

Sommario

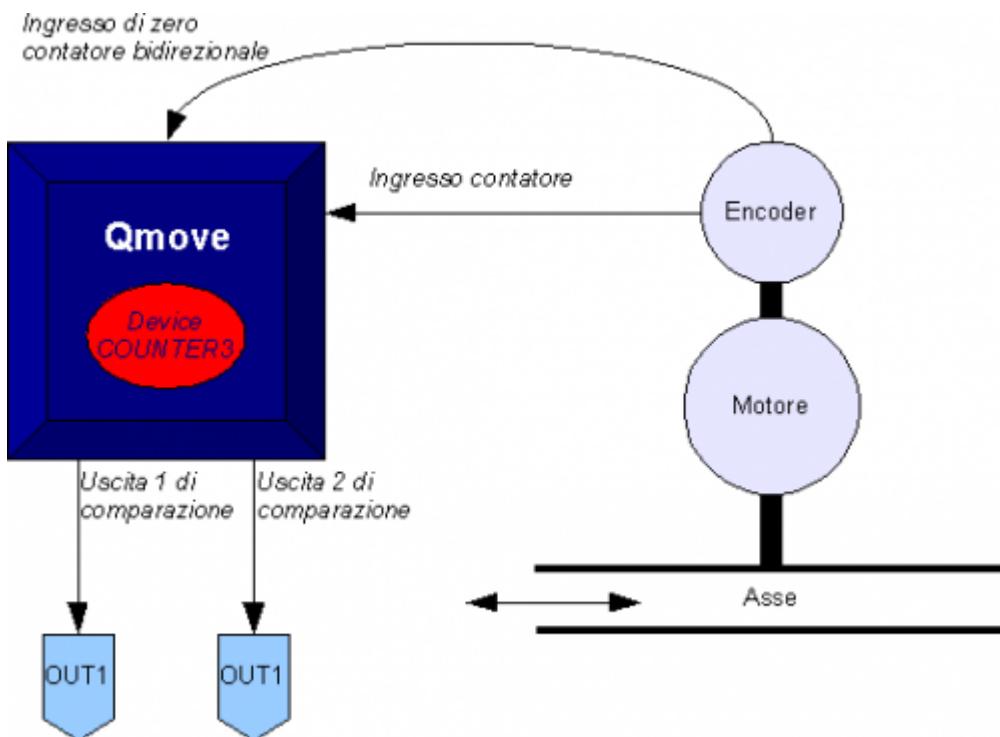
AN003- Esempio di utilizzo e calibrazione del device COUNTER3	3
Dichiarazione del device nella unit di configurazione	3
Corretta parametrizzazione del device	3
Sviluppo di un applicativo	4

AN003- Esempio di utilizzo e calibrazione del device COUNTER3

In questa sezione si vogliono descrivere le prime operazioni che si troverà a fare l'utente nel suo primo approccio con il device COUNTER3. Inoltre si vuole fornire un semplice esempio in cui si utilizzi il device COUNTER3.

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 COUNTER3, 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:

```

;----- Dichiaraione device interni -----
INTDEVICE
Asse COUNTER3 2 .CNT02 3 .OUT01 2 .OUT02
  
```

Si definisce un device COUNTER3 con nome "Asse" 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'accoppiamento del conteggio (numero 3) e infine gli indirizzi di due uscite utilizzate dai comparatori.

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 scaricato l'applicativo sullo strumento e avendolo fatto girare, 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.

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 trasduttore bidirezionale sia un encoder digitale. Supponiamo che l'encoder sia direttamente calettato su un motore che debba movimentare un asse. Sarà necessario impostare correttamente i parametri *measure* e *pulse* del device in modo che quest'ultimo possa interpretare gli impulsi encoder che arriveranno al QMove, lo strumento quindi potrà 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:

```
Asse:measure = 50;
Asse: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 e del prodotto QMove). 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:

- 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.

Sviluppo di un applicativo

Nella sezione precedente è stato spiegato quali sono i primi passi da seguire. In questa sezione è inserito 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](#)

```
; ****
; Unit Qcl
; ****

; Operazioni di parametrizzazione del device
; variabili utilizzate
; slSet1: Quota di comparazione per uscita 2.OUT01
; slSet2: Quota di comparazione per uscita 2.OUT02

Asse:measure = 1000
Asse:pulse = 1000          ;come calcolare measure e pulse è spiegato nel paragrafo apposito.
Asse:capture = 1           ;la posizione istantanea dell'asse viene catturata sul primo fronte di
                           ;discesa dopo l'attivazione di st_intenbl

IF slSet1 EQ 0
  slSet1 = 500
ENDIF
IF slSet2 EQ 0
  slSet2 = 100
ENDIF

; Funzione di homing abilitata ad ogni passaggio sul finecorsa
; variabili utilizzate
; slPrsPos : posizione di home impostata

MAIN:
  IF ifAbilZ
    INTENBL Asse          ;Attesa dell'ingresso di abilitazione impulso di zero trasduttore
  ELSE
    INTDSBL Asse          ;Abilita la cattura dell'impulso di zero trasduttore
  ENDIF
  IF Asse:st_capture
    ;Se è stata catturata la posizione istantanea
    IF ifAxeFermo
      Asse:delta = -(Asse:delta-slPrsPos) ;Calcolo quota da sommare al conteggio
      DELCNT Asse          ;Imposto il nuovo valore del conteggio
      RSCAPTURE Asse        ;Resetto st_capture dell'Asse
    ENDIF
  ENDIF

; Comparazioni sul conteggio solo se ifAbilComp attivo, altrimenti le uscite
; sono disattive
;

  IF ifAbilComp
    IF NOT gfApp01
      Asse:mode1 = 5        ;Attiva out1 se posit>setpoint1
      Asse:selout1 = 0       ;e disattiva out1 se posit<setpoint1
      Asse:setpoint1 = slSet1 ;impostazione del setpoint comparatore 1
      Asse:timer1 = 0        ;nessun ritardo nella commutazione dell'uscita
      Asse:mode2 = 6        ;Disattiva out2 se posit>setpoint2
      Asse:selout2 = 1       ;e attiva out2 se posit<setpoint2 comparatore 2
      Asse:setpoint2 = slSet2 ;impostazione del setpoint
```

```

Asse:timer2 = 0      ;nessun ritardo nella commutazione dell'uscita
gfApp01 = 1
gfApp02 = 0
ENDIF
ELSE
IF NOT gfApp02
Asse:mode1 = 0
RESOUT out201      ;Disattiva out1
Asse:selout1 = 0
Asse:mode2 = 0
RESOUT out202      ;Disattiva out2
Asse:selout2 = 1
gfApp01 = 0
gfApp02 = 1
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.