiarato è spiegato precedentemente, e perciò in questa sezione è omessa la unit di configurazione. [Vedi qui](#)

Questo applicativo utilizza 4 ingressi: 1 per avanti manuale, 1 per indietro manuale, 1 per comando di start ed 1 per comando di stop. Saranno disponibili la modalità di funzionamento manuale ed automatica: in manuale è possibile utilizzare i comandi di jog mentre in automatico è possibile far effettuare un posizionamento alla posizione ed alla velocità desiderata:

```

*****
; Unit di configurazione (sono riportate solo le dichiarazioni di variabili e
; sono omesse le dichiarazioni del bus e del device)
*****
;-----
; Definizione Variabili SYSTEM
;-----
SYSTEM
slQuotaPos L           ;Variabile per quota di posizionamento
slVelAsse L           ;Variabile per velocità asse
;-----
; Definizione Variabili GLOBAL
;-----
GLOBAL
gfMovMan F           ;Flag segnalazione movimenti manuali in corso
gfMovAuto F          ;Flag segnalazione movimenti automatici in corso
;-----
; Definizione Variabili INPUT
;-----
INPUT
ifAvMan F 2.INP01     ;Ingresso di avanti manuale
ifInMan F 2.INP02     ;Ingresso di indietro manuale
ifStart F 2.INP03     ;Ingresso di START asse
ifStop F 2.INP04      ;Ingresso di STOP asse
;-----
; Definizione Variabili OUTPUT
;-----
OUTPUT
ofToll F 2.OUT01      ;Uscita di asse in tolleranza
ofAxeFermo F 2.OUT02 ;Uscita di asse fermo
;-----
;-----
; Unit qcl
;-----
; Operazioni di parametrizzazione del device
;-----
Asse:measure = 10000
Asse:pulse = 40000 ;come calcolare measure e pulse è spiegato nel paragrafo apposito.*
Asse:maxvel = 100000 ;come calcolare maxvel è spiegato nel paragrafo apposito.*
Asse:maxpos = 999999 ;Quota massima
Asse:minpos = -999999 ;Quota minima
Asse:maxfollerr = 10000 ;Errore di inseguimento massimo
Asse:unitvel = 0 ;Unità di tempo della velocità (velocità in Um/min)
Asse:decpt = 0 ;Cifre decimali nel calcolo della velocità
Asse:rampmode = 0 ;Tipo di rampe utilizzate (in questo caso stesso tempo di accelerazione
; o decelerazione da velocità zero a velocità massima)
Asse:tacdec = 100 ;Tempo di accelerazione e di decelerazione
Asse:pgain = 10 ;Guadagno proporzionale
Asse:feedfw = 1000 ;Feedforward

INIT Asse ;Inizializza il device
WAIT Asse:st_init ;Attendi che il device sia inizializzato
CNTUNLOCK Asse ;Sblocca acquisizione posizione
WAIT NOT Asse:st_cntlock ;Attendi che l'acquisizione della posizione sia sbloccato
CNTDIR Asse ;Imposta il senso dell'acquisizione di posizione
WAIT NOT Asse:st_cntrev ;Attendi che sia impostato il senso dell'acquisizione di posizione
REGION Asse ;Abilita la regolazione
WAIT NOT Asse:st_regoff ;Attendi l'abilitazione alla regolazione
RESUME Asse ;Togli l'asse dallo stato di emergenza
WAIT NOT Asse:st_emrg ;Attendi che l'asse non sia in emergenza
LOOPON Asse ;Aggancia loop di reazione dell'asse
WAIT Asse:st_loopon ;Attendi che sia agganciato il loop di reazione dell'asse
IF (slVelAsse EQ 0) ;Nel caso in cui il set di velocità dell'asse sia zero
  slVelAsse = 50 ;Imposta una velocità di posizionamento
ENDIF
IF (slQuotaPos EQ 0) ;Nel caso in cui la quota di posizionamento dell'asse sia zero

```

```

slQuotaPos = 2000 ;Imposta una quota di posizionamento
ENDIF
;-----
; Operazioni di Posizionamento
;-----
;----- variabili utilizzate -----
; slVelAsse: Variabile impostabile che rappresenta la velocità dell'asse (espressa in % della vel.max.)
; slQuotaPos: Variabile impostabile che rappresenta la quota di posizionamento dell'asse
;----- flag utilizzati -----
; gfMovMan: Movimento manuale in corso
; gfMovAuto: Movimento automatico in corso
;-----
MAIN:
;-----
; Gestione uscite
offToll = Asse:st_toll ;Imposto l'uscita di tolleranza come lo stato di tolleranza
ofAxeFermo = Ass:st_still ;Imposto l'uscita di asse fermo come lo stato di asse fermo
;-----
; Gestione movimenti automatici
IF ifStart ;Attende l'ingresso di START
IF NOT gfMovMan ;Controlla che non ci siano movimenti manuali
IF Asse:st_still ;Controlla che l'asse sia fermo
Asse:setvel=(slVelAsse*Asse:maxvel)/100 ;Imposta la velocità dell'asse
Asse:setpos = slQuotaPos ;Imposta la quota di posizionamento
START Asse ;Esegue lo start dell'asse
gfMovAuto = 1 ;Segnala movimento automatico in corso
ENDIF
ENDIF
IF ifStop ;Attende l'ingresso di STOP
IF NOT Asse:st_still ;Controlla che l'asse NON sia fermo
STOP Asse ;Esegue lo stop dell'asse
ENDIF
ENDIF
IF gfMovAuto ;Controlla segnalazione movimento automatico in corso
IF Asse:st_still ;Controlla che l'asse sia fermo
gfMovAuto = 0 ;Resetta stato di movimento Automatico
ENDIF
ENDIF
;-----
; Gestione movimenti manuali (JOG)
IF ifAvMan ;Attende l'ingresso di movimento manuale
IF NOT (gfMovAuto OR gfMovMan) ;Controlla che non ci siano movimenti automatici o manuali in corso
IF Asse:st_still ;Controlla che l'asse sia fermo
Asse:setvel=(slVelAsse*Asse:maxvel)/100 ;Imposta la velocità dell'asse (percentuale della velocità massima)
MANFW Asse ;Avanti asse in manuale
gfMovMan = 1 ;Segnala movimento manuale in corso
ENDIF
ENDIF
IF ifInMan ;Attende l'ingresso di movimento manuale
IF NOT (gfMovAuto OR gfMovMan) ;Controlla che non ci siano movimenti automatici o manuali in corso
IF Asse:st_still ;Controlla che l'asse sia fermo
Asse:setvel=(slVelAsse*Asse:maxvel)/100 ;Imposta la velocità dell'asse
MANBW Asse ;Avanti asse in manuale
gfMovMan = 1 ;Segnala movimento manuale in corso
ENDIF
ENDIF
IF gfMovMan ;Se l'asse si muove in manuale
IF NOT (ifAvMan OR ifInMan) ;Se gli ingressi di avanti e indietro manuale sono OFF
STOP Asse ;Ferma l'asse
gfMovMan = 0 ;Togli la segnalazione di asse in movimento manuale
ENDIF
ENDIF
;-----
; Operazioni finali
WAIT 1
JUMP MAIN
END

```

\*come calcolare measure e pulse è spiegato nel paragrafo apposito.

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.