

\$title Posizionatore per EUROSITECH

\$version 2.0

\$drive mentor

\$author CT

\$COMPANY S.E. Impianti srl

```
// Le task sono eseguite su sei livelli di priorita'
// INITIAL      Livello 1 - a priorita' piu' alta
// EVENT        Livello 2
// ENCODER       Livello 3 (sincronizz. con funzioni interne al Mentor)
// CLOCK        Livello 4 (sincronizz. con funzioni interne al Mentor)
// BACKGROUND   Livello 5 - a priorita' piu' bassa
```

setup:{

```
////////// SETUP INIZIALE //////////
#15.06 = 100      // vel. in RPM/10

#16.07 =          400      // quota in decimi di mm / 1999 solo parte int
#16.06 =          400      // quota in decimi di mm - ( #16.07 * 1999 )

#15.10 =          2        // step encoder per mm solo parte intera
#15.09 =          0030     //step encoder per mm solo la parte decimale
#15.08 =          0

#16.08 = 20        // acceleraz. da 0 a vel max (#15.06*10) in millesimi se
#16.09 = 20 // deceleraz. da vel max (#15.06*10) a 0 in millesimi se #1
#15.07 = 100       // velocita' di JOG e ricerca HOME SW in RPM
#16.10 = 100       // offset in decimi di millimetro dopo zero
#15.14 =          0        // guadagno proporzionale di posizione
#15.13 =          0        // guadagno feed forward di posizione
#15.16 =          0
#15.15 =          0

#15.12 = 0         //step encoder per giro motore / 255 solo la parte inter
#15.11 = 0         //step encoder per giro motore - (#15.12*255)
#15.21 = 0         // quota di avanzamento in decimi di mm se zero ovvero i
#15.22 = 0         //inverte il senso di ricerca di zero
#15.23 = 0         // inverte il senso durante posizionamento
#15.24 = 1         // selezione scala tacc/tdec 0 = msec, 1 = decimi di sec
#15.25 = 0         // esegue l'azzeramento sul segnale zero dell'en
#15.36 =          1        // 1 = setup eseguito
////////// Salvare i parametri !!!! //////////
}
```

NOTES {

DESCRIZIONE DELL' APPLICAZIONE e SCHEMA CONNESSIONI I/O

Selezione Automatico / Manuale = Terminale TB3-27=0

Automatico ---> Terminale TB3-27 = 24 V

Manuale -----> Terminale TB3-27 = 0 V

Movimentazione JOG (contemporaneamente a Terminale TB3-27 = 0V)

Comando JOG CW (24 V su Terminale TB3-28)

Comando JOG CCW (24 V su Terminale TB3-29)

Posizionamento (necessita di 24 V su Terminale TB3-27 = Automatico)
 Comando di posizionamento = 24 V su Terminale TB3-28
 All'accensione, il primo comando di posizionamento corrisponde alla
 ricerca di zero (homing)
 Sensore di zero connesso a Terminale TB3-29

Impostazioni di default da inserire direttamente in Unidrive

#15.06 = 100 // vel. in RPM/10

se #15.21 = 0 la quota di avanzamento è data in decimi di mm

#16.07 = // quota in decimi di mm / 1999 solo parte int

#16.06 = // quota in decimi di mm - (#16.07 * 1999)

se #15.21 = 1 la quota di avanzamento è data in count encoder

#16.07 = // quota in conteggi encoder / 1999 solo parte

#16.06 = // quota in conteggi encoder - (#16.07 * 1999

#15.10 = // step encoder per mm * 10⁵ / 1999² solo pa

#15.09 = // [step encoder per mm * 10⁵] / 1999 - (#15

#15.08 = // step encoder per mm * 10⁵ - (#15.10 * 1999

#16.08 = 40 // acceleraz. da 0 a vel max (#15.06*10) in millesimi se

#16.09 = 40 // deceleraz. da vel max (#15.06*10) a 0 in millesimi se

#15.07 = 100 // velocita' di JOG e ricerca HOME SW in RPM

#16.10 = 100 // offset in decimi di millimetro dopo zero

#15.14 = // guadagno proporzionale di posizione / 255 sol

#15.13 = // guadagno proporzionale di posizione - #15.14

#15.16 = // guadagno feed forward di posizione / 255 solo

#15.15 = // guadagno feed forward di posizione - #15.16 *

#15.12 =0 //step encoder per giro motore / 255 solo la parte inter

#15.11 =0 //step encoder per giro motore - (#15.12*255)

#15.22 = 1 //inverte il senso di ricerca di zero

#15.23 = 1 // inverte il senso durante posiziamiento

#15.24 = 0 // selezione scala tacc/tdec 0 = msec, 1 = decim

#15.25 = 0 // esegue l'azzeramento sul segnale zero dell'en

#15.36 = // 1 = setup eseguito

}

////////// INITIAL //
 // La task initial e' eseguita una sola volta all' accensione o dopo un reset
 // e' utilizzabile per l' inizializzazione delle variabili

initial {

//// Impostazioni MENTOR /////

// #1.10=1 // riferimento bipolare

#1.18=0 //azzeramento riferimento velocita' digitale

#1.14=1 // seleziona riferimento velocita' digitale

#1.15=0 // seleziona #1.18 come riferimento di velocita'

// #6.04=3 // Sequencing mode=PLC mode: RUN quando sequencing bit <> 0

// #6.30=1 // Seleziona Sequencing bit 0 per RUN

// #2.02=0 // Disabilita rampe unidrive

//#3.10=200 // Kp vel - guadagno proporzionale anello di velocita'

//#3.11=100 // Ki vel - guadagno integrale anello di velocita'

```

// #3.12=50      // Kd vel - guadagno derivativo anello di velocita'

// #8.27=1        // logica positiva

#14.12 = 1      //Abilitazione controllo di posizione
#14.14 = 1      //
#91.01 = 0
_Q20%.7=1
pippo%=2

//////////      Impostazioni MD29      //////////

_Q20%.1=1      // abilita loop di posizione
_Q20%.6=1      // abilita trasferimento del rif.di vel. in #1.21
_Q16%=163840   // Nessuna limitazione sul segnale in uscita
_Q5%=10000     // Kp pos - guadagno proporzionale di posizione
_Q6%=0         // Ki pos - guadagno integrale di posizione
_Q20%.0=0      // Seleziona _Q7% come guadagno FeedForward
_Q7%=1000      // Kff pos - guadagno feed fforward di posizione
_Q12%=1000000  // Accelerazione (max 1638400)
_Q13%=1000000  // Decelerazione (max 1638400)
_Q14%=100000   // Vel. massima in count/secondo (max 163840)
_Q32%.2=0      // Albero elettrico disabilitato
_Q3%=0         // Azzerà riferimento avanzamento in velocita'
_Q32%.3=1      // Seleziona blocco avanz. in velocita'
_Q32%.4=0      // Disabilita funzione di CAMMA elettronica

_Q32%.8=0      // Rampe trapezoidali per _Q2% e per _Q26%
_Q32%.19=0     // Rampe trapezoidali per _Q3% e per aggancio in alb. el.
_Q32%.20=0     // Rampe trapezoidali per _Q27%

_Q11%=100000000 // Rapporto albero elettrico 1:1
_Q32%.1=0      // Inseguimento albero elettrico in senso concorde
_Q40%=0        // Pos. di ritorno dopo sgancio albero el.in pos. (_Q32%.3=0)
_Q26%=0        // Offset di posizione
_Q27%=0        // Offset di velocita'
_Q15%=200000   // Vel. di recupero durante aggancio in fase (max 2147000000)

_Q32%.17=1     // Reset contatori

//////////      Configurazione I/O digitali dell' Unidrive      //////////
Call Set_up_IO:
IF #15.36 = 0 THEN CALL setup:
//////////      inizializzazione variabili      //////////
//// assegnazione mnemonici alle variabili      ////
$define Accel% _Q12%
$define Decel% _Q13%
$define Max_Speed% _Q14%
$define Final_Pos% _Q2%

// definisce terminale TB3-27 come automatico (se=1) o manuale (se=0)
$define Automatico% 1
$define Manuale% 0
$define Automatico_Manuale% #15.34

// definisce terminale TB3-28 come JOG CW (in manuale) o comando posiz. (in auto

```

```

$define JOG_CW% #15.35
$define Start_Mot% #15.35

// definisce terminale TB3-29 come JOG CCW (in manuale)
// definisce terminale TB3-29 ingressomicrointerruttore per la ricerca di zero
$define JOG_CCW% #15.36
$define Home_sw% #15.36

// definisce terminale TB2-17 come uscita IN POSIZIONE (1)/ MOTORE IN MOTO (0)
$define In_Posiz_OUT% #15.31
$define Posizionato% 1
$define In_Moto% 0

Home_ok%=1 // ricerca i zero non effettuata
Encoder_step%=140 // step encoder per giro
Max_RPM_Speed%=1400 // massima velocita' motore in RPM
Max_Speed_in_count%=800000 // massima velocita' in counts/secondo
Jog_Speed_in_count%=100000 // velocita' per _Q3%
Pos_Accel%=1000000 // accelerazione in count/sec2
Pos_Decel%=1000000 // decelerazione in count/sec2
Numerator_coeff_mm_count%=1 // numeratore coeff. conversione mm in count
Denom_coeff_mm_count%=1 // denominatore coeff. convers. mm in count
Va_a_Pos%=0 // posizione da raggiungere
Fronte_Start_Mot%=0 // =1 su fronte basso alto di #16.35
Non_Trovato%=0
Trovato%=0
Direction_flag%=0
Conta_inc%=0

////////// Calcolo coeff. conversione per quote //////////
// Step_per_mm = float(#15.10*1999*1999+#15.09*1999+#15.08)/100000
//Step_per_mm = float(#15.10)*19.99*1.999+(float(#15.09)*1999+float(#15.08))/100
//Count_per_mm=Step_per_mm*4
//Count_per_mm=16.128
Divider%=10000
Step_per_mm=(FLOAT(#15.10)*Divider%+#15.09)/Divider%
Count_per_mm=Step_per_mm*4
}

////////////////////////////////////// BACKGROUND ////////////////////////////////////////
// La task background deve essere utilizzata per l' esecuzione di funzioni
// e comandi non necessariamente legati al tempo
// Viene eseguita dopo la task initial e quando nessuna task
// legata al tempo e' in esecuzione
// Solitamente la task background e' la task di supervisione del processo
// e deve essere eseguita ciclicamente. L' istruzione GOTO all' etichetta
// iniziale e' necessaria per garantire la ciclicita'

background {
Top:

// guadagni loop di posizione
//_Q5%=#15.14
//_Q7%=#15.13

// impulsi/giro encoder
//Encoder_step%=#15.12+#15.11

```

```

        // Calcolo velocita' massima #15.06 espresso in RPM/10
        Max_Speed%=int((float(#15.06)*4*10/60)*Encoder_

// Calcolo accelerazione
IF #15.24=0 THEN
    Accel%=(Max_Speed%/#16.08)*1000
    Decel%=(Max_Speed%/#16.09)*1000
ELSE
    Accel%=(Max_Speed%/#16.08)*10
    Decel%=(Max_Speed%/#16.09)*10
ENDIF

// Calcolo quota
Va_a_Pos%=#16.07*1999+#16.06

IF Automatico_Manuale%=Manuale% THEN
    Call jog:
ELSE
    /// se selezionata funzione Automatica ///
    _Q32%.3=0
    #1.15=0

    /// attivazione comando di posizionamento sul fronte ///
    IF Start_Mot%=0 THEN
        trigg_Start_Mot%=1
    ELSE
        Fronte_Start_Mot%=trigg_Start_Mot% & Start_Mot%
        trigg_Start_Mot%=0
    ENDIF

    IF Fronte_Start_Mot%=1 THEN
        Fronte_Start_Mot%=0
        IF Home_ok%=0 THEN
            Call homing:
        ELSE
            // inizializza variabili di posizionamento //
            // Calcolo quota di incremento
            IF #15.21 = 0 THEN
                _Q2%=_Q8%
                _Q4%=_Q8%
                _Q9%=_Q8%
                _Q0%=_Q8%
                _Q1%=_Q8%
                _Q10%=_Q8%
                Inc_Pos%=0
                Final_Pos%=0
                DELAY(3)
                Inc_Pos%=INT((Va_a_Pos%*Count_per_mm/10)+93)
            ELSE
                Inc_Pos%=Va_a_Pos%
            ENDIF
        ENDIF

        IF #15.23=0 THEN

```

```

                                Final_Pos%=Final_Pos%+Inc_Pos%
ELSE
    Final_Pos%=Final_Pos%-Inc_Pos%
ENDIF

    Call posiziona:

ENDIF

ENDIF

ENDIF

GOTO Top:
}

////////////////// CLOCK ////////////////////
// La task clock e' eseguita a tempo costante con cadenza pogrammata
// nel parametro #14.04 con risoluzione 1 mS
// #14.04 = 1    (valore minimo) ---> clock eseguita ogni 1 mS
// #14.04 = 100 (valore massimo) ---> clock eseguita ogni 100 mS
// La task clock interrompe la task background
clock {
    IF _Q32%.3=1 THEN
        _Q2%=_Q8%
        _Q4%=_Q8%
        _Q9%=_Q8%
        _Q0%=_Q8%
        _Q1%=_Q8%
        _Q10%=_Q8%
    ENDIF
}

////////////////// ENCODER ////////////////////
// La task encoder e' eseguita a tempo costante con cadenza dipendente
// dalla Frequenza di Switching sel. nel Drive in para #0.41 (copia del #5.18).
// Task Encoder eseguita ogni 5.52 mS con #0.41= 3 o 6 o 12
// Task Encoder eseguita ogni 7.84 mS con #0.41= 4.5 o 9
// La task encoder interrompe le task clock e background
//encoder {
//}

////////////////// EVENT ////////////////////
// La task Event e' eseguita esclusivamente a seguito dell' intervento
// di un evento specifico (event flag #85.02 bit 0=1) tra i possibili due:
// 1) Il Timer/Counter hardware ha riciclato
//     se #85.01 bit 0 =1
// 2) L' ingresso TTL 0 e' attivato. Questo ingresso e' connesso al pin 4
//     del connettore della seriale RS485. L' immagine del suo stato e'
//     il parametro #86.01
//     #86.01=0 con ingresso non connesso o a +5V
//     #86.01=1 con ingresso a massa
//
// La programmazione del Timer counter avviene tramite i parametri
// #85.01, #85.02, #85.03, #85.04, #85.05
// L' event flag #85.02 bit 0 e' azzerato leggendo lo Status Register #85.02
//event {
//}

```

```
////////// USER SUBROUTINES //////////
```

```
Set_up_IO: {
////////// CONFIGURAZIONE I/O DIGITALI //////////
// Uscita terminale TB2-17 #09.03
// Uscita terminale TB2-18 #09.04
// Uscita terminale TB2-19 #09.05
// Ingresso terminale TB3-27 #08.07
// Ingresso terminale TB3-28 #08.08
// Ingresso terminale TB3-29 #08.09
#9.19=1531 // Mappatura Source Terminale TB2-17
#9.21=1532 // Mappatura Source Terminale TB2-18
#9.23=1533 // Mappatura Source Terminale TB2-19
#8.17=1534 // Mappatura Destination Terminale TB3-27
#8.18=1535 // Mappatura Destination Terminale TB3-28
#8.19=1536 // Mappatura Destination Terminale TB3-29
////////// FINE CONFIGURAZIONE I/O DIGITALI //////////
}
```

```
////////// ricerca di zero //////////
homing: {
_Q32%.3=1
// Calcolo velocita' di ricerca dello zero #15.07 espresso in RPM
Jog_Speed_in_count%=int((float(#15.07)*4/60)*Encoder_step%)
Home_trovato%=Non_Trovato%
// Se richiesta inversione del senso di rotazione su tutte le movimentazioni //
IF #15.23=1 THEN
Jog_Speed_in_count%=-Jog_Speed_in_count%
ENDIF
//// Se richiesta inversione del senso di rotazione nella ricerca di zero ///
IF #15.22=1 THEN
Jog_Speed_in_count%=-Jog_Speed_in_count%
ENDIF
```

```
_Q3% = - Jog_Speed_in_count% // carica velocita' di ricerca index
```

```
////////// Attesa ingaggio switch di zero //////////
DO WHILE Home_sw%=0
Motore%=In_Moto% // out "in posizione"=0
LOOP
_Q3% = Jog_Speed_in_count% / 10 // carica velocita' di ricerca lenta
```

```
////////// Attesa di uscita dallo switch di zero //////////
DO WHILE Home_sw%=1
Motore%=In_Moto% // out "in posizione"=0
LOOP
//_Q14%=ABS(_Q3%) // serve per posiz. su index se richiesto
_Q3%=0
IF #15.25=1 THEN
```

```
////////// Gestione arresto su index //////////
```

```
_Q3% = Jog_Speed_in_count% / 10 // carica velocita' di ricerca lenta
_Q20%.3=1 // abilita lettura index
DO WHILE _Q20%.3 <> 0
```

```

a%=0
LOOP
_Q32%.3=0
// _Q14%=_Q3%
_Q3%=0
_Q2%=_Q17% // va alla quota congelata
DO WHILE _Q2% <> _Q9%
a%=0
LOOP

ENDIF
//////////////////// FINE GESTIONE ARRESTO SU INDEX //////////////////////
delay(1)
_Q32%.3=0
Inc_Pos%=int(float(#16.10)*Count_per_mm/10)

// Calcolo velocita' massima #15.06 espresso in RPM/10
// Max_Speed%=int((float(#15.06)*4*10/60)*Encoder_step%)

// Calcolo accelerazione
IF #15.24=0 THEN
Accel%=(Max_Speed%/#16.08)*1000
Decel%=(Max_Speed%/#16.09)*1000
ELSE
Accel%=(Max_Speed%/#16.08)*10
Decel%=(Max_Speed%/#16.09)*10
ENDIF

Final_Pos%=Final_Pos%+Inc_Pos%

Call posiziona:

Call set_origin:

In_Posiz_OUT%=Posizionato% // out "in posizione" alto
}

//===== S E T O R I G I N =====
set_origin:{
Home_ok%=1
Home_trovato% = Trovato%
_Q7%=0
delay(1)
_Q32%.17=1 // Reset contatori
DO WHILE _Q2% <> 0
a%=0
LOOP
//////////////////// _Q7%=#18.20
}

//////// movimentazione manuale jog //////////
jog: {
_Q32%.3=1 // selezione controllo in velocita'
Jog_Speed_in_count%=int((float(#15.07)*4/60)*Encoder_step%)
DO WHILE Automatico_Manuale%=Manuale%
IF (JOG_CW% & JOG_CCW%)=1 THEN

```

```

    /// se CW e CCW entrambi selezionati arresta movimento ///
    _Q3%=0 // azzera velocita'
    In_Posiz_OUT%=Posizionato% // out "in posizione" alto
    #1.15=0
ELSE
    IF (JOG_CW% | JOG_CCW%) = 0 THEN
        _Q3%=0 // azzera velocita'
        In_Posiz_OUT%=Posizionato% // out "in posizione" alto
        #1.15=0
    ELSE
        IF JOG_CW% <> 0 THEN
            #1.15=1
            #1.20=#15.07
            In_Posiz_OUT%=In_Moto% // out "in posizione"=1
        ELSE
            #1.15=1
            #1.20=-#15.07
            In_Posiz_OUT%=In_Moto% // out "in posizione"=1
        ENDIF
    ENDIF
ENDIF
LOOP
_Q3%=0 // azzera velocita'
}

////////// posizionamento //////////
posiziona: {
In_Posiz_OUT%=In_Moto% // uscita di "in posizione" a livello basso
DO WHILE Final_Pos%<> _Q9%
    // #15.08=INT(_Q1%/30)
    In_Posiz_OUT%=In_Moto% // uscita di "in posizione" a livello basso
LOOP
In_Posiz_OUT%=Posizionato% // uscita di "in posizione" a livello alto
    // #15.08=INT(_Q1%/30)
}

```

