

Iso Virtual  
Applicazione VTB per IsoNs  
[www.promax.it](http://www.promax.it)



Le informazioni contenute nel manuale sono solo a scopo informativo e possono subire variazioni senza preavviso e non devono essere intese con alcun impegno da parte di Promax srl. Promax srl non si assume nessuna responsabilità od obblighi per errori o imprecisioni che possono essere riscontrate in questo manuale. Eccetto quanto concesso dalla licenza, nessuna parte di questa pubblicazione può essere riprodotta, memorizzata in un sistema di archiviazione o trasmessa in qualsiasi forma o con qualsiasi mezzo, elettronico, meccanico, di registrazione o altrimenti senza previa autorizzazione di Promax srl. Qualsiasi riferimento a nomi di società e loro prodotti è a scopo puramente dimostrativo e non allude ad alcuna organizzazione reale.

Rev. 2.0.2 © Promax srl

## 1 PREFAZIONE

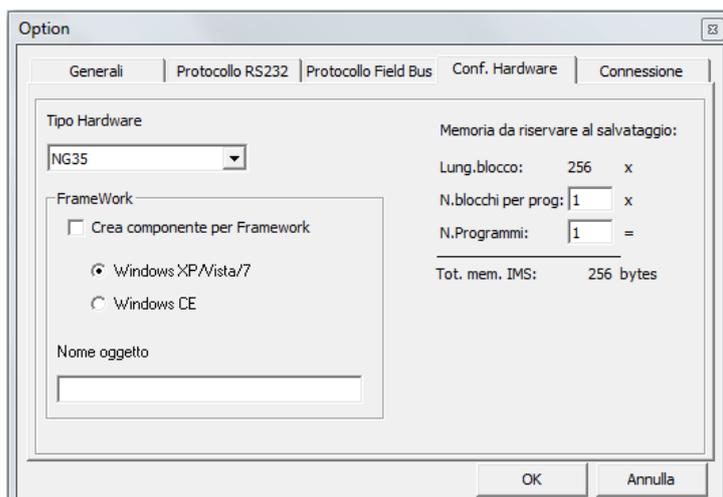
Questo manuale descrive l' applicazione VTB per ISONS.

## 2 COMPONENTI BASE

Per iniziare un applicazione VTB-ISONS è necessario inserire i seguenti componenti base:

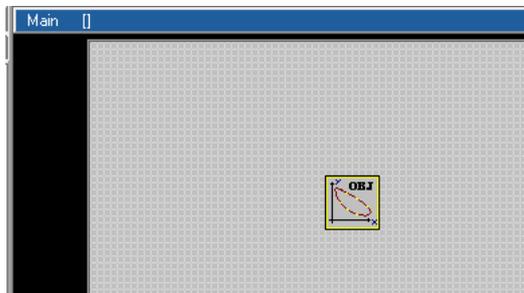
### 2.1 Selezione Hardware

*Strumenti → Opzioni → Conf. Hardware (Es NG35)*



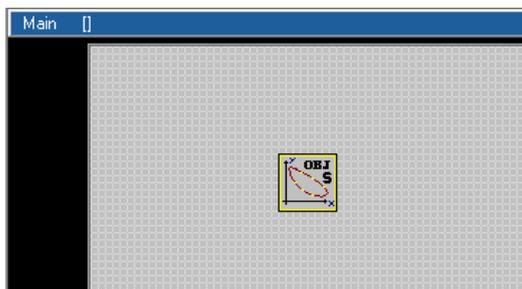
### 2.2 Inserire Oggetto *cobjinterpola*

*Oggetti → Motor Control → cobjinterpola.vco*



Si può inserire l' oggetto Interpola con Rampe ad S.

In questo modo si rende attivo il parametro di ISONS **JERK** per gestire le accelerazioni con rampa ad S.



### 2.3 Proprietà dell' oggetto *cobjinterpola*

1. **nome** → objinterpola1 (default)
2. **N. Assi** → Numero di assi interpolati per il processo
3. **N.tratti** → Profondità Look Ahead Default 16
4. **Vper** → Non Cambiare
5. **Div. Vper** → Non Cambiare
6. **Abilita Arcto** → Non Cambiare

Interpola1	
Property	Events
Property	Value
Nome	Interpola1
Left	130
Top	75
N.assi	3
N.tratti	16
Vper	1024
Div. Vper	1024
Abilita arcto	1

Massima profondità di look ahead (Il valore della profondità deve essere un multiplo binario es: 8 – 16 – 32 etc.)

#### NGQuark

- 2 Assi → 32
- 3 Assi → 16
- 4 Assi → 8

#### NGMEVO-NGMEVO

- 2 Assi → 128
- 3 Assi → 128
- 4 Assi → 128

#### NG35

- Tutti → 2048

### 2.4 Inserire Oggetto Iso Virtual

*Oggetti → IsoNs → - ISOVirtual - \$Rev 2.3.0 (ISOVirtual - \$Rev 1.1.9 è Obsoleto)*

L' oggetto ISOVirtual light è usato per i sistemi **NGQ or NGQx**  
 Questo componente utilizza meno memoria.



### 2.5 Proprietà di IsoVirtual

1. **nome** → ISOV1 (default)
2. **Indice processo** → Non Cambiare
3. **Parametri Custom** → Vedi in seguito

ISOV1	
Property	Events
Property	Value
Nome	ISOV1
Left	95
Top	35
Indice processo	1
Parametri custom	0

## 2.6 Inserire la Variabile Fixed0

Questa variabile è usata per sincronizzare il processo ISONS PC

Oggetti → Fixed Var

VAR Interne	VAR Bit	Define	VAR Static	VAR VSD	VAR Fixed
Fixed0		LONG	EXP	<input type="checkbox"/>	
Addr	Variabile	Tipo			

Seleziona il primo indirizzo (click nella posizione 0)

Addr	Variabile	Tipo
0		
1		

Premere il pulsante Aggiungi

Addr	Variabile	Tipo
0	Fixed0	LONG
1	*****	*****
2	*****	*****
3	*****	*****

## 3 Inserire il tipo di Assi

Dopo aver inserito i componenti base è possibile inserire il tipo di Assi da controllare.

### 3.1 Scelta del tipo di Assi

Oggetti → IsoNs → IsoCanOpen

Tipo Assi

**IsoCanOpen**

Questa contiene tutti gli assi di tipo CanOpen

Settaggio tipico

AX1	
Property	Events
Property	Value
Nome	AX1
Left	90
Top	175
Nodo	1
Indice asse ISO	0
Nome processo	ISOV1
Nome quota pdo	qi
Mask Allarmi	0xFFFFFFFF
Mask Parametri	0x00

- Nome → Nome Asse
- Nodo → Nodo CanOpen
- Indice Asse Iso → Indica il numero di Asse ISO (0 -X , 1-Y , 2 – Z ecc.)
- Nome Processo → Nome del Processo ISOVIRTUAL Associato
- Nome quota pdo → Nome del PDO CanOpen per quota interpolata

### IsoPid.vco

Utilizzato per Drives Analogici +/- 10V con retrazione da encoder, oppure per drives STEP/DIR con loop da encoder ESTERNO

#### IsoPid- NG35 Filtro Digitale



Per drives +/-10 V con retrazione da encoder solo per NG35 e espansione NGIO

#### Proprietà

- Nome → Nome Oggetto IsoPid
  - Indice Asse Iso → Indica il numero di Asse ISO (0 -X , 1-Y , 2 – Z ecc.)
  - Nome Processo → Nome del Processo ISOVIRTUAL Associato
  - Indice Asse NGIO → ndice canale NGIO dove è collegato l' asse
    - 0 → Ch0 prima NGIO
    - 1 → Ch1 prima NGIO
    - 2 → Ch0 Seconda NGIO
    - 3 → Ch1 Seconda NGIO ecc.
- 
- Kp,Ki,Kv,
  - Err Saturazione
  - Divisore,Dir → Parametri PID Settati dall' applicazione ISONS
  - Enable Kp,Ki,Kd → Non Cambiare (True)
  - T0 Level → Livello tacca di ZERO encoder
    - 0 → Bassa
    - 1 → Alta

- Soglia ServoErr
- Ritardo ServoErr → Soglia Servo error e ritardo,Parametri Settati dall' applicazione ISONS

#### NGPP- NG-PP + PID



Usato per STEPPER MOTOR con chiusura anello di spazio da encoder esterno



**Proprietà**

Nome	→	Nome Oggetto IsoPid
Indice Asse Iso	→	Indica il numero di Asse ISO (0 -X , 1-Y , 2 – Z ecc.)
Nome Processo	→	Nome del Processo ISOVIRTUAL Associato
Indice Asse NGPP	→	Indica il canale NGPP dove è connesso il driverr
		0 → Ch0 prima NGPP
		1 → Ch1 prima NGPP
		2 → Ch2 prima NGPP
		3 → Ch3 prima NGPP
		4 → Ch0 Seconda NGPP
		5 → Ch1 Seconda NGPP ecc
Indice Asse NGIO	→	Indice canale NGIO dove è collegato l' asse
		0 → Ch0 prima NGIO
		1 → Ch1 prima NGIO
		2 → Ch0 Seconda NGIO
		3 → Ch1 Seconda NGIO ecc.
Kp,Ki,Kv, Err Saturazione Divisore,Dir Enable Kp,Ki,Kd T0 Level	→	Parametri PID Settati dall' applicazione ISONS
	→	Non Cambiare (True)
	→	Livello tacca di ZERO encoder
		0 → Bassa
		1 → Alta
Soglia ServoErr Ritardo ServoErr Max Freq Scalav Enable Out	→	Soglia Servo error e ritardo,Parametri Settati dall' applicazione ISONS
	→	Reserved

**IsoDouble\_Enc.vco**

Contiene gli oggetti CanOpen con Loop da encoder ESTERNO.

**CanOpen + PID**

Used for external closed loop in CanOpen ESTUN DRIVES with interpolation mode. Normally this Oggetto is used for high precision Assi



**Proprietà**

Nome	→	Nome Oggetto
Nodo	→	Nodo CanOpen del Driver
Indice Asse Iso	→	Indica il numero di Asse ISO (0 -X , 1-Y , 2 – Z ecc.)
Nome Processo	→	Nome del Processo ISOVIRTUAL Associato
Nome quota pdo	→	Nome del PDO CanOpen per quota interpolata
Indice Asse NGIO	→	Indice canale NGIO dove è collegato l' asse
		0 → Ch0 prima NGIO
		1 → Ch1 prima NGIO
		2 → Ch0 Seconda NGIO
		3 → Ch1 Seconda NGIO ecc.
Kp,Ki,Kv, Err Saturazione		
Divisore,Dir	→	Parametri PID Settati dall' applicazione ISONS
Enable Kp,Ki,Kd	→	Non Cambiare (True)
T0 Level	→	Livello tacca di ZERO encoder
		0 → Bassa
		1 → Alta
Soglia ServoErr		
Ritardo ServoErr	→	Soglia Servo error e ritardo - Parametri Settati dall' applicazione ISONS
Home Delay	→	Ritardo in Milliscondi per Homing (tipico 1000)

**IsoPP.vco**

Contiene assi STEP DIR ad anello aperto

**IsoPP NGMEVO Passo-Passo**

Usato per NGMEVO con STEP DIR Assi



**Proprietà**

Nome	→	Nome Oggetto
Indice Asse Iso	→	Indica il numero di Asse ISO (0 -X , 1-Y , 2 – Z ecc.)
Nome Processo	→	Nome del Processo ISOVIRTUAL Associato
NGMEVO Channel	→	Indica il canale NGMEVO dove è connesso il driver
		0 → Ch0
		1 → Ch1
		2 → Ch2
		3 → Ch3

**IsoPP NG35 (NG-PP) Passo-Passo**

Usato per NG35+NGPP con STEP DIR Assi



**Proprietà**

Nome	→	Nome Oggetto
Indice Asse Iso	→	Indica il numero di Asse ISO (0 -X , 1-Y , 2 – Z ecc.)
Nome Processo	→	Nome del Processo ISOVIRTUAL Associato
NG-PP Channel	→	Indica il canale NGPP dove è connesso il driver
		0 → Ch0 prima NGPP
		1 → Ch1 prima NGPP
		2 → Ch2 prima NGPP
		3 → Ch3 prima NGPP
		4 → Ch0 seconda NGPP
		5 → Ch1 seconda NGPP ecc.

### IsoPP\_slave.vco

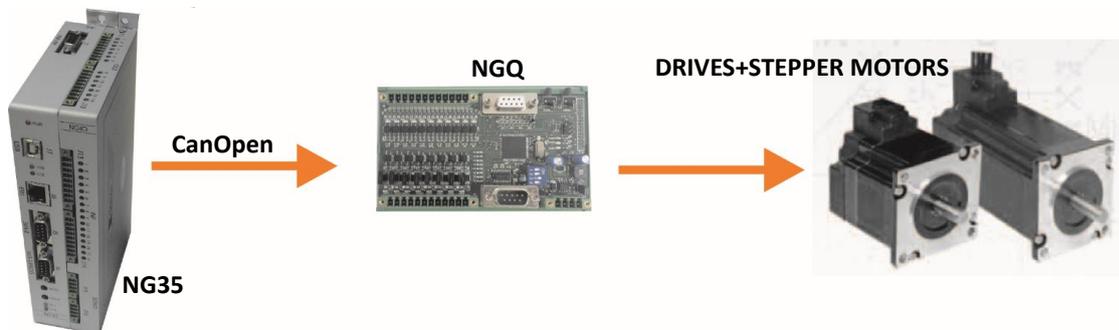
Contiene Assi STEP DIR ad anello aperto su schede Slave CanOpen, controllate da un MASTER

Es:

NG35 → Master  
 NGQ → Slave CanOpen con step Assi

#### IsoPP\_slave NGMEVO Slave

Usata per NGQ slave con controllo assi STEP DIR



#### Proprietà

Nome → Nome Oggetto  
 Indice Asse Iso → Indica il numero di Asse ISO (0 -X , 1-Y , 2 – Z ecc.)  
 Nome Processo → Nome del Processo ISOVIRTUAL Associato  
 NGMEVO Channel → Indica il canale NGQ dove è connesso il driver  
     0 → Ch0  
     1 → Ch1  
     2 → Ch2  
     3 → Ch3  
 Nodo → Nodo CanOpen NGMEVO o NGQ  
 Nome quota pdo → Nome del PDO CanOpen per quota interpolata

## 4 Inserire I/O

Quando tutti gli ASSI sono stati inseriti, si possono aggiungere dei moduli I/O per il controllo del ciclo PLC.

### 4.1 Scelta del tipo di I/O

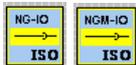
Oggetti → IsoNs → IsoCanOpen

#### Iso-IO.vco

Contiene tutti i moduli di I/O

##### Iso-IO – NG-IO - NGM-IO

I/O a bordo di NGIO -NGMIO (Espansioni NG35 -NGMEVO)

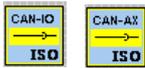


#### Proprietà

Nome → Nome Oggetto  
 Nome Processo → Nome del Processo ISOVIRTUAL Associato  
 Indice ISO-IO (16 bit) → Numero di gruppo di appartenenza da 0 a 15  
     0 → Gruppo 1 - I/O da 0 a 15  
     1 → Gruppo 2 - I/O da 16 a 31  
     3 → Gruppo 3 – I/O da 32 a 47 ecc.  
 Indice NG-IO → Indica il numero di NGIO o NGMIO (partendo da 0)  
     0 → Prima NGIO-NGMIO nel bus locale  
     1 → Seconda NGIO-NGMIO nel bus locale  
     2 → Terza NGIO-NGMIO nel bus locale ecc.  
 Hardware enable → Non Cambiare (true)

**Iso-IO - CAN-IO – CAN-AX**

I/O on the NGQ (CAN-IO) o NGQx(CAN-AX) slave CanOpen



**Proprietà**

Nome	→	Nome Oggetto
Nodo	→	Nodo CanOpen NGQ - NGQx -NGMEVOs
Allarme cfg	→	Numero di allarme generato Quando la scheda è in errore - default 41 Se si hanno più schede NGQ,NGQx,NGMEVOs , incrementare questo valore (42,43 etc)
Nome Processo	→	Nome del Processo ISOVIRTUAL Associato
Indice ISO-IO (16 bit)	→	Numero di gruppo di appartenenza da 0 a 15 0       →     Gruppo 1 - I/O da 0 a 15 1       →     Gruppo 2 - I/O da 16 a 31 3       →     Gruppo 3 – I/O da 32 a 47 ecc.
Hardware enable	→	Non Cambiare (true)
Variabile Inp	→	Nome del PDO ingressi digitali definito dal configuratore
Variabile Out	→	Nome del PDO uscite digitali definito dal configuratore

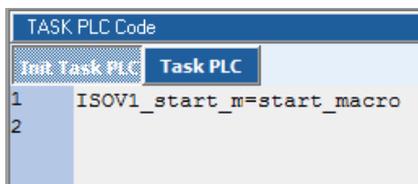
**CAN-PPN,- EMC-IO Da non usare obsolete**

**5 Utilizzo delle funzioni M interne al CNC**

In questo esempio in nome dell' oggetto ISO VIRTUAL è ISOV1

**5.1 Inizializzare in VTB l' entry point alla funzione ISOV1\_start\_m**

Inserire il codice in INIT TASK PLC



Start\_macro è una funzione nella Main → Funzioni di Pagina

Questa funzione è chiamata automaticamente Quando la funzione **M** è invocata dal PC

-----  
 ' **Funzione start macro**  
 -----

```
function start_macro() as void
ISOV1_M_ack=1 ' acknoleged per ISONS PC
select ISOV1_M_cmd
  case 1
    ' Funzione M1
  case 2
    ' Funzione M2
  case 3
    ' Funzione M3
  case else
    ISOV1_M_ack=0 ' M non riconosciuta
endselect
endfunction
```

## 5.2 Leggere i parametri della funzione M

IsoNs può scrivere i parametri alla funzione M da GCODEe:

### IsoNs Codice

```
$_PARAM1=100
$_PARAM2=130
M5
```

Si può leggere da applicazione VTB i parametri dalle seguenti variabili:

```
ISOV1_M_1 → $_PARAM1 ISOV1_M_2 → $_PARAM2
ISOV1_M_3 → $_PARAM3 ISOV1_M_4 → $_PARAM4
ISOV1_M_5 → $_PARAM5 ISOV1_M_6 → $_PARAM6
ISOV1_M_7 → $_PARAM7 ISOV1_M_8 → $_PARAM8
ISOV1_M_9 → $_PARAM9 ISOV1_M_10 → $_PARAM10
```

### VTB Codice

```
function start_macro() as void
    ISOV1_M_ack=1 'acknoleged per ISONS PC
    select ISOV1_M_cmd
        case 5
            ' Funzione M5
            if ISOV1_M_1=100 && ISOV1_M_2=130
                ....
            endif
            ISOV1_status_m_run=0 ' Libera l' applicazione IsoNs
        case else
            ISOV1_M_ack=0 ' M non riconosciuta
    endselect
endfunction
```

## 5.3 Scrittura dei Parametri M da Applicazione VTB

Dall' applicazione VTB è possibile scrivere i parametri per l' applicazione IsoNs, dopo la chiamata alla funzione M

### IsoNs Codice

```
M5
IF $_PARAM1=100
...
...
END_IF
```

### VTB Codice

```
function start_macro() as void
    ISOV1_M_ack=1 'acknoleged per ISONS PC
    select ISOV1_M_cmd
        case 5
            ' Funzione M5
            ISOV1_M_1=100
            ISOV1_status_m_run=0 ' Libera l' applicazione IsoNs
        case else
            ISOV1_M_ack=0 ' M non riconosciuta
    endselect
endfunction
```

## 5.4 M flags

Nell' applicazione VTB, le funzioni M utilizzano i seguenti flags:

<b>ISOV1_M_ack</b>	Se scritto a <b>True</b> indica a IsoNs PC che la funzione M è stata processata Se scritto a <b>False</b> indica che la funzione M non è stata trovata – <b>Error</b>
<b>ISOV1_M_cmd</b>	Contiene il numero di M chiamata da IsoNs PC
<b>ISOV1_status_m_run</b>	Il part program di IsoNs attende che questo flag torni a <b>False</b> prima di proseguire Un valore a <b>True</b> blocca il PartProgram <b>GCODE</b>
<b>ISOV1_status_m_stop</b>	Questo flag viene scritti a <b>True</b> dall' applicazione IsoNs Quando deve essere effettuata una <b>RICHIESTA DI STOP</b> (es. Stop da pulsante) L' applicazione VTB deve <b>INTERROMPERE</b> tutti i cicli M in esecuzione e resettare questo flag.

## 5.5 Esempio M3 M4 M5 start/stop Mandrino

L' applicazione IsoNs può scrivere la velocità Mandrino (Funzione S) automaticamente in una variabile VTB **ISOV1\_generic(9)** (solo se il parametro IsoNs **WR\_SPD9=1**). Quando in Gcode si esegue la funzione **Svalore** automaticamente il valore è scritto in **ISOV1\_generic(9)**. Il prossimo esempio si riferisce ad una Scheda NG35 con NGIO per uscita analogica del Mandrino

**Nota:**

<b>MAXDAC</b>	è una define di VTB 2047 (Divisioni DAC)
<b>MAXSPEEDSPINDLE</b>	è una define di VTB – Numero giri max del Mandrino es: 24000
<b>CwOut</b>	è un bit di un uscita digitale che setta la rotazione Oraria del Mandrino
<b>CCwOut</b>	è un bit di un uscita digitale che setta la rotazione AntiOraria del Mandrino
<b>SpindleStart</b>	è un bit di un uscita digitale che AVVIA il Mandrino
<b>VelSpindle</b>	è una variabile Long VTB

**IsoNs Codice**

```
S1200 // 1200 rpm
M3 // start Mandrino in CW mode
```

**VTB Codice**

```
function Start_Macro() as char
ISOV1_m_ACK=1
select ISOV1_M_cmd
  case 3 ' start Mandrino in CW
    Cw=true ' set Cw mode
    Ccw=false
    VelSpindle=(ISOV1_generic(9)*MAXDAC)/MAXSPEEDSPINDLE ' calcola la velocità di rotazione
    ng_dac(0,VelSpindle) ' invia all' uscita analogica NGIO
    SpindleStart=true ' Start Mandrino
    ISOV1_status_m_run=0
  case 4 ' start Mandrino in CCW
    Cw=false
    Ccw=true ' set CCw mode
    VelSpindle=(ISOV1_generic(9)*MAXDAC)/MAXSPEEDSPINDLE ' calcola la velocità di rotazione
    ng_dac(0,VelSpindle) ' invia all' uscita analogica NGIO
    SpindleStart=true ' Start Mandrino
    ISOV1_status_m_run=0
  case 5 ' Stop Mandrino
    SpindleStart=false ' Stop Mandrino
    VelSpindle=0 ' set Speed a 0
    ng_dac(0,VelSpindle) ' invia all' uscita analogica NGIO
    ISOV1_status_m_run=0
  case else
    ISOV1_m_ACK=0
endselect
endfunction
```

## 6 Standard I/O

La seguente tabella descrive tutte le I/O Virtuali scambiate tra VTB e l' applicazione PC

I/O	VARIABILE VTB	DESCRIZIONE
Richiesta di RUN esterno	ISOV1_ext_run	<p>Questo bit è posto a True (1) Quando il ciclo PLC effettua una richiesta di <b>RUN</b> all' applicazione IsoNs. Es: Quando viene premuto il pulsante esterno di <b>START</b>.</p> <pre> if input_ext_run = 1 &amp;&amp; ISOV1_status_run=0 ' testa pulsante e se non in run   ISOV1_ext_run=1 ' richiesta di RUN ' Il PC legge questo flag e dà uno start programma endif                     </pre>
Richiesta di STOP esterno	ISOV1_ext_stop	<p>Questo bit è posto a True (1) Quando il ciclo PLC effettua una richiesta di <b>STOP</b> all' applicazione IsoNs. Es: Quando viene premuto il pulsante esterno di <b>STOP</b>.</p> <pre> if input_ext_stop = 1 &amp;&amp; ISOV1_status_run=1 ' testa pulsante e se siamo in run   ISOV1_ext_stop=1 ' richiesta di STOP ' Il PC legge questo flag e dà uno STOP programma endif                     </pre>
Richiesta di PAUSA esterna	ISOV1_ext_pausa	<p>Questo bit è posto a True (1) Quando il ciclo PLC effettua una richiesta di <b>PAUSA</b> all' applicazione IsoNs. Es: Quando viene premuto il pulsante esterno di <b>PAUSA</b>.</p> <pre> if input_ext_pausa = 1 &amp;&amp; ISOV1_status_pausa=0 ' testa pulsante e se non siamo in PAUSA   ISOV1_ext_pausa=1 ' richiesta di PAUSA ' Il PC legge questo flag e dà una PAUSA endif                     </pre>
Switch limite Negativo Asse X	ISOV1_ext_fcm_x	<p>Copiare in questo bit l' ingresso dove è connesso lo switch di limite <b>Negativo</b> Asse X</p> <pre>ISOV1_ext_fcm_x=InputLimtXneg</pre> <p>Automaticamente viene fermato il PartProgram IsoNs</p>
Switch limite Positivo Asse X	ISOV1_ext_fcp_x	<p>Copiare in questo bit l' ingresso dove è connesso lo switch di limite <b>Positivo</b> Asse X</p> <pre>ISOV1_ext_fcp_x=InputLimtXpos</pre> <p>Automaticamente viene fermato il PartProgram IsoNs</p>
Switch limite Negativo Asse Y	ISOV1_ext_fcm_y	<p>Copiare in questo bit l' ingresso dove è connesso lo switch di limite <b>Negativo</b> Asse Y</p> <pre>ISOV1_ext_fcm_y=InputLimtYneg</pre> <p>Automaticamente viene fermato il PartProgram IsoNs</p>

<p><b>Switch limite Positivo Asse Y</b></p>	<p>ISOV1_ext_fcp_y</p>	<p>Copiare in questo bit l' ingresso dove è connesso lo switch di limite <b>Positivo</b> Asse Y</p> <p><b>ISOV1_ext_fcp_y=InputLimtYpos</b></p> <p>Automaticamente viene fermato il PartProgram IsoNs</p>
<p><b>Switch limite Negativo Asse Z</b></p>	<p>ISOV1_ext_fcm_z</p>	<p>Copiare in questo bit l' ingresso dove è connesso lo switch di limite <b>Negativo</b> Asse Z</p> <p><b>ISOV1_ext_fcm_z=InputLimtzneg</b></p> <p>Automaticamente viene fermato il PartProgram IsoNs</p>
<p><b>Switch limite Positivo Asse Z</b></p>	<p>ISOV1_ext_fcp_z</p>	<p>Copiare in questo bit l' ingresso dove è connesso lo switch di limite <b>Positivo</b> Asse Z</p> <p><b>ISOV1_ext_fcp_z=InputLimtZpos</b></p> <p>Automaticamente viene fermato il PartProgram IsoNs</p>
<p><b>Switch limite Negativo Asse A</b></p>	<p>ISOV1_ext_fcm_a</p>	<p>Copiare in questo bit l' ingresso dove è connesso lo switch di limite <b>Negativo</b> Asse A</p> <p><b>ISOV1_ext_fcm_a=InputLimtAneg</b></p> <p>Automaticamente viene fermato il PartProgram IsoNs</p>
<p><b>Switch limite Positivo Asse A</b></p>	<p>ISOV1_ext_fcp_a</p>	<p>Copiare in questo bit l' ingresso dove è connesso lo switch di limite <b>Positivo</b> Asse A</p> <p><b>ISOV1_ext_fcp_a=InputLimtApos</b></p> <p>Automaticamente viene fermato il PartProgram IsoNs</p>
<p><b>Switch limite Negativo Asse B</b></p>	<p>ISOV1_ext_fcm_b</p>	<p>Copiare in questo bit l' ingresso dove è connesso lo switch di limite <b>Negativo</b> Asse B</p> <p><b>ISOV1_ext_fcm_b=InputLimtBneg</b></p> <p>Automaticamente viene fermato il PartProgram IsoNs</p>
<p><b>Switch limite Positivo Asse B</b></p>	<p>ISOV1_ext_fcp_b</p>	<p>Copiare in questo bit l' ingresso dove è connesso lo switch di limite <b>Positivo</b> Asse B</p> <p><b>ISOV1_ext_fcp_b=InputLimtBpos</b></p> <p>Automaticamente viene fermato il PartProgram IsoNs</p>
<p><b>Switch limite Negativo Asse C</b></p>	<p>ISOV1_ext_fcm_c</p>	<p>Copiare in questo bit l' ingresso dove è connesso lo switch di limite <b>Negativo</b> Asse C</p> <p><b>ISOV1_ext_fcm_c=InputLimtCneg</b></p> <p>Automaticamente viene fermato il PartProgram IsoNs</p>

<p><b>Switch limite Positivo Asse C</b></p>	<p>ISOV1_ext_fcp_c</p>	<p>Copiare in questo bit l' ingresso dove è connesso lo switch di limite <b>Positivo</b> Asse C</p> <p><b>ISOV1_ext_fcp_c=InputLimtCpos</b></p> <p>Automaticamente viene fermato il PartProgram IsoNs</p>
<p><b>Switch limite Negativo Asse U</b></p>	<p>ISOV1_ext_fcm_u</p>	<p>Copiare in questo bit l' ingresso dove è connesso lo switch di limite <b>Negativo</b> Asse U</p> <p><b>ISOV1_ext_fcm_u=InputLimtUneg</b></p> <p>Automaticamente viene fermato il PartProgram IsoNs</p>
<p><b>Switch limite Positivo Asse U</b></p>	<p>ISOV1_ext_fcp_u</p>	<p>Copiare in questo bit l' ingresso dove è connesso lo switch di limite <b>Positivo</b> Asse U</p> <p><b>ISOV1_ext_fcp_u=InputLimtUpos</b></p> <p>Automaticamente viene fermato il PartProgram IsoNs</p>
<p><b>Switch limite Negativo Asse V</b></p>	<p>ISOV1_ext_fcm_v</p>	<p>Copiare in questo bit l' ingresso dove è connesso lo switch di limite <b>Negativo</b> Asse V</p> <p><b>ISOV1_ext_fcm_v=InputLimtVneg</b></p> <p>Automaticamente viene fermato il PartProgram IsoNs</p>
<p><b>Switch limite Positivo Asse V</b></p>	<p>ISOV1_ext_fcp_v</p>	<p>Copiare in questo bit l' ingresso dove è connesso lo switch di limite <b>Positivo</b> Asse V</p> <p><b>ISOV1_ext_fcp_v=InputLimtVpos</b></p> <p>Automaticamente viene fermato il PartProgram IsoNs</p>
<p><b>Switch limite Negativo Asse W</b></p>	<p>ISOV1_ext_fcm_w</p>	<p>Copiare in questo bit l' ingresso dove è connesso lo switch di limite <b>Negativo</b> Asse W</p> <p><b>ISOV1_ext_fcm_w=InputLimtwneg</b></p> <p>Automaticamente viene fermato il PartProgram IsoNs</p>
<p><b>Switch limite Positivo Asse W</b></p>	<p>ISOV1_ext_fcp_w</p>	<p>Copiare in questo bit l' ingresso dove è connesso lo switch di limite <b>Positivo</b> Asse W</p> <p><b>ISOV1_ext_fcp_w=InputLimtWpos</b></p> <p>Automaticamente viene fermato il PartProgram IsoNs</p>
<p><b>Sensor3 Acq</b></p>	<p>ISOV1_ext_acq</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso il sensore di acquisizione <b>funzione G102</b></p> <p><b>ISOV1_ext_acq=InputAcq</b></p> <p>Automaticamente l' applicazione VTB ferma gli assi.</p>

<p>Stop Assi</p>	<p>ISOV1_stop_assi</p>	<p>Quando questo bit è settato, l' applicazione IsoNs GCODE viene interrotta Questo bit è in alternativa a <b>ISOV1_ext_stop</b>. Per es: può essere usato per frozare un STOP Assi da evento VTB.</p> <pre> <b>if</b> InputForceStop = 1 &amp;&amp; ISOV1_status_run=1 <b>' Forza Stop Assi</b>       ISOV1_stop_assi=1 <b>' Il Part Program PC viene interrotto</b> <b>endif</b>                     </pre>
<p>Stop Assi con emergenza</p>	<p>ISOV1_stop_emcy</p>	<p>Quando questo bit è settato, l' applicazione IsoNs GCODE viene interrotta e viene dato un allarme di <b>EMERGENZA</b> Tutti i Drives vengono disabilitati</p> <pre> <b>if</b> InputForceEmcy = 1 &amp;&amp; ISOV1_status_run=1 <b>' Forza uno STOP assi con emergenza</b>       ISOV1_stop_EMCY=1 <b>' IL PC interrompe il Part Program con ALLARME</b> <b>endif</b>                     </pre>
<p>Ingresso di Emergenza</p>	<p>ISOV1_ext_emcy</p>	<p>Copiare in questo bit l' ingresso digitale dove è connessa l' <b>ERMERGENZA GENERALE</b> (tipicamente N.C.)</p> <pre> ISOV1_ext_emcy=!InputGeneralEmcy                     </pre> <p>Automaticamente viene fermato il PartProgram IsoNs co ALLARME</p>
<p>Manuale JOG X -</p>	<p>ISOV1_ext_jogm_x</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso il <b>PULSANTE ESTERNO JOG X -</b></p> <pre> ISOV1_ext_jogm_x=JogExtXm                     </pre> <p>L' asse viene mosso in direzione <b>NEGATIVA</b> all velocità <b>VJOG</b> inserita nei parametri di IsoNs PC</p>
<p>Manuale JOG X +</p>	<p>ISOV1_ext_jogp_x</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso il <b>PULSANTE ESTERNO JOG X +</b></p> <pre> ISOV1_ext_jogp_x=JogExtXp                     </pre> <p>L' asse viene mosso in direzione <b>POSITIVA</b> all velocità <b>VJOG</b> inserita nei parametri di IsoNs PC</p>
<p>Manuale JOG Y -</p>	<p>ISOV1_ext_jogm_y</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso il <b>PULSANTE ESTERNO JOG Y -</b></p> <pre> ISOV1_ext_jogm_y=JogExtYm                     </pre> <p>L' asse viene mosso in direzione <b>NEGATIVA</b> all velocità <b>VJOG</b> inserita nei parametri di IsoNs PC</p>
<p>Manuale JOG Y +</p>	<p>ISOV1_ext_jogp_y</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso il <b>PULSANTE ESTERNO JOG Y +</b></p> <pre> ISOV1_ext_jogp_y=JogExtYp                     </pre> <p>L' asse viene mosso in direzione <b>POSITIVA</b> all velocità <b>VJOG</b> inserita nei parametri di IsoNs PC</p>

<p><b>Manuale JOG Z -</b></p>	<p>ISOV1_ext_jogm_z</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso il <b>PULSANTE ESTERNO JOG Z -</b></p> <p><b>ISOV1_ext_jogm_z=JogExtZm</b></p> <p>L' asse viene mosso in direzione <b>NEGATIVA</b> all velocità <b>VJOG</b> inserita nei parametri di IsoNs PC</p>
<p><b>Manuale JOG Z +</b></p>	<p>ISOV1_ext_jogp_z</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso il <b>PULSANTE ESTERNO JOG Z +</b></p> <p><b>ISOV1_ext_jogp_z=JogExtZp</b></p> <p>L' asse viene mosso in direzione <b>POSITIVA</b> all velocità <b>VJOG</b> inserita nei parametri di IsoNs PC</p>
<p><b>Manuale JOG A -</b></p>	<p>ISOV1_ext_jogm_a</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso il <b>PULSANTE ESTERNO JOG A -</b></p> <p><b>ISOV1_ext_jogm_a=JogExtAm</b></p> <p>L' asse viene mosso in direzione <b>NEGATIVA</b> all velocità <b>VJOG</b> inserita nei parametri di IsoNs PC</p>
<p><b>Manuale JOG A +</b></p>	<p>ISOV1_ext_jogp_a</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso il <b>PULSANTE ESTERNO JOG A +</b></p> <p><b>ISOV1_ext_jogp_a=JogExtAp</b></p> <p>L' asse viene mosso in direzione <b>POSITIVA</b> all velocità <b>VJOG</b> inserita nei parametri di IsoNs PC</p>
<p><b>Manuale JOG B -</b></p>	<p>ISOV1_ext_jogm_b</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso il <b>PULSANTE ESTERNO JOG B -</b></p> <p><b>ISOV1_ext_jogm_b=JogExtBm</b></p> <p>L' asse viene mosso in direzione <b>NEGATIVA</b> all velocità <b>VJOG</b> inserita nei parametri di IsoNs PC</p>
<p><b>Manuale JOG B +</b></p>	<p>ISOV1_ext_jogp_b</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso il <b>PULSANTE ESTERNO JOG B +</b></p> <p><b>ISOV1_ext_jogp_b=JogExtBp</b></p> <p>L' asse viene mosso in direzione <b>POSITIVA</b> all velocità <b>VJOG</b> inserita nei parametri di IsoNs PC</p>
<p><b>Manuale JOG C -</b></p>	<p>ISOV1_ext_jogm_c</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso il <b>PULSANTE ESTERNO JOG C -</b></p> <p><b>ISOV1_ext_jogm_c=JogExtCm</b></p> <p>L' asse viene mosso in direzione <b>NEGATIVA</b> all velocità <b>VJOG</b> inserita nei parametri di IsoNs PC</p>

<p><b>Manuale JOG C +</b></p>	<p>ISOV1_ext_jogp_c</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso il <b>PULSANTE ESTERNO JOG C +</b></p> <p><b>ISOV1_ext_jogp_c=JogExtCp</b></p> <p>L' asse viene mosso in direzione <b>POSITIVA</b> all velocità <b>VJOG</b> inserita nei parametri di IsoNs PC</p>
<p><b>Manuale JOG U -</b></p>	<p>ISOV1_ext_jogm_u</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso il <b>PULSANTE ESTERNO JOG U -</b></p> <p><b>ISOV1_ext_jogm_u=JogExtUm</b></p> <p>L' asse viene mosso in direzione <b>NEGATIVA</b> all velocità <b>VJOG</b> inserita nei parametri di IsoNs PC</p>
<p><b>Manuale JOG U +</b></p>	<p>ISOV1_ext_jogp_u</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso il <b>PULSANTE ESTERNO JOG U +</b></p> <p><b>ISOV1_ext_jogp_u=JogExtUp</b></p> <p>L' asse viene mosso in direzione <b>POSITIVA</b> all velocità <b>VJOG</b> inserita nei parametri di IsoNs PC</p>
<p><b>Manuale JOG V -</b></p>	<p>ISOV1_ext_jogm_v</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso il <b>PULSANTE ESTERNO JOG V -</b></p> <p><b>ISOV1_ext_jogm_v=JogExtVm</b></p> <p>L' asse viene mosso in direzione <b>NEGATIVA</b> all velocità <b>VJOG</b> inserita nei parametri di IsoNs PC</p>
<p><b>Manuale JOG V +</b></p>	<p>ISOV1_ext_jogp_v</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso il <b>PULSANTE ESTERNO JOG V +</b></p> <p><b>ISOV1_ext_jogp_v=JogExtVp</b></p> <p>L' asse viene mosso in direzione <b>POSITIVA</b> all velocità <b>VJOG</b> inserita nei parametri di IsoNs PC</p>
<p><b>Manuale JOG W -</b></p>	<p>ISOV1_ext_jogm_w</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso il <b>PULSANTE ESTERNO JOG W -</b></p> <p><b>ISOV1_ext_jogm_w=JogExtWm</b></p> <p>L' asse viene mosso in direzione <b>NEGATIVA</b> all velocità <b>VJOG</b> inserita nei parametri di IsoNs PC</p>
<p><b>Manuale JOG W +</b></p>	<p>ISOV1_ext_jogp_w</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso il <b>PULSANTE ESTERNO JOG W +</b></p> <p><b>ISOV1_ext_jogp_w=JogExtWp</b></p> <p>L' asse viene mosso in direzione <b>POSITIVA</b> all velocità <b>VJOG</b> inserita nei parametri di IsoNs PC</p>

<p><b>Manuale JOG - Asse generico</b></p>	<p>ISOV1_ext_jogm</p>	<p>Questo bit è usato per muovere in direzione <b>NEGATIVA</b> un asse generico. L' asse viene indicato dalla variabile <b>ISOV1_asse_man</b>. Copiare in questo bit l' ingresso digitale di <b>JOG -</b></p> <p><b>ISOV1_ext_jogm=JogExtM</b></p> <p>ISOV1_asse_man=0 → X Asse is set ISOV1_asse_man=1 → Y Asse is set ISOV1_asse_man=2 → Z Asse is set ISOV1_asse_man=3 → A Asse is set ISOV1_asse_man=4 → B Asse is set ISOV1_asse_man=5 → C Asse is set ISOV1_asse_man=6 → U Asse is set ISOV1_asse_man=7 → V Asse is set ISOV1_asse_man=8 → W Asse is set</p>
<p><b>Manuale JOG + Asse generico</b></p>	<p>ISOV1_ext_jogp</p>	<p>Questo bit è usato per muovere in direzione <b>POSITIVA</b> un asse generico. L' asse viene indicato dalla variabile <b>ISOV1_asse_man</b>. Copiare in questo bit l' ingresso digitale di <b>JOG +</b> (vedi sopra)</p> <p><b>ISOV1_ext_jogp=JogExtP</b></p>
<p><b>Homing X Switch</b></p>	<p>ISOV1_ext_fcx_x</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso lo <b>SWITCH DI HOMING X</b></p> <p><b>ISOV1_ext_fcx_x=InputHoming_X</b></p> <p>Questo bit è usato Quando il comeo HOME è invocato</p>
<p><b>Homing Y Switch</b></p>	<p>ISOV1_ext_fcx_y</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso lo <b>SWITCH DI HOMING Y</b></p> <p><b>ISOV1_ext_fcx_y=InputHoming_Y</b></p> <p>Questo bit è usato Quando il comeo HOME è invocato</p>
<p><b>Homing Z Switch</b></p>	<p>ISOV1_ext_fcx_z</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso lo <b>SWITCH DI HOMING Z</b></p> <p><b>ISOV1_ext_fcx_z=InputHoming_Z</b></p> <p>Questo bit è usato Quando il comeo HOME è invocato</p>
<p><b>Homing A Switch</b></p>	<p>ISOV1_ext_fcx_a</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso lo <b>SWITCH DI HOMING A</b></p> <p><b>ISOV1_ext_fcx_a=InputHoming_A</b></p> <p>Questo bit è usato Quando il comeo HOME è invocato</p>
<p><b>Homing B Switch</b></p>	<p>ISOV1_ext_fcx_b</p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso lo <b>SWITCH DI HOMING B</b></p> <p><b>ISOV1_ext_fcx_b=InputHoming_B</b></p> <p>Questo bit è usato Quando il comeo HOME è invocato</p>

<p><b>Homing C Switch</b></p>	<p><b>ISOV1_ext_fcz_c</b></p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso lo <b>SWITCH DI HOMING C</b></p> <p><b>ISOV1_ext_fcz_c=InputHoming_C</b></p> <p>Questo bit è usato Quando il comeo HOME è invocato</p>
<p><b>Homing U Switch</b></p>	<p><b>ISOV1_ext_fcz_u</b></p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso lo <b>SWITCH DI HOMING U</b></p> <p><b>ISOV1_ext_fcz_u=InputHoming_U</b></p> <p>Questo bit è usato Quando il comeo HOME è invocato</p>
<p><b>Homing V Switch</b></p>	<p><b>ISOV1_ext_fcz_v</b></p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso lo <b>SWITCH DI HOMING V</b></p> <p><b>ISOV1_ext_fcz_v=InputHoming_V</b></p> <p>Questo bit è usato Quando il comeo HOME è invocato</p>
<p><b>Homing W Switch</b></p>	<p><b>ISOV1_ext_fcz_W</b></p>	<p>Copiare in questo bit l' ingresso digitale dove è connesso lo <b>SWITCH DI HOMING W</b></p> <p><b>ISOV1_ext_fcz_w=InputHoming_W</b></p> <p>Questo bit è usato Quando il comeo HOME è invocato</p>

## 7 Status Word

L' applicazione VTB usa una **STATUS WORD** per comunicare con l' applicazione PC. Normalmente questa STATUS WORD è settata dalle funzioni interne di VTB, ma l' applicazione può leggere lo stato. La STATUS WORD è mappata a bit

Nome Bit VTB	DESCRIZIONE
ISOV1_status_run	A <b>True</b> (1) Quando l' applicazione IsoNs PC è in RUN
ISOV1_status_move	A <b>True</b> (1) Quando assi in movimento
ISOV1_status_pausa	A <b>True</b> (1) Quando l' applicazione IsoNs PC è in PAUSA
ISOV1_status_error	A <b>True</b> (1) Quando un allarme è presente (EMCY drives, Emergenza ecc.)
ISOV1_status_rzero	A <b>True</b> (1) durante la fase di ricerca di HOMING
ISOV1_status_rsens	A <b>True</b> (1) Quando l' acquisizione sensore è iniziata A <b>False</b> (0) Quando l' acquisizione sensore è terminata Funzione di IsoNs G102
ISOV1_status_para_upd	A <b>True</b> (1) Quando l' applicazione PC scrive i parametri in VTB Questo flag non viene resettato automaticamente, ma in modo manuale dall' applicazione VTB. In alcuni casi è necessario sapere Quando IsoNs scrive i Parametri in modo da poter effettuare operazioni su questi.
ISOV1_status_m_run	A <b>True</b> (1) Quando è in esecuzione un ciclo M. Viene resettato dal funzione di gestione cicli M <b>Vedi capitolo 5 - "Utilizzo delle funzioni M interne al CNC "</b>
ISOV1_status_m_stop	A <b>True</b> (1) da IsoNs Quando è richiesto uno <b>STOP</b> L' applicazione VTB deve interrompere eventuali cicli M rimasti attivi e resettare questo flag.
ISOV1_status_home_x	A <b>True</b> (1) Quando il relativo asse ha effettuato la procedura di HOMING
ISOV1_status_home_y	
ISOV1_status_home_z	
ISOV1_status_home_a	
ISOV1_status_home_b	
ISOV1_status_home_c	
ISOV1_status_home_u	
ISOV1_status_home_v	
ISOV1_status_home_w	
ISOV1_status_enable_x	A <b>True</b> (1) Quando il relativo Asse è abilitato A <b>False</b> (0) Quando il relativo Asse è disabilitato
ISOV1_status_enable_y	
ISOV1_status_enable_z	
ISOV1_status_enable_a	
ISOV1_status_enable_b	
ISOV1_status_enable_c	
ISOV1_status_enable_u	
ISOV1_status_enable_v	
ISOV1_status_enable_w	

## 8 Gestione PLC I/O

L' applicazione VTB può essere usata per la gestione del ciclo PLC

Prima è necessario avere inserito nel progetto i relativi OGGETTI (**Vedi Capitolo 4 - "Inserire I/O"**)

L' applicazione VTB usa una definizione a Bit per la gestione delle I/O.

Il numero massimo I/O è:

**256 Ingressi Digitali**

**256 Uscite Digitali**

Le I/O sono a blocchi di 16 bit

Se viene usata una NGQ-NGQx con 11 ingressi digitali, non è possibile utilizzare gli ingressi da 11 a 15 da 24 a 31 ecc. del relativo blocco, a causa che questi non sono presenti nella scheda NGQ-NGQx

Se viene usata una NGIO-NGMIO con 14 uscite digitali, non è possibile utilizzare le uscite da 14 a 15 da 30 a 31 ecc. del relativo blocco, a causa che queste non sono presenti nella scheda NGIO-NGMIO

Possono essere comunque sempre utilizzati le uscite 14,15 30,31 ecc. per flag interni

### 8.1 Definizione bit Ingressi Digitali

ISOV1.inp0 → Ingresso Digitale 1

ISOV1.inp1 → Ingresso Digitale 2

ISOV1.inp2 → Ingresso Digitale 3

.

.

.

ISOV1.inp255 → Ingresso Digitale 255

### 8.2 Definizione bit Uscite Digitali

ISOV1.out0 → Uscita Digitale 1

ISOV1.out1 → Uscita Digitale 2

ISOV1.out2 → Uscita Digitale 3

.

.

ISOV1.out255 → Uscita Digitale 255

Es:

```
if ISOV1.inp0 = 1 && ISOV1.inp1=0
    ISOV1.out0=true 'set uscita 0
endif
```

## 9 FEED Override Esterno

Con VTB è possibile gestire un potenziometro esterno come funzione di Override sugli Assi  
 Generalmente il potenziometro è collegato su un ingresso Analogico  
 La variabile è :

**ISOV1\_vper**

Il range di valore è da 0 a 1024, si può copiare direttamente l' ingresso analogico nella variabile

Es:

Ingresso Analogico Nr. 1

Inserire questo codice nel ciclo Main o Task Plc

**ISOV1\_vper=ng\_adc(0)**

Se viene utilizzato un ingresso analogico (12 bit – range da 0 a 4095)

dividere per 4 **ISOV1\_vper**

**ISOV1\_vper=ng\_adc(0)**

**ISOV1\_vper=ISOV1\_vper/4**

## 10 Volantino Elettronico

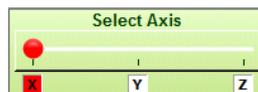
Questo capitolo descrive come gestire il Volantino Elettronico.

### Variabile VTB

**ISOV1\_soft\_sel\_man**

→ **Se = 1** la selezione asse è controllata da selettore virtuale su PC

→ **Se = 0** la selezione asse è controllata da variabile interna VTB



**ISOV1\_qvola**

→

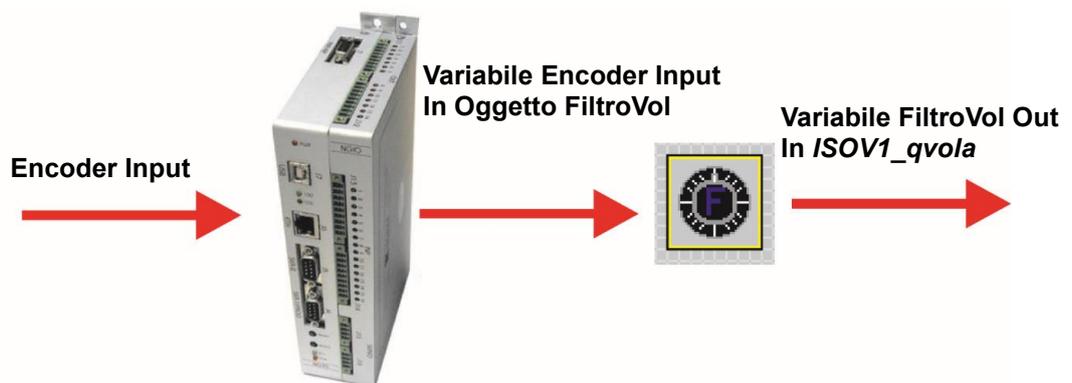
Quota encoder incrementale del volantino elettronico

Generalmente questa variabile è presa dall' oggetto FiltroVol.

Questo per ottenere uno smoothing sugli impulsi del volantino.



**HeWheel**



**ISOV1\_asse\_man**

→

Seleziona l' Asse da muovere con il volantino

0 → X            1 → Y            2 → Z

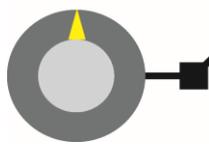
3 → A            4 → B            5 → C

6 → U            7 → V            8 → W

Generalmente questa variabile è presa da un selettore

Vale solo se **ISOV1\_soft\_sel\_man=0**

**Selettore Assi**

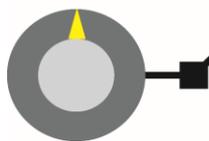


- Input 1 - X -  $ISOV1\_asse\_man = 0$
- Input 2 - Y -  $ISOV1\_asse\_man = 1$
- Input 3 - Z -  $ISOV1\_asse\_man = 2$

$ISOV1\_soft\_sel\_man$  → Se = 0,  $ISOV1\_asse\_man$  è gestito tramite codice VTB  
 Se = 1,  $ISOV1\_asse\_man$  è gestito dal PC (selettore virtuale)

$ISOV1\_msofv$  → Setta il moltiplicatore impulsi (generalmente x1 x10 x100)  
 Se  $ISOV1\_soft\_sel\_man = 1$  questa variabile è controllata dal PC  
 Questa variabile viene combinata con il parametro IsoNs  $DSOFV\_X, \_Y$  etc  
 $DSOFV\_$  generalmente contiene il numero impulsi Volantino moltiplicati x 4  
 Es: i/g Volantino=100  $DSOFV\_ = 400$

**Selettore Moltiplicatore**



- Input 1 - x1 -  $ISOV1\_msofv = 1$
- Input 1 - x10 -  $ISOV1\_msofv = 10$
- Input 1 - x100 -  $ISOV1\_msofv = 100$

**Esempio VTB**

Sistema NG35+NGIO con le seguenti connessioni:

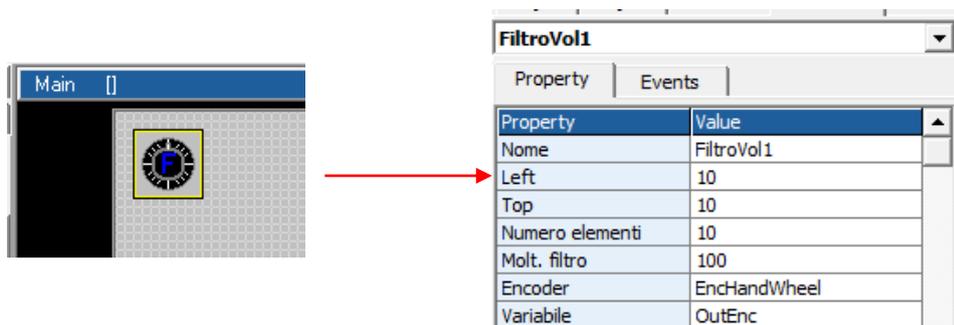
- Ch0 Encoder → Encoder Volantino
- Digital Input 0 → Selettore Asse X
- Digital Input 1 → Selettore Asse Y
- Digital Input 2 → Selettore Asse Z
- Digital Input 3 → Selettore Moltiplicatore x1
- Digital Input 4 → Selettore Moltiplicatore x10
- Digital Input 5 → Selettore Moltiplicatore x100
- Digital Input 6 → Input JOG Positivo
- Digital Input 7 → Input JOG Negativo

**Variable usate:**

- EncHeWheel → Long
- OutEnc → Long
- Input → Long

Inserire un Object FiltroVol in Main con i seguenti parametri :

**Oggetti**→**Motor Control**→**CFiltrovol.vco**



Inserire il seguente codice in **Init Task Plc**:

```
ISOV1_soft_sel_man=0 ' Abilita il selettore interno a VTB
FiltroVol1.enable=1 ' Abilita il filtro
```

Inserire il seguente codice in **Task Plc**:

```
-----
' Legge l' encoder del volantino
' e lo copia in EncHeWheel
' Nota: le ( ) sono per passare il puntatore – MOLTO IMPORTANTE
-----
ng_enc(0,EncHeWheel())

-----
'Setta il Moltiplicatore
-----
if ISOV1.inp3 'Moltiplicatore x1
    ISOV1_msofv=1
endif
if ISOV1.inp4 'Moltiplicatore x10
    ISOV1_msofv=10
endif
if ISOV1.inp5 'Moltiplicatore x100
    ISOV1_msofv=100
endif
```

```
-----  
'Set l' asse da muovere  
-----
```

```
if ISOV1.inp0           'Asse X  
    ISOV1_asse_man=0  
endif  
if ISOV1.inp1           'Asse Y  
    ISOV1_asse_man=1  
endif  
if ISOV1.inp2           'Asse Z  
    ISOV1_asse_man=2  
endif
```

```
-----  
'Update Hewheel da FiltroVol  
-----
```

```
if !ISOV1_status_run  
    FiltroVol1.enable=true           'Abilita filtro hewheel  
    ISOV1_qvola=OutEnc  
else  
    FiltroVol1.enable=false         'Disabilita filtro hewheel  
    ISOV1_qvola=0  
endif
```

```
-----  
'Update Jog Input  
-----
```

```
ISOV1_ext_jogp=ISOV1.inp6  
ISOV1_ext_jogm=ISOV1.inp7
```

## 11 Parametri Macchina

Con l' applicazione VTB è possibile leggere e scrivere i parametri macchina. Generalmente questa funzione non è necessaria, poiché i parametri macchina sono già gestiti dal sistema IsoNs e applicazione sistema VTB , ma si può comunque leggere o scrivere o parametri.

Tutti i parametri sono disponibili nell' Array **ISOV1\_PARA**  
 La dimensione dell' Array dipende dal numero di Assi configurati  
 Tutti i Parametri sono di tipo Long (32 bit con segno)

### ISOV1\_PARA array

0-49 Parametri generali	50-99 Parametri Asse X	100-149 Parametri Asse Y	....
-------------------------	------------------------	--------------------------	------

### 11.1 Parametri Generali

Idx ISOV1_Para	Nome	Descrizione
0	VRIPOS	Velocità Assi di riposizionamento dopo Pausa (se non è stata configurata la Macro <b>GOPAUSE</b> )
1	SGLP	Soglia Spigolo
2	SGLR	Soglia errore arco
3	ACQ_MODE	Tipo di acquisizione <b>0 → Velocità</b> <b>1 → step by step</b>
4	ACQ_VEL	Velocità di acquisizione
5	ACQ_STEP	Lunghezza dello step per acquisizione tipo 1
6	ACQ_TIME	Tempo tra Step e Step per acquisizione tipo 1
7	ACC_QSTOP	Accelerazione di Quick Stop
8	JERK	Jerk
9	NOSHORT	Short linee
Free 10 a 49	.....	.....

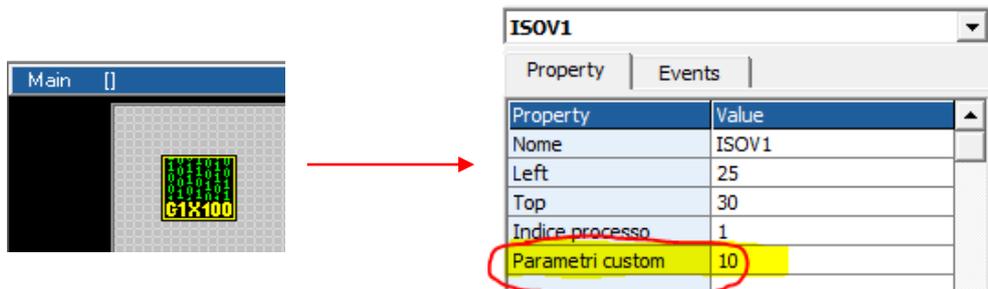
**11.2 Parametri Asse X**  
(50 x blocks)

Idx ISOV1_Para	Nome	Descrizione
50	VJOG	JOG Feed Asse (mm/min)
51	ACC_JOG	Accelerazione JOG Asse (count)
52	LIMITE_N_	Limite Software Negativo Asse
53	LIMITE_P_	Limite Software Positivo Asse
54	DSOFV	Divisore Volantino Elettronico
55	RZERO_MODE	Homing modo
56	RZERO_OFFSET	Homing Offset
57	RZERO_PRESET	Homing Preset
58	RZERO_VEL	Homing High Feed Asse
59	RZERO_VELF	Homing Low Feed Asse
60	RZERO_ACC	Homing Accelerazione Asse
61	MSOF	Count/Revolution Impulsi Asse
62	DSOF	Sviluppo a giro motore Asse
63	GANTRY	Gantry Asse
64	SGL_3D	Soglia 3D Asse
65	BACKSLASH	Asse Backslash (um)
66	TBCK	Asse Time Backslash (TAU)
67	TSHF	Speed Shift
<b>Free 67 a 79</b>		
<b>I seguenti parametri sono presenti solo per sistemi analogici +/- 10V</b>		
80	PID_KP	(Proportional Costant) Asse
81	PID_KI	(Integral Costant) Asse
82	PID_KV	(Feed Costant) Asse
83	PID_I_LIMIT	IL (Integration Limit) Asse
84	PID_DIV	PID Magnitudine Asse
85	PID_SERVO	Servo Error Asse (um)
86	PID_TIME_SERVO	Tempo Servo Error Asse
87	PID_DIR	Direzione Analogica Asse
88	PID_OFFS_ANA	Offset Analogica Asse
<b>Free 88 to 99</b>		
<b>100 a 149 Parametri Asse Y</b>		
<b>150 a 199 Parametri Asse Z (Se presente)</b>		
<b>200 a 249 Parametri Asse A (Se presente)</b>		
<b>250 a 299 Parametri Asse B (Se presente)</b>		
<b>300 a 349 Parametri Asse C (Se presente)</b>		
<b>350 a 399 Parametri Asse U (Se presente)</b>		
<b>400 a 449 Parametri Asse V (Se presente)</b>		
<b>450 a 499 Parametri Asse W (Se presente)</b>		

### 11.3 Parametri Custom

Nell' applicazione IsoNs PC è possibile dichiarare dei Parametri Custom relativi all' applicazione. Questi Parametri possono essere letti dalla' applicazione VTB.

Per prima cosa è necessario dichiarare nell' Oggetto ISOVIRTUAL (ISOV1) il NUMERO MASSIMO DI PARAMETRI CUSTOM per la relativa applicazione



I parametri CUSTOM sono disponibili nel seguente ARRAY:

**ISOV1\_PARA(ISOV1\_P\_CUSTOM)** - Long

Dove **ISOV1\_P\_CUSTOM** è una define calcolata automaticamente dal compilatore di VTB.

A questo indirizzo è presente il primo parametro custom configurato.

Il secondo si trova in **ISOV1\_PARA(ISOV1\_P\_CUSTOM+1)** ecc.

Il valore di **ISOV1\_P\_CUSTOM** è:

**ISOV1\_P\_CUSTOM = 50 \* Numero Assi + 50**

**Es: 4 Assi configurati e 3 Parametri custom**

CUSTOM_1	Custom parameter 1	100
CUSTOM_2	Custom parameter 2	100
CUSTOM_3	Custom parameter 3	100

**CUSTOM\_1** Deve avere un Indirizzo **250**

**CUSTOM\_2** Deve avere un Indirizzo **251**

**CUSTOM\_3** Deve avere un Indirizzo **252**

**CustomPar1=ISOV1\_PARA(ISOV1\_P\_CUSTOM)**

**CustomPar2=ISOV1\_PARA(ISOV1\_P\_CUSTOM+1)**

**CustomPar3=ISOV1\_PARA(ISOV1\_P\_CUSTOM+2)**

## 12 Allarmi mappati a Bit

Tutti gli allarmi sono mappati a Bit. Se l' applicazione setta il relativo Bit, l' allarme è attivato. Generalmente questi allarmi sono attivati automaticamente dal sistema VTB IsoNs, Quando è necessario. Comunque l' applicazione può leggere e scrivere gli allarmi.

Nome Bit	Descrizione Allarme
ISOV1.allarm0	X Limite Negativo Raggiunto
ISOV1.allarm1	X Limite Positivo Raggiunto
ISOV1.allarm2	Y Limite Negativo Raggiunto
ISOV1.allarm3	Y Limite Positivo Raggiunto
ISOV1.allarm4	Z Limite Negativo Raggiunto
ISOV1.allarm5	Z Limite Positivo Raggiunto
ISOV1.allarm6	A Limite Negativo Raggiunto
ISOV1.allarm7	A Limite Positivo Raggiunto
ISOV1.allarm8	B Limite Negativo Raggiunto
ISOV1.allarm9	B Limite Positivo Raggiunto
ISOV1.allarm10	C Limite Negativo Raggiunto
ISOV1.allarm11	C Limite Positivo Raggiunto
ISOV1.allarm12	U Limite Negativo Raggiunto
ISOV1.allarm13	U Limite Positivo Raggiunto
ISOV1.allarm14	V Limite Negativo Raggiunto
ISOV1.allarm15	V Limite Positivo Raggiunto
ISOV1.allarm16	W Limite Negativo Raggiunto
ISOV1.allarm17	W Limite Positivo Raggiunto
ISOV1.allarm18	X SERVO-EMERGENZA
ISOV1.allarm19	Y SERVO-EMERGENZA
ISOV1.allarm20	Z SERVO-EMERGENZA
ISOV1.allarm21	A SERVO-EMERGENZA
ISOV1.allarm22	B SERVO-EMERGENZA
ISOV1.allarm23	C SERVO-EMERGENZA
ISOV1.allarm24	U SERVO-EMERGENZA
ISOV1.allarm25	V SERVO-EMERGENZA
ISOV1.allarm26	W SERVO-EMERGENZA
ISOV1.allarm27	EMERGENZA Generale Attivata
ISOV1.allarm28	Errore Acquisizione
ISOV1.allarm29	Short line trovate (se attivato NO_SHORT=2)
ISOV1.allarm30	Libero
ISOV1.allarm31	Libero
ISOV1.allarm32	X Asse Errore Configurazione (CanOpen or Ethercat)
ISOV1.allarm32	Y Asse Errore Configurazione (CanOpen or Ethercat)

ISOV1.allarm34	Z Asse Errore Configurazione (CanOpen or Ethercat)
ISOV1.allarm35	A Asse Errore Configurazione (CanOpen or Ethercat)
ISOV1.allarm36	B Asse Errore Configurazione (CanOpen or Ethercat)
ISOV1.allarm37	C Asse Errore Configurazione (CanOpen or Ethercat)
ISOV1.allarm38	U Asse Errore Configurazione (CanOpen or Ethercat)
ISOV1.allarm39	V Asse Errore Configurazione (CanOpen or Ethercat)
ISOV1.allarm40	W Asse Errore Configurazione (CanOpen or Ethercat)
ISOV1.allarm41	Libero
ISOV1.allarm42	Libero
	....
ISOV1.allarm255	Libero

### 13 Allarmi CanOpen

La gestione degli allarmi Canopen (EMCY-OBJ), è direttamente controllata dal relativo OGGETTO ASSE CARICATE.

Quando l' allarme si presenta, il relativo bit viene settato:

**X SERVO-EMERGENZA**

**Y SERVO-EMERGENZA**

**Y SERVO-EMERGENZA**

ecc

In **ISOV1\_last\_allarm(nodo Asse)** si può leggere il codice di errore CanOpen

## 14 Esempi

Di seguito alcuni esempi VTB - IsoNs

### 14.1 NGQ-NGMEVO 3 Assi Step/Dir

Link RS32 su COM1 NGQ-NGMEVO

#### Digital Inputs

Input 1	→	Switch Home X (N.C.)
Input 2	→	Switch Home Y (N.C.)
Input 3	→	Switch Home Z (N.C.)
Input 4	→	EMERGENZA GENERALE (N.C.)
Input 5	→	Pulsante JOG X+ (N.O.)
Input 6	→	Pulsante JOG X- (N.O.)
Input 7	→	Pulsante JOG Y+ (N.O.)
Input 8	→	Pulsante JOG Y- (N.O.)
Input 9	→	Pulsante JOG Z+ (N.O.)
Input 10	→	Pulsante JOG Z- (N.O.)

#### Analog Inputs

Inputs 1 → Potenziometro Override Assi

Per Abilitare il potenziometro è necessario selezionare il pulsante "Ext OW" da interfaccia di IsoNs



#### Digital Outputs

Output 1	→	X Asse Abiliato
Output 2	→	Y Asse Abiliato
Output 3	→	Z Asse Abiliato
Output 4	→	CNC Errore

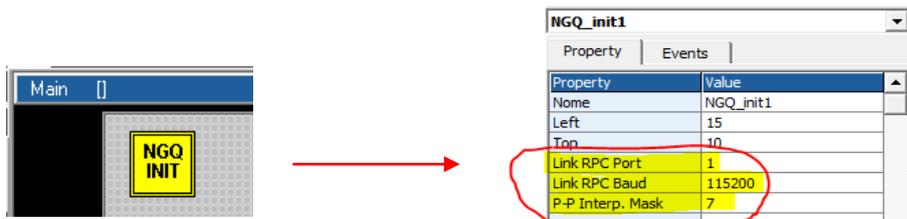
#### Assi Outputs

Step/Dir Ch 1	→	X
Step/Dir Ch 2	→	Y
Step/Dir Ch 3	→	Z

- 1) Aprire un nuovo progetto VTB e selezionare NGQ hardware o NGMEVO (L' esempio è NGQ)  
Selezionare 4 Ms sample

Start Page:	<input type="text" value="1"/>
Sample:	<input type="text" value="4"/> mS
Task Time:	<input type="text" value="5"/> x 4 = <input style="background-color: red; color: white;" type="text" value="20"/> mS
Screensave:	<input checked="" type="checkbox"/> Enable <input type="text" value="30"/> sec

2) Set link su COM1(o COM2) NGQ-NGMEVO e PP Interp mask su 7 (Canali X Y Z abilitati)



3) Inserire un Oggetto ISOVIRTUAL e lasciare le Proprietà default

Oggetti → Iso\_Ns → IsoVirtual.vco

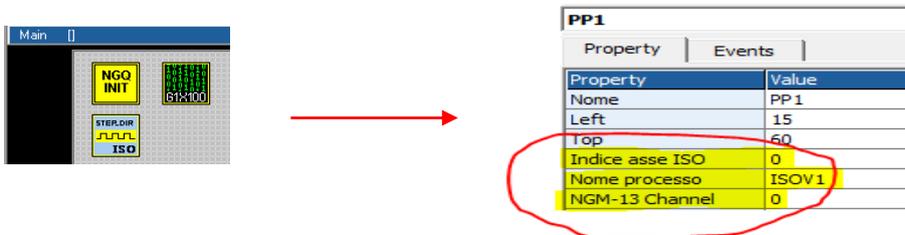


4) Inserire Asse X ISOPP (è lo stesso per NGMEVO o NGQ)

Oggetti → Iso\_Ns → IsoPP.vco

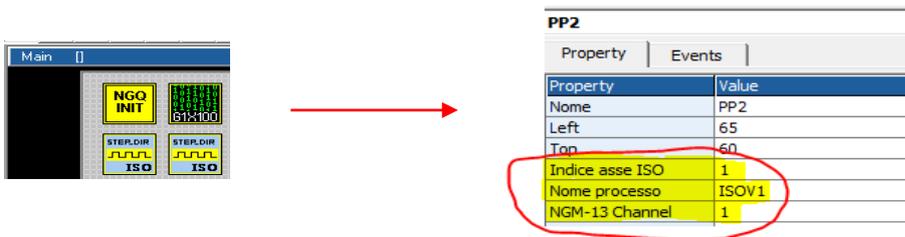


5) Settare le seguenti Proprietà



6) Inserire Asse Y ISOPP e settare le seguenti Proprietà

Oggetti → Iso\_Ns → IsoPP.vco



7) Inserire Asse Z ISOPP e settare le seguenti Proprietà

Oggetti → Iso\_Ns → IsoPP.vco



PP3	
Property	Events
Property	Value
Nome	PP3
Left	115
Top	88
Indice asse ISO	2
Nome processo	ISOV1
NGM-13 Channel	2

8) Inserire ISO I/O

Oggetti → Iso\_Ns → Iso-IO.vco



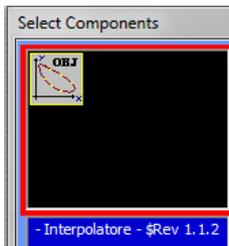
8) Settare le seguenti Proprietà



PP3	
Property	Events
Property	Value
Nome	PP3
Left	115
Top	88
Indice asse ISO	2
Nome processo	ISOV1
NGM-13 Channel	2

9) Inserire Oggetto ObjInterpola

Oggetti → Motor Control → CobjInterpola.vco



10) Settare le seguenti Proprietà



Interpola1	
Property	Events
Property	Value
Nome	Interpola 1
Left	65
Top	110
N.assi	3
N.tratti	16
Vper	1024
Div. Vper	1024
Abilita arcto	1

11) Inserire nella Task main (o task plc) la gestione del codice

```

Codice TASK PLC
Init Task PLC  Task PLC
1  '-----
2  ' Legge ingresso analogico
3  ' per controllo override
4  '-----
5  ISOV1_vper=ng_adc(0)
6  '-----
7  ' Test ingressi digitali
8  '-----
9  ISOV1_ext_fcz_x=!ISOV1.inp0  ' Homing switch X
10 ISOV1_ext_fcz_y=!ISOV1.inp1  ' Homing switch Y
11 ISOV1_ext_fcz_z=!ISOV1.inp2  ' Homing switch Z
12 ISOV1_ext_emcy=!ISOV1.inp3  ' EMERGENZA
13 ISOV1_ext_jogp_x=ISOV1.inp4  ' JOG X +
14 ISOV1_ext_jogm_x=ISOV1.inp5  ' JOG X -
15 ISOV1_ext_jogp_y=ISOV1.inp6  ' JOG Y +
16 ISOV1_ext_jogm_y=ISOV1.inp7  ' JOG Y -
17 ISOV1_ext_jogp_z=ISOV1.inp8  ' JOG Z +
18 ISOV1_ext_jogm_z=ISOV1.inp9  ' JOG Z -
19 '-----
20 ' Test Uscite digitali
21 '-----
22 ISOV1.out0=ISOV1_status_enable_x  ' X enabled
23 ISOV1.out1=ISOV1_status_enable_y  ' Y enabled
24 ISOV1.out2=ISOV1_status_enable_z  ' Z enabled
25 ISOV1.out2=ISOV1_status_error  ' CNC error

```

```

'-----
' Legge ingresso analogico
' per controllo override
'-----
ISOV1_vper=ng_adc(0)
'-----
' Test ingressi digitali
'-----
ISOV1_ext_fcz_x=!ISOV1.inp0  ' Homing switch X
ISOV1_ext_fcz_y=!ISOV1.inp1  ' Homing switch Y
ISOV1_ext_fcz_z=!ISOV1.inp2  ' Homing switch Z
ISOV1_ext_emcy=!ISOV1.inp3  ' General EMERGENZA
ISOV1_ext_jogp_x=ISOV1.inp4  ' JOG X +
ISOV1_ext_jogm_x=ISOV1.inp5  ' JOG X -
ISOV1_ext_jogp_y=ISOV1.inp6  ' JOG Y +
ISOV1_ext_jogm_y=ISOV1.inp7  ' JOG Y -
ISOV1_ext_jogp_z=ISOV1.inp8  ' JOG Z +
ISOV1_ext_jogm_z=ISOV1.inp9  ' JOG Z -
'-----
' Test Uscite digitali
'-----
ISOV1.out0=ISOV1_status_enable_x  ' X enabled
ISOV1.out1=ISOV1_status_enable_y  ' Y enabled
ISOV1.out2=ISOV1_status_enable_z  ' Z enabled
ISOV1.out2=ISOV1_status_error  ' CNC error

```

## 14.2 NG35+2xNGIO 3 Assi Analogici +/- 10V e Volantino Elettronico

Link ETHERNET IP: "10.0.0.80" (default)

Il seguente progetto usa un volantino elettronico connesso al Ch 2 seconda NGIO input encoder , Selettore per JOG Assi e Mandrino controllato in velocità.

Per abilitare il Selettore è necessario inserire nella init TASK PLC il seguente codice:

ISOV1\_soft\_sel\_man=0 ' Abilita il selettore interno a VTB

### Digital Inputs

Input 1	→	Switch Home X (N.C.)
Input 2	→	Switch Home Y (N.C.)
Input 3	→	Switch Home Z (N.C.)
Input 4	→	EMERGENZA GENERALE (N.C.)
Input 5	→	Selettore JOG X (N.O.)
Input 6	→	Selettore JOG Y (N.O.)
Input 7	→	Selettore JOG Z (N.O.)
Input 8	→	Pulsante JOG - (N.O.)
Input 9	→	Pulsante JOG + (N.O.)
Input 10	→	Volantino Speed x1
Input 11	→	Volantino Speed x10
Input 12	→	Volantino Speed x100

### Analog Inputs

Inputs 1	→	Potenzimetro Override Assi
----------	---	----------------------------

Per Abilitare il potenziometro è necessario selezionare il pulsante "Ext OW" da interfaccia di IsoNs



### Digital Outputs

Output 1	→	X Asse Abiliato
Output 2	→	Y Asse Abiliato
Output 3	→	Z Asse Abiliato
Output 4	→	CNC Errore
Output 5	→	Mandrino start/stop
Output 6	→	Mandrino CW (M3)
Output 7	→	Mandrino CCW (M4)

### Assi Inputs

Encoder Ch 1 (prima NGIO)	→	Encoder Asse X
Encoder Ch 2 (prima NGIO)	→	Encoder Asse Y
Encoder Ch 1 (seconda NGIO)	→	Encoder Asse Z
Encoder Ch 2 (seconda NGIO)	→	Encoder Volantino

### Assi Outputs

Analog out 1 (prima NGIO)	→	X Speed +/-10V
Analog out 2 (prima NGIO)	→	Y Speed +/-10V
Analog out 1 (seconda NGIO)	→	Z Speed +/-10V
Analog out 2 (seconda NGIO)	→	Speed Mandrino

- 1) Aprire un nuovo progetto VTB e selezionare NG35  
Selezionare 1 Ms sample

Start Page:

Sample:  mS

Task Time:  x 1 =  mS

Screensave:  Enable  sec

- 2) nserire un Oggetto ISOVIRTUAL e lasciare le Proprietà default

Oggetti → Iso\_Ns → IsoVirtual.vco



- 3) Inserire Asse X ISOPID

Oggetti → Iso\_Ns → IsoPid.vco



- 4) Settare le seguenti Proprietà

PID1	
Property	Events
Property	Value
Nome	PID1
Left	60
Top	10
Indice asse ISO	0
Nome processo	ISOV1
Indice asse NG-IO	0

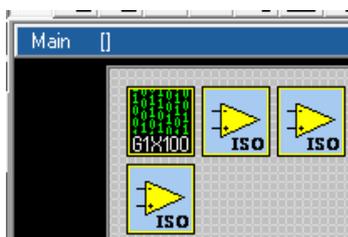
- 5) Inserire Asse Y ISOPID e Settare le seguenti Proprietà

Oggetti → Iso\_Ns → IsoPid.vco

PID2	
Property	Events
Property	Value
Nome	PID2
Left	100
Top	10
Indice asse ISO	1
Nome processo	ISOV1
Indice asse NG-IO	0

6) Inserire Asse Z ISOPID e Settare le seguenti Proprietà

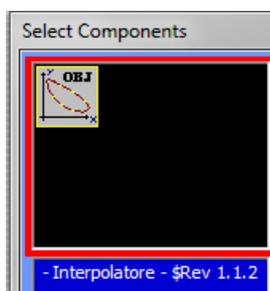
Oggetti → Iso\_Ns → IsoPid.vco



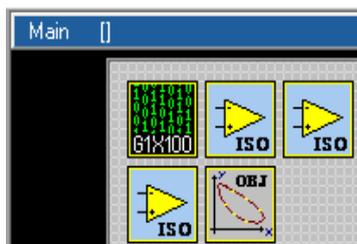
PID3	
Property	Events
Property	Value
Nome	PID3
Left	10
Top	55
Indice asse ISO	2
Nome processo	ISOV1
Indice asse NG-IO	1

7) Inserire Oggetto ObjInterpola

Oggetti → Motor Control → CobjInterpola.vco



8) Settare le seguenti Proprietà



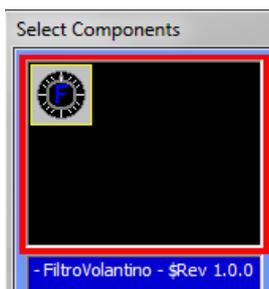
Interpola1	
Property	Events
Property	Value
Nome	Interpola 1
Left	65
Top	110
N.assi	3
N.tratti	16
Vper	1024
Div. Vper	1024
Abilita arcto	1

9) Dichiarare le seguenti Variabili Globali

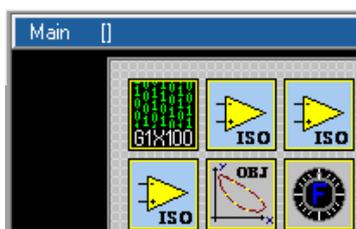
Internal VAR	Bit VAR	Define	Static VAR	VSD VAR	Fixed VA
			No	EXP	<input type="checkbox"/>
Variable	Type	Shared	Export in Class		
EncoderInput	LONG	No			
EncoderOut	LONG	No			

**10) Inserire Oggetto FiltroVol per gestione volante**

Oggetti → Motor Control → CfiltroVol.vco



**11) Settare le seguenti proprietà**



FiltroVol1	
Property	Events
Property	Value
Nome	FiltroVol1
Left	100
Top	55
Numero elementi	10
Molt. filtro	100
Encoder	EncoderInput
Variabile	EncoderOut

**12) Inserire in "Init task PLC " l' entry point per Funzion M a abilitare selettore VTB**

```

TASK PLC Code
Init Task PLC Task PLC
1 ISOV1_soft_sel_man=0 ' Enable the internal VTB selector
2 ISOV1_Start_m=StartMacro ' entry point MACRO management
    
```

**13) Inserire 2 Define in Variabili Globali → Define**

MAXSPEEDSPINDLE dipende da Massimo rpm a 10 Volt del Mandrino

Internal VAR	Bit VAR	Define	Static VAR
Variable		Type	
MAXDAC			2047
MAXSPEEDSPINDLE			24000

14) Inserire in Task Main Funzioni di Pagina la gestione M

```

Init Pagina | Master Event | Master Ciclo | Funzioni di Pagina
1 function StartMacro() as char
2 dim VelSpindle as long
3
4     ISOV1_m_ACK=1
5     select ISOV1_M_cmd
6         case 3
7             ISOV1.out5=true           ' start Mandrino in CW
8             ISOV1.out6=false         ' set CW
9             ' calcola la Speed
10            VelSpindle=(ISOV1_generic(9)*MAXDAC)/MAXSPEEDSPINDLE
11            ng_dac(3,VelSpindle)     ' Aggiorna la Speed al Dac
12            ISOV1.out4=true           ' Start Mandrino
13            ISOV1_status_m_run=0
14        case 4
15            ISOV1.out5=false          ' start Mandrino in CCW
16            ISOV1.out6=true           ' reset CW
17            ' calcola la Speed
18            VelSpindle=(ISOV1_generic(9)*MAXDAC)/MAXSPEEDSPINDLE
19            ng_dac(3,VelSpindle)     ' set CCW
20            ISOV1.out4=true           ' calcola la Speed
21            ISOV1_status_m_run=0     ' Aggiorna la Speed al Dac
22        case 5
23            ISOV1.out4=false          ' Stop Mandrino
24            VelSpindle=0              ' Stop Mandrino
25            ng_dac(3,VelSpindle)     ' set Speed a 0
26            ISOV1_status_m_run=0     ' Aggiorna la Speed al Dac
27        case else
28            ISOV1_m_ACK=0
29    endselect
30 endfunction
    
```

```

function StartMacro() as char
dim VelSpindle as long
    
```

```

ISOV1_m_ACK=1
select ISOV1_M_cmd
    case 3
        ISOV1.out5=true           ' start Mandrino in CW
        ISOV1.out6=false         ' set CW
        ' calcola la Speed
        VelSpindle=(ISOV1_generic(9)*MAXDAC)/MAXSPEEDSPINDLE
        ng_dac(3,VelSpindle)     ' Aggiorna la Speed al Dac
        ISOV1.out4=true           ' Start Mandrino
        ISOV1_status_m_run=0
    case 4
        ISOV1.out5=false          ' start Mandrino in CCW
        ISOV1.out6=true           ' reset CW
        ' calcola la Speed
        VelSpindle=(ISOV1_generic(9)*MAXDAC)/MAXSPEEDSPINDLE
        ng_dac(3,VelSpindle)     ' set CCW
        ISOV1.out4=true           ' calcola la Speed
        ISOV1_status_m_run=0     ' Aggiorna la Speed al Dac
    case 5
        ISOV1.out4=false          ' Stop Mandrino
        VelSpindle=0              ' Stop Mandrino
        ng_dac(3,VelSpindle)     ' set Speed a 0
        ISOV1_status_m_run=0     ' Aggiorna la Speed al Dac
    case else
        ISOV1_m_ACK=0
    
```

```

endselect
endfunction
    
```

15) Inserire in Task Main (Master Ciclo) o Task PLC la chiamata alle funzioni di gestione I/O

Page Init	Master Event	Master Cycle	Page Functions
1	AssiHoming()	' Controlla homing Switch	
2	GetEMCY()	' Prende EMERGENZA stato	
3	AssiManualJog()	' Controlla JOG	
4	SpeedHeWheel()	' Controlla Speed Volantino	
5	GetOverride()	' Legge potenziometri di Override	
6	SetOutputs()	' Set Digital outputs	

```

AssiHoming()      ' Controlla homing Switch
GetEMCY()         ' Prende EMERGENZA stato
AssiManualJog()  ' Controlla JOG
SpeedHeWheel()   ' Controlla Speed Volantino
GetOverride()    ' Legge potenziometri di Override
SetOutputs()     ' Set Digital outputs
    
```

16) Inserire nelle funzioni Task Main (Funzioni di Pagina)

Page Init	Master Event	Master Cycle	Page Functions
1	'*****		
2	' Controlla switch Assi homing		
3	'*****		
4	function AssiHoming() as void		
5	ISOV1_ext_fcx_x=!ISOV1.inp0	' Homing switch X	
6	ISOV1_ext_fcx_y=!ISOV1.inp1	' Homing switch Y	
7	ISOV1_ext_fcx_z=!ISOV1.inp2	' Homing switch Z	
8	endfunction		

```

*****
' Controlla switch Assi homing
*****
function AssiHoming() as void
    ISOV1_ext_fcx_x=!ISOV1.inp0 ' Homing switch X
    ISOV1_ext_fcx_y=!ISOV1.inp1 ' Homing switch Y
    ISOV1_ext_fcx_z=!ISOV1.inp2 ' Homing switch Z
endfunction
*****
' Controlla EMERGENZA Generale
*****
function GetEMCY() as void
    ISOV1_ext_emcy=!ISOV1.inp3 ' EMERGENZA Generale
endfunction
*****
' Controlla JOG
*****
function AssiManualJog() as void
    if ISOV1.inp4 ' Set Asse X
        ISOV1_asse_man=0
    endif
    if ISOV1.inp5 ' Set Asse Y
        ISOV1_asse_man=1
    endif
    if ISOV1.inp6 ' Set Asse Z
        ISOV1_asse_man=2
    endif
endfunction
    
```

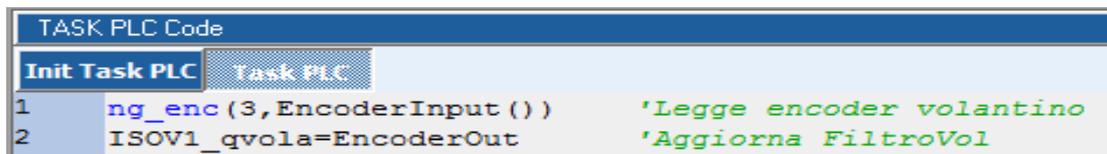
```

endif
ISOV1_ext_jogp=ISOV1.inp8 'Aggiorna Jog Input +
ISOV1_ext_jogm=ISOV1.inp7 'Aggiorna Jog Input -
endfunction
*****
' Controlla Manuale JOG
' L' aggiornamento encoder volante è in Task PLC
*****
function SpeedHeWheel() as void
    if ISOV1.inp9          'Moltiplicatore x1
        ISOV1_msofv=1
    endif
    if ISOV1.inp10       'Moltiplicatore x10
        ISOV1_msofv=10
    endif
    if ISOV1.inp11      'Moltiplicatore x100
        ISOV1_msofv=100
    endif
endfunction
*****
' Controlla Potenzimetri di Override
*****
function GetOverride()as void
    ISOV1_vper=ng_adc(0)
endfunction
*****
' Set digital outputs
*****
function SetOutputs()as void
    ISOV1.out0=ISOV1_status_enable_x      ' X Abilitato
    ISOV1.out1=ISOV1_status_enable_y      ' Y Abilitato
    ISOV1.out2=ISOV1_status_enable_z      ' Z Abilitato
    ISOV1.out3=ISOV1_status_error        ' CNC errore
endfunction

```

**17) Inserire nel Task PLC l' aggiornamento encoder volante**

Questa funzione deve essere sincrona con il Task PLC



```

ng_enc(3,EncoderInput())      'Legge encoder volante
ISOV1_qvola=EncoderOut        'Aggiorna FiltroVol

```

### 14.3 NG35+1xNGIO 3 Assi CanOpen

Link ETHERNET IP: "10.0.0.80" (default)

Il seguente progetto usa Assi CanOpen con Drives ESTUN

#### Digital Inputs

Input 1	→	Switch Home X (N.C.)
Input 2	→	Switch Home Y (N.C.)
Input 3	→	Switch Home Z (N.C.)
Input 4	→	EMERGENZA GENERALE (N.C.)
Input 5	→	Pulsante JOG X+ (N.O.)
Input 6	→	Pulsante JOG X- (N.O.)
Input 7	→	Pulsante JOG Y+ (N.O.)
Input 8	→	Pulsante JOG Y- (N.O.)
Input 9	→	Pulsante JOG Z+ (N.O.)
Input 10	→	Pulsante JOG Z (N.O.)

#### Analog Inputs

Inputs 1 → Potenziometro Override Assi

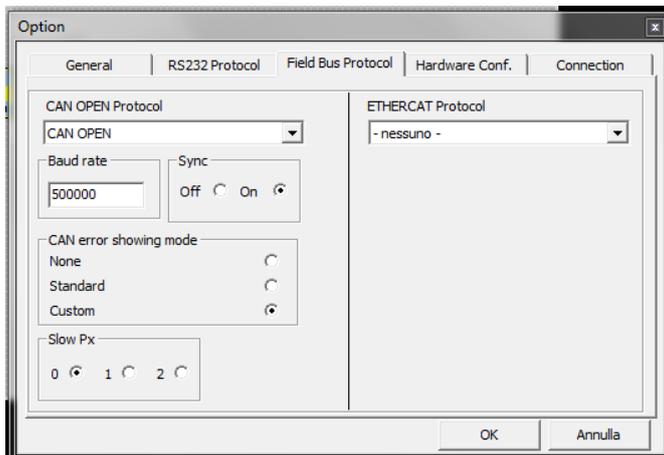
Per Abilitare il potenziometro è necessario selezionare il pulsante "Ext OW" da interfaccia di IsoNs



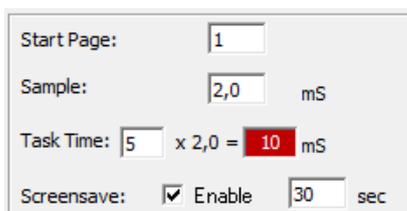
#### Digital Outputs

Output 1	→	X Asse Abilitato
Output 2	→	Y Asse Abilitato
Output 3	→	Z Asse Abilitato
Output 4	→	CNC Errore

#### 1) Aprire un nuovo progetto VTB e selezionare NG35 – Abilitare il CanOpen Fieldbus

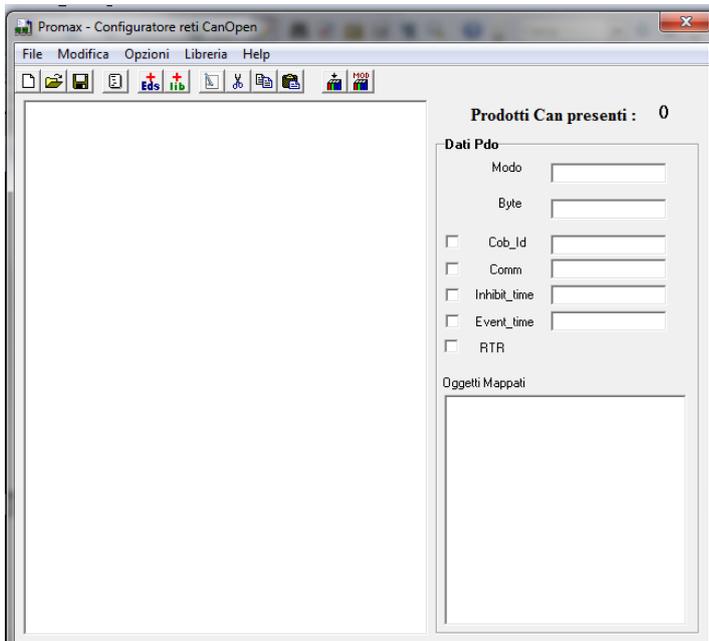


#### Selezionare 2 Ms sample

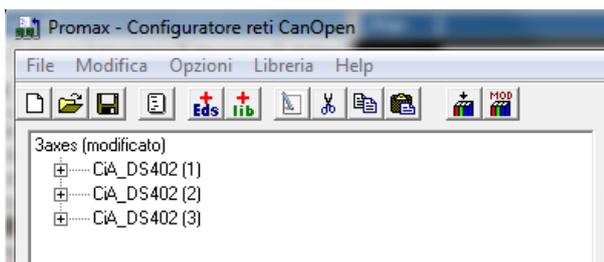
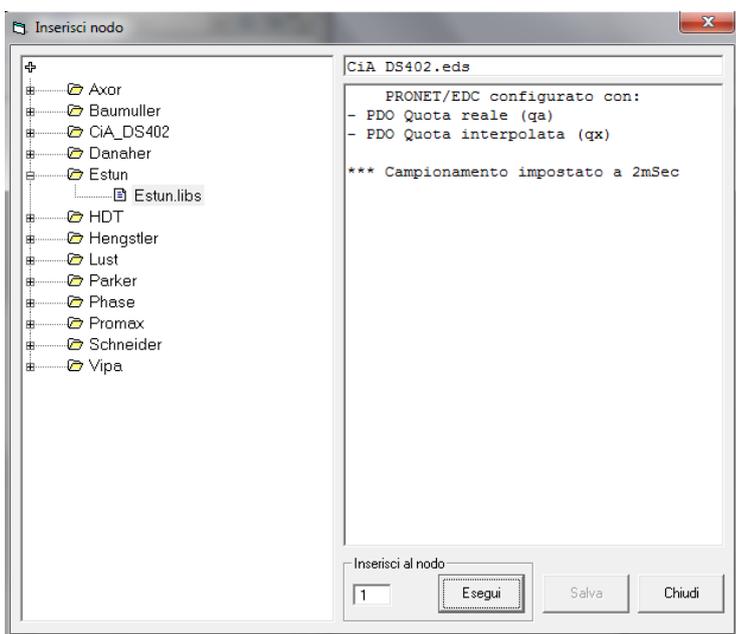


2) Prepare il file COP con il Configuratore CanOpen

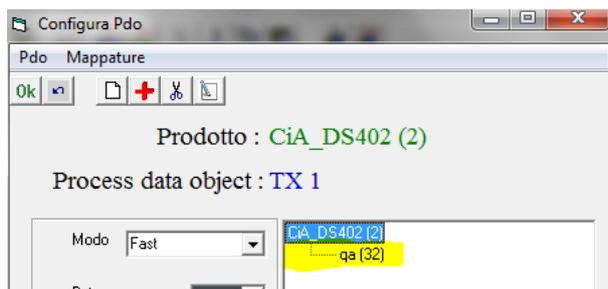
A) Aprire il Configuratore CanOpen



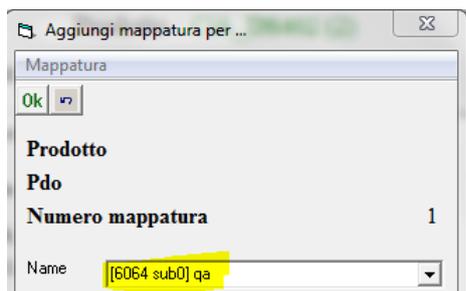
B) Aggiungere da Lib Drive tipo Estun e Inserire nodo 1 – Premere Pulsante “esegui”  
Ripetere per Nodo 2 e Nodo 3



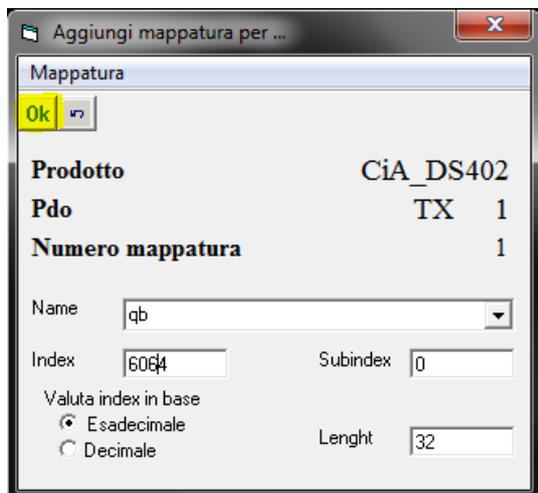
**C) Espandere il nodo 2 e double click su pdo\_Tx1 (Fast)**



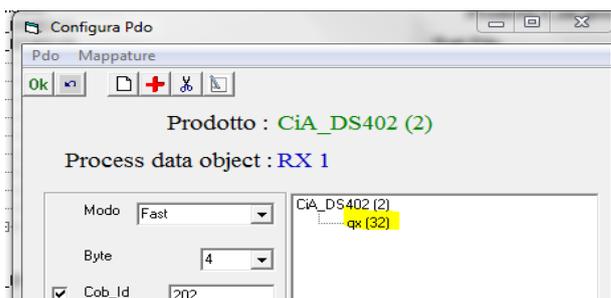
**D) Double click su qa(32)**



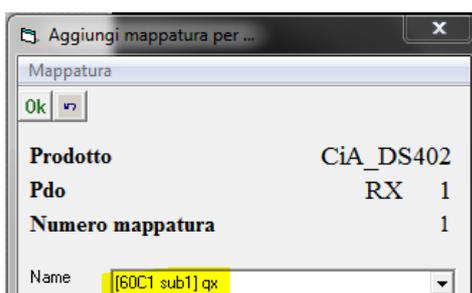
**Cancellare tutto il testo in Name – (6064 sub0) qa e Inserire solo il testo Qb e premere il Pulsante Ok**



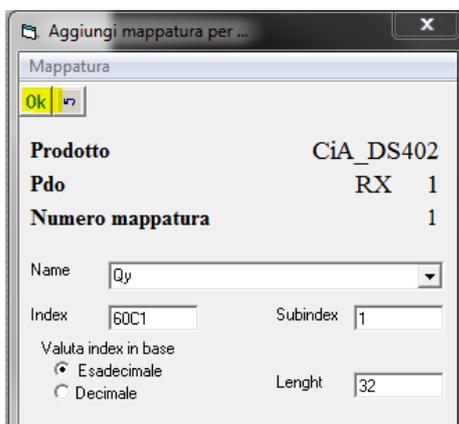
**E) Double click su pdo\_Rx1 (fast)**



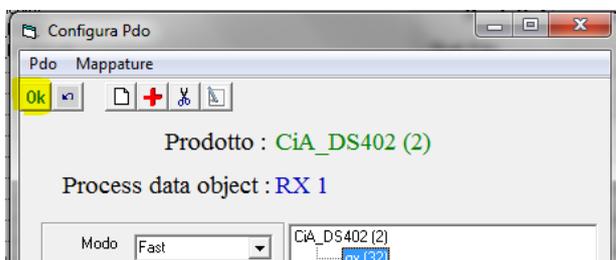
**F) Double click on qx(32)**



**Cancelare tutto il testo in Name– (60C1 sub1) qx e Inserire solo il testo Qy e premere il Pulsante Ok**

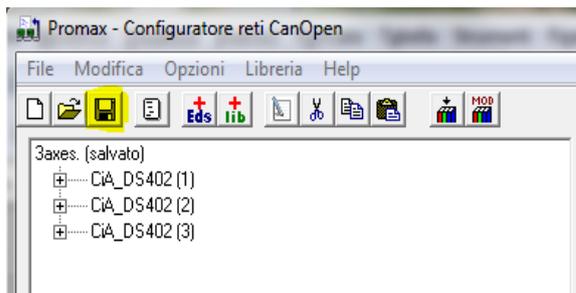


**G) Premere Ok**



**Ripetere in punti da C a G per nodo 3 inserendo Qc(Qb) e Qz(Qy)**

**H) Premere Save per salvare la configurazione**



Adesso la configurazione Assi CanOpen è pronta.  
I Drives CanOpen devono essere settati nel seguente Modo:

**X Asse**            **Nodo 1**            **Baud 500 Kb**  
**Y Asse**            **Nodo 2**            **Baud 500 Kb**  
**Z Asse**            **Nodo 3**            **Baud 500 Kb**

**3) Inserire Oggetto ISOVIRTUAL e lasciare le Proprietà di default**

Oggetti → Iso\_Ns → IsoVirtual.vco

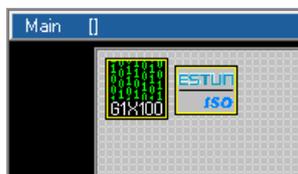


**4) Inserire Asse X ISOCanOpen**

Oggetti → Iso\_Ns → IsoCanOpen.vco



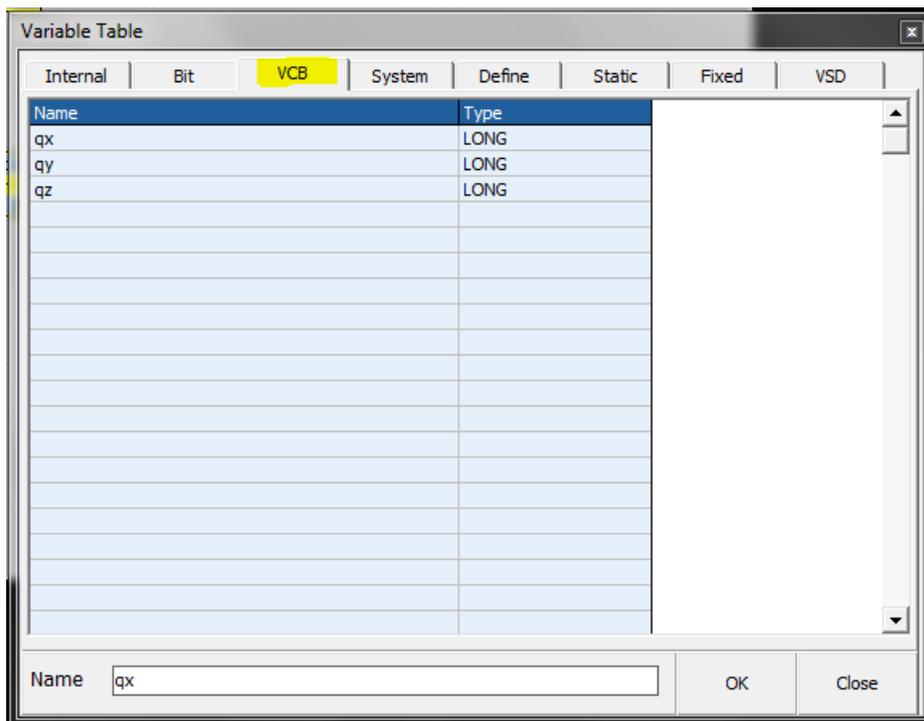
**5) Settare le seguenti Proprietà**



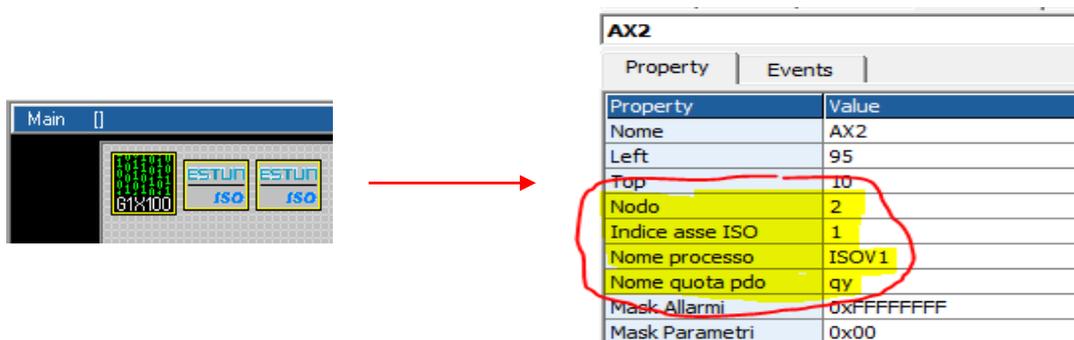
AX1	
Property	Events
Nome	AX1
Left	50
Top	10
Nodo	1
Indice asse ISO	0
Nome processo	ISOV1
Nome quota pdo	qx
Mask Allarmi	0xFFFFFFFF
Mask Parametri	0x00

**Nota:**

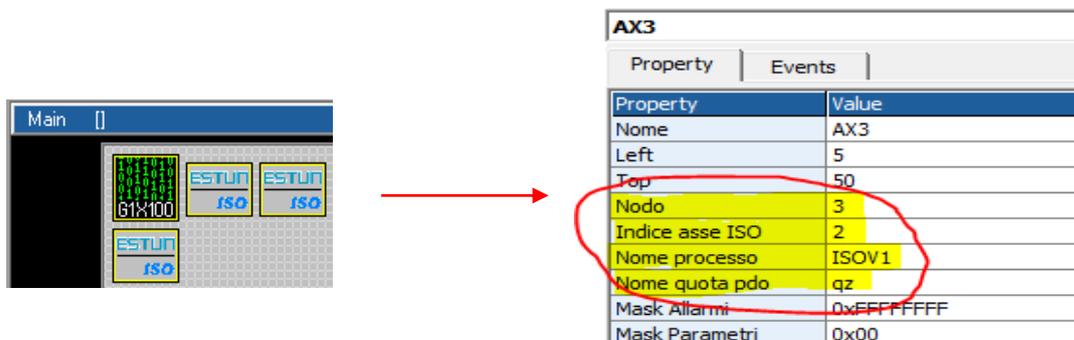
Per selezionare i PDO QX, QY e QZ, occorre avere prima creato la configurazione Canopen.  
Quando si fa Double Click su "Nome quota pdo", si apre la seguente finestra di inserimento variabili.  
Selezionare VCB tab e scegliere la variabile facendo double click sul nome



**6) Inserire Asse Y ISOCanOpen e Settare le seguenti Proprietà**  
*Oggetti* → *Iso\_Ns* → *IsoCanOpen.vco*

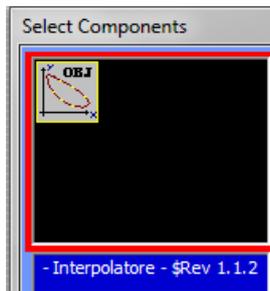


**7) Inserire Asse Z ISOCanOpen e Settare le seguenti Proprietà**  
*Oggetti* → *Iso\_Ns* → *IsoCanOpen.vco*



8) Inserire l' Oggetto ObjInterpola

Oggetti → Motor Control → CobjInterpola.vco



9) Settare le seguenti Proprietà



Interpola1	
Property	Events
Nome	Interpola1
Left	65
Top	110
N.assi	3
N.tratti	16
Vper	1024
Div. Vper	1024
Abilita arcto	1

10) Inserire in Task main (o task plc) la gestione del codice

```

Codice TASK PLC
Init Task PLC  Task PLC
1      '-----
2      ' Legge ingresso analogico
3      ' per controllo override
4      '-----
5      ISOV1_vper=ng_adc(0)
6      '-----
7      ' Test ingressi digitali
8      '-----
9      ISOV1_ext_fcz_x=!ISOV1.inp0  ' Homing switch X
10     ISOV1_ext_fcz_y=!ISOV1.inp1  ' Homing switch Y
11     ISOV1_ext_fcz_z=!ISOV1.inp2  ' Homing switch Z
12     ISOV1_ext_emcy=!ISOV1.inp3   ' EMERGENZA
13     ISOV1_ext_jogp_x=ISOV1.inp4  ' JOG X +
14     ISOV1_ext_jogm_x=ISOV1.inp5  ' JOG X -
15     ISOV1_ext_jogp_y=ISOV1.inp6  ' JOG Y +
16     ISOV1_ext_jogm_y=ISOV1.inp7  ' JOG Y -
17     ISOV1_ext_jogp_z=ISOV1.inp8  ' JOG Z +
18     ISOV1_ext_jogm_z=ISOV1.inp9  ' JOG Z -
19     '-----
20     ' Test Uscite digitali
21     '-----
22     ISOV1.out0=ISOV1.status_enable_x  ' X enabled
23     ISOV1.out1=ISOV1.status_enable_y  ' Y enabled
24     ISOV1.out2=ISOV1.status_enable_z  ' Z enabled
25     ISOV1.out2=ISOV1.status_error     ' CNC error

```

```

'-----
' Legge ingresso analogico
' per controllo override
'-----
ISOV1_vper=ng_adc(0)
'-----
' Test ingressi digitali
'-----
ISOV1_ext_fcz_x=!ISOV1.inp0  ' Homing switch X
ISOV1_ext_fcz_y=!ISOV1.inp1  ' Homing switch Y
ISOV1_ext_fcz_z=!ISOV1.inp2  ' Homing switch Z
ISOV1_ext_emcy=!ISOV1.inp3   ' General EMERGENZA
ISOV1_ext_jogp_x=ISOV1.inp4  ' JOG X +
ISOV1_ext_jogm_x=ISOV1.inp5  ' JOG X -
ISOV1_ext_jogp_y=ISOV1.inp6  ' JOG Y +
ISOV1_ext_jogm_y=ISOV1.inp7  ' JOG Y -
ISOV1_ext_jogp_z=ISOV1.inp8  ' JOG Z +
ISOV1_ext_jogm_z=ISOV1.inp9  ' JOG Z -
'-----
' Test Uscite digitali
'-----
ISOV1.out0=ISOV1.status_enable_x  ' X enabled
ISOV1.out1=ISOV1.status_enable_y  ' Y enabled
ISOV1.out2=ISOV1.status_enable_z  ' Z enabled
ISOV1.out2=ISOV1.status_error     ' CNC error

```

## 14.4 NGMEVO+NGQx (CanOpen) 3 Assi Step/Dir, Mandrino e Volantino

Link RS32 su COM1 NGMEVO

Il seguente progetto usa un Volantino Elettronico connesso sul Ch 1 encoder NGQx , Selettore per JOG Assi e Mandrino in Analogica output 1 NGQx. La scheda NGQx è in Link CanOpen con la scheda NGMEVO (Master)

Per Abilitare il Selettore è necessario Inserire nella init TASK PLC il seguente codice:

**ISOV1\_soft\_sel\_man=0 ' Abilita il selettore interno a VTB**

### Digital Inputs

Input 1 (NGMEVO ISOV1.inp0)	→	Switch Home X (N.C.)
Input 2 (NGMEVO ISOV1.inp1)	→	Switch Home Y (N.C.)
Input 3 (NGMEVO ISOV1.inp2)	→	Switch Home Z (N.C.)
Input 4 (NGMEVO ISOV1.inp3)	→	EMERGENZA GENERALE (N.C.)
.		
.		
Input 1 (NGQx ISOV1.inp16)	→	Selettore JOG X (N.O.)
Input 2 (NGQx ISOV1.inp17)	→	Selettore JOG Y (N.O.)
Input 3 (NGQx ISOV1.inp18)	→	Selettore JOG Z (N.O.)
Input 4 (NGQx ISOV1.inp19)	→	Pulsante JOG - (N.O.)
Input 5 (NGQx ISOV1.inp20)	→	Pulsante JOG + (N.O.)
Input 6 (NGQx ISOV1.inp21)	→	Volantino Speed x1
Input 7 (NGQx ISOV1.inp22)	→	Volantino Speed x10
Input 8 (NGQx ISOV1.inp23)	→	Volantino Speed x100

### Analog Inputs

Inputs 1 (NGQx) → Potenziometro Override Assi

Per Abilitare il potenziometro è necessario selezionare il pulsante “Ext OW” da interfaccia di IsoNs



### Digital Outputs

Output 1 (NGMEVO ISOV1.out0)	→	X Asse Abilitato
Output 2 (NGMEVO ISOV1.out1)	→	Y Asse Abilitato
Output 3 (NGMEVO ISOV1.out2)	→	Z Asse Abilitato
Output 4 (NGMEVO ISOV1.out3)	→	CNC Errore
.		
.		
Output 1 (NGQx ISOV1.out16)	→	Mandrino start/stop
Output 2 (NGQx ISOV1.out17)	→	Mandrino CW (M3)
Output 3 (NGQx ISOV1.out18)	→	Mandrino CCW (M4)

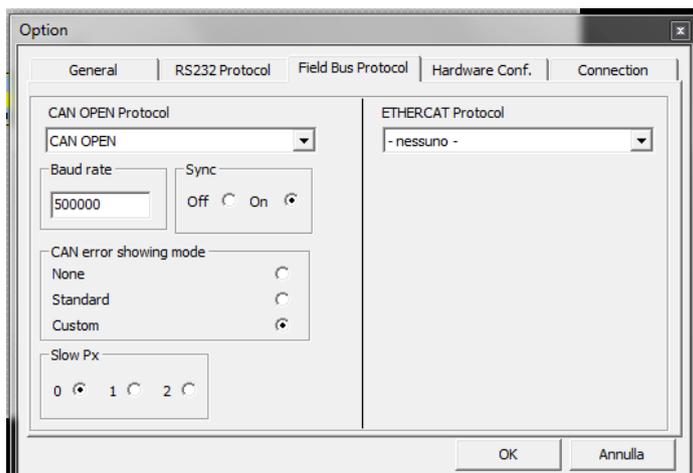
### Encoder Inputs

Encoder Ch 1 (NGQx) → Encoder Volantino

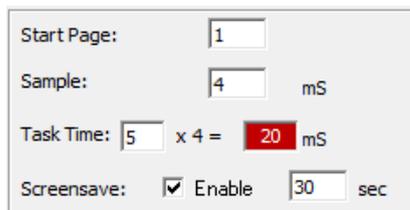
### Analog Outputs

Analog out 1 (NGQx) → SPEED Mandrino

1) Aprire un nuovo progetto VTB e selezionare NGMEVO – Abilitare il CanOpen

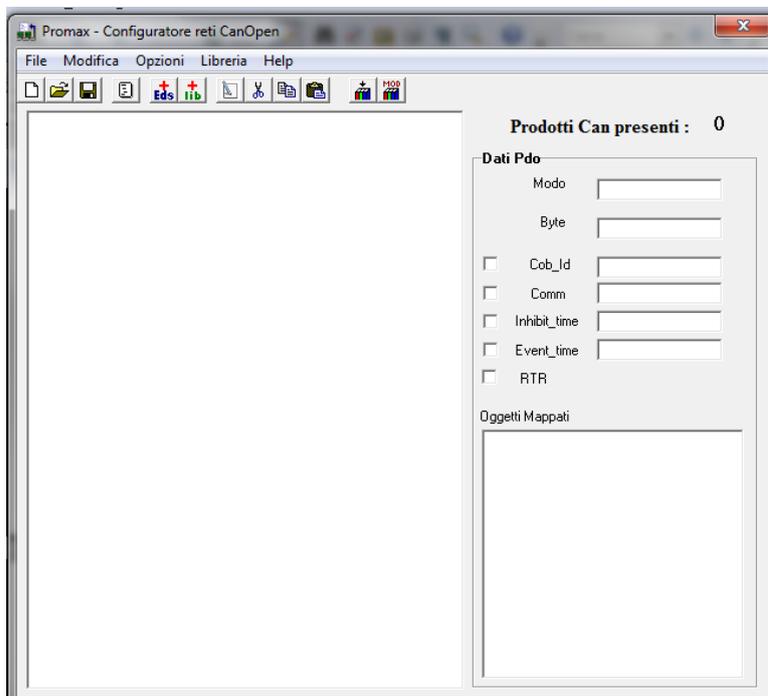


Selezionarea 4 Ms sample

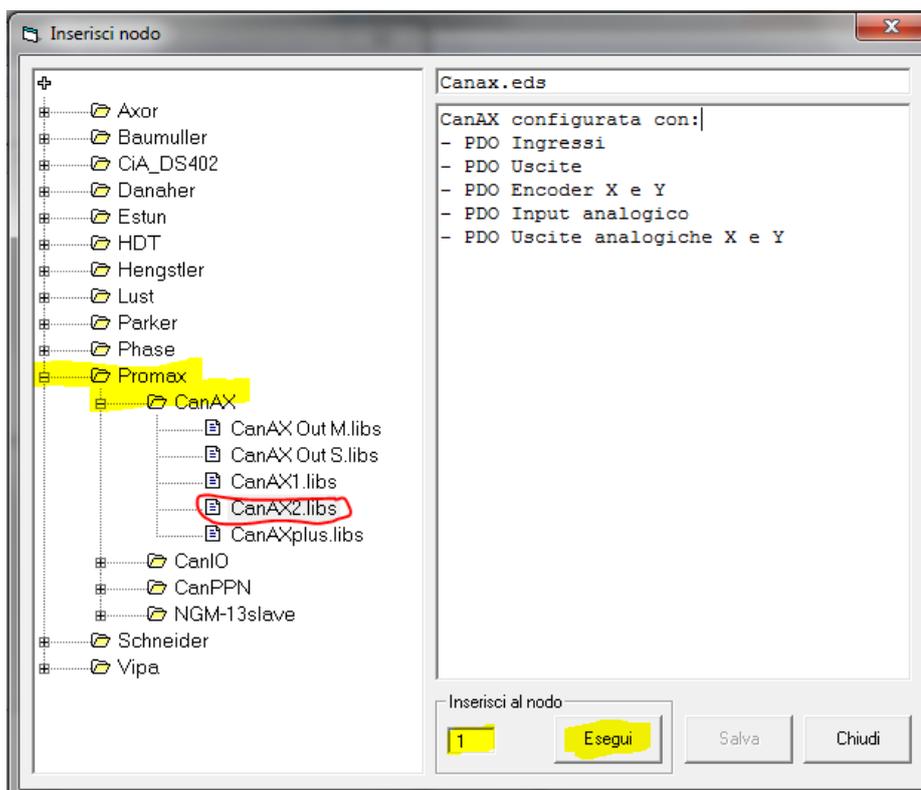


2) Preparare il file e COP con il configuratore CanOpen

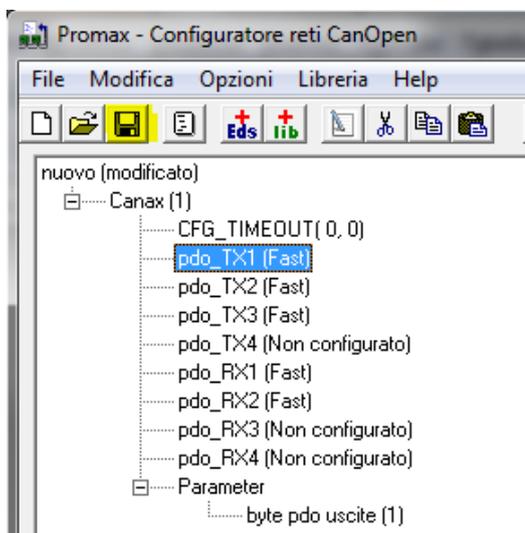
A) Aprire il configuratore CanOpen



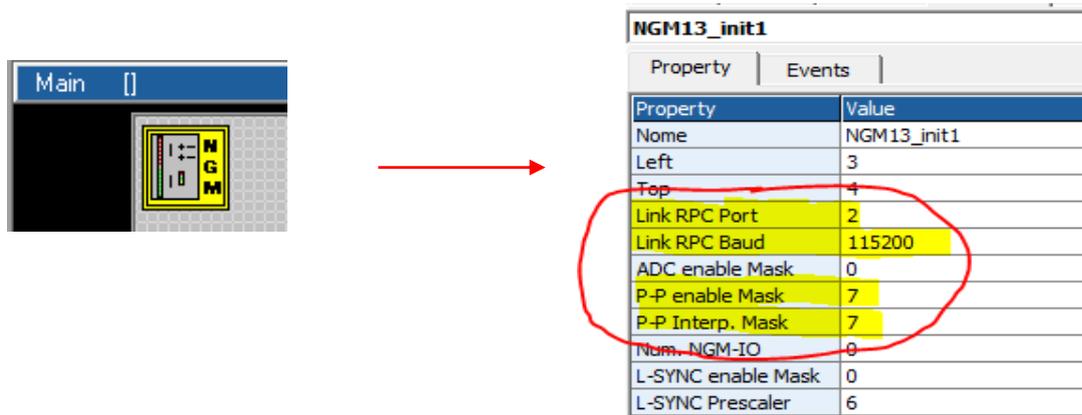
B) Inserire da Lib Promax tipo, Canax → CanAX2.lib e Inserire node 1 – Premere il Pulsante “esegui”



C) Premere il Pulsante Save



3) **Settare il link su COM2 NGMEVO e PP Interp mask a 7 (Canali X Y Z abilitati)**



NGM13_init1	
Property	Events
Nome	NGM13_init1
Left	3
Top	4
Link RPC Port	2
Link RPC Baud	115200
ADC enable Mask	0
P-P enable Mask	7
P-P Interp. Mask	7
Num. NGM-IO	0
L-SYNC enable Mask	0
L-SYNC Prescaler	6

4) **Inserire Oggetto ISOVIRTUAL e Lasciare le Proprietà di default**

Oggetti → Iso\_Ns → IsoVirtual.vco

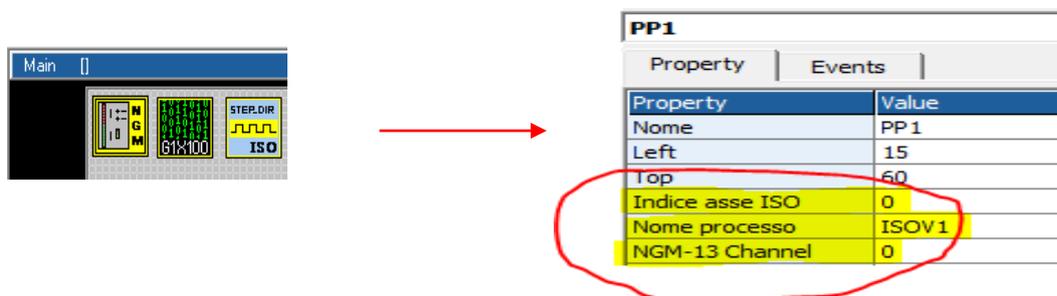


5) **Inserire Asse X ISOPP**

Oggetti → Iso\_Ns → IsoPP.vco



6) **Settare le seguenti Proprietà**



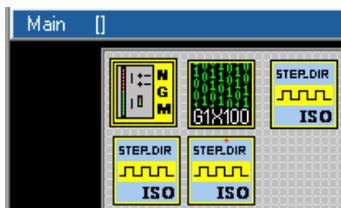
PP1	
Property	Events
Nome	PP1
Left	15
Top	60
Indice asse ISO	0
Nome processo	ISOV1
NGM-13 Channel	0

**7) Inserire Asse Y ISOPP e Settare le seguenti Proprietà**  
*Oggetti → Iso\_Ns → IsoPP.vco*



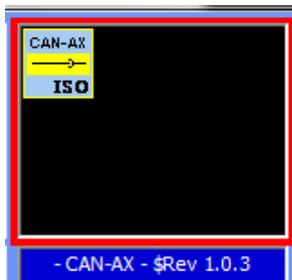
PP2	
Property	Events
Property	Value
Nome	PP2
Left	65
Top	60
Indice asse ISO	1
Nome processo	ISOV1
NGM-13 Channel	1

**8) Inserire Asse Z ISOPP e Settare le seguenti Proprietà**  
*Oggetti → Iso\_Ns → IsoPP.vco*



PP3	
Property	Events
Property	Value
Nome	PP3
Left	115
Top	88
Indice asse ISO	2
Nome processo	ISOV1
NGM-13 Channel	2

**9) Inserire ISO-IO Can-Ax (è lo stesso per NGQx)**  
*Oggetti → Iso\_Ns → ISO-IO.vco*



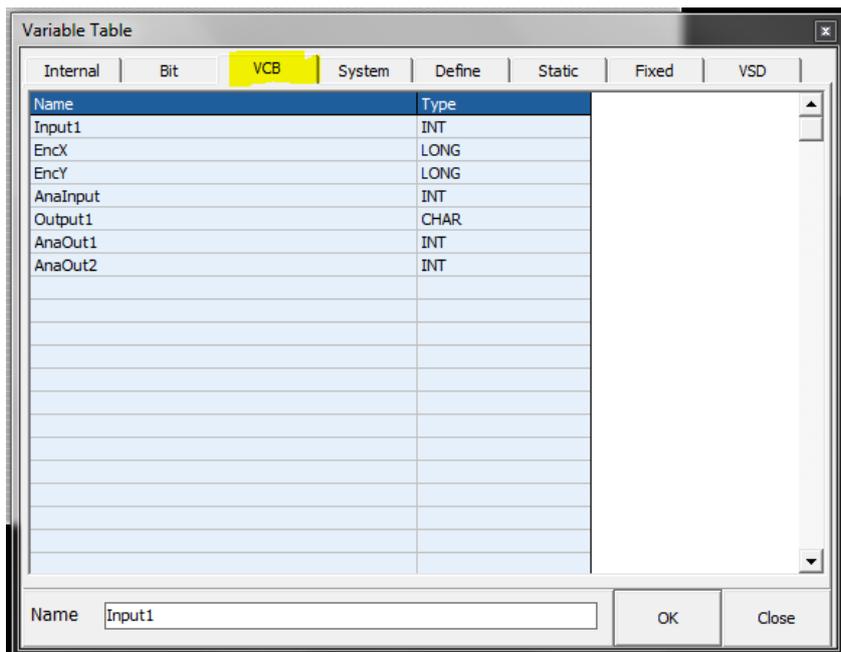
**10) Settare le seguenti Proprietà**



digio1	
Property	Events
Property	Value
Nome	digio 1
Nodo	0
Allarme cfg	41
Nome processo	ISOV1
Indice ISO-IO (16 bit)	1
Hardware enable	True
Variabile inp	Input1
Variabile out	Output1

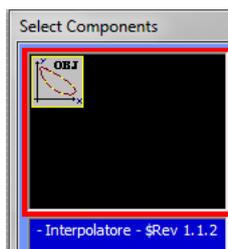
**Nota:**

Per selezionare le variabili Inp e Variabili Out, occorre avere prima creato la configurazione Canopen. Quando si fa Double Click su "Variabile Inp" o "Variabile out", si apre la seguente finestra di inserimento variabili. Selezionare VCB tab e scegliere la variabile facendo double click sul nome



**11) Inserire l' Oggetto ObjInterpola**

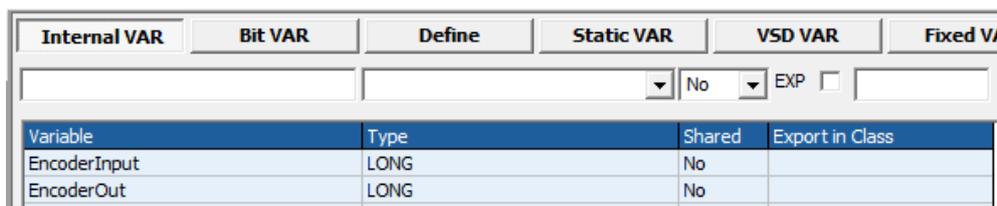
Oggetti → Motor Control → CobjInterpola.vco



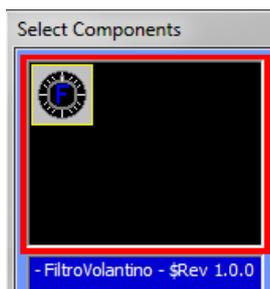
**12) Settare le seguenti Proprietà**



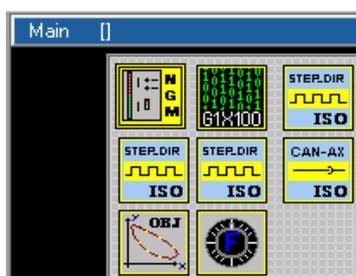
**13) Dichiarare le seguenti Variabili Globali**



**14) Inserire l' Oggetto FiltroVol per Volantino**  
**Oggetti → Motor Control → CfiltroVol.vco**



**15) Settare le seguenti Proprietà**



FiltroVol1	
Property	Events
Property	Value
Nome	FiltroVol1
Left	100
Top	55
Numero elementi	10
Molt. filtro	100
Encoder	EncoderInput
Variabile	EncoderOut

**16) Inserire in "Init task PLC " l' entry point per Funzion M a abilitare selettore VTB**

```

TASK PLC Code
Init Task PLC Task PLC
1 ISOV1_soft_sel_man=0 ' Enable the internal VTB selector
2 ISOV1_Start_m=StartMacro ' entry point MACRO management
    
```

**17) Inserire 2 Define in Variabili Globali → Define**

MAXSPEEDSPINDLE dipende da Massimo rpm a 10 Volt del Mandrino

Internal VAR	Bit VAR	Define	Static VAR
Variable		Type	
MAXDAC			2047
MAXSPEEDSPINDLE			24000

18) Inserire in Task Main Funzioni di Pagina la gestione M

```

Page Init | Master Event | Master Cycle | Page Functions
1 function StartMacro() as char
2 dim VelSpindle as long
3
4     ISOV1_m_ACK=1
5     select ISOV1_M_cmd
6         case 3             ' start spindle in CW
7             ISOV1.out17=true     ' set CW mode
8             ISOV1.out18=false    ' reset CCW mode
9             ' calculate the Speed
10            VelSpindle=(ISOV1_generic(9)*MAXDAC)/MAXSPEEDSPINDLE
11            ng_dac(3, VelSpindle) ' Update the Speed on Dac out
12            ISOV1.out16=true     ' Start spindle
13            ISOV1_status_m_run=0
14        case 4             ' start spindle in CCW
15            ISOV1.out17=false    ' reset CW mode
16            ISOV1.out18=true    ' set CCW mode
17            ' calculate the Speed
18            VelSpindle=(ISOV1_generic(9)*MAXDAC)/MAXSPEEDSPINDLE
19            ng_dac(3, VelSpindle) ' Update the Speed on Dac out
20            ISOV1.out16=true     ' Start spindle
21            ISOV1_status_m_run=0
22        case 5             ' Stop spindle
23            ISOV1.out16=false    ' Stop spindle
24            VelSpindle=0        ' set Speed to 0
25            ng_dac(3, VelSpindle) ' Update the Speed on Dac out
26            ISOV1_status_m_run=0
27        case else
28            ISOV1_m_ACK=0
29    endselect
30 endfunction
    
```

**function StartMacro() as char**  
**dim VelSpindle as long**

```

ISOV1_m_ACK=1
select ISOV1_M_cmd
    case 3             ' start Mandrino in CW
        ISOV1.out17=true     ' set CW
        ISOV1.out18=false    ' reset CCW
        ' calcola la Speed
        VelSpindle=(ISOV1_generic(9)*MAXDAC)/MAXSPEEDSPINDLE
        AnaOut1=VelSpindle  ' Aggiorna la Speed sul PDO
        ISOV1.out16=true     ' Start Mandrino
        ISOV1_status_m_run=0
    case 4             ' start Mandrino in CCW
        ISOV1.out17=false    ' reset CW
        ISOV1.out18=true    ' set CCW
        ' calcola la Speed
        VelSpindle=(ISOV1_generic(9)*MAXDAC)/MAXSPEEDSPINDLE
        AnaOut1=VelSpindle  ' Aggiorna la Speed sul PDO
        ISOV1.out16=true     ' Start Mandrino
        ISOV1_status_m_run=0
    case 5             ' Stop Mandrino
        ISOV1.out16=false    ' Stop Mandrino
        VelSpindle=0        ' set Speed a 0
        AnaOut1=VelSpindle  ' Aggiorna la Speed sul PDO
        ISOV1_status_m_run=0
    case else
        ISOV1_m_ACK=0
endselect
endfunction
    
```

19) Inserire in Task Main (Master Ciclo) o Task PLC la chiamata alle funzioni di gestione I/O

Page Init	Master Event	Master Cycle	Page Functions
1	AssiHoming()		' Controlla homing Switch
2	GetEMCY()		' Prende EMERGENZA stato
3	AssiManualJog()		' Controlla JOG
4	SpeedHeWheel()		' Controlla Speed Volantino
5	GetOverride()		' Legge potenziometri di Override
6	SetOutputs()		' Set Digital outputs

```

AssiHoming()      ' Controlla homing Switch
GetEMCY()         ' Prende EMERGENZA stato
AssiManualJog()  ' Controlla JOG
SpeedHeWheel()   ' Controlla Speed Volantino
GetOverride()    ' Legge potenziometri di Override
SetOutputs()     ' Set Digital outputs
    
```

20) Inserire nelle funzioni Task Main (Funzioni di Pagina)

Page Init	Master Event	Master Cycle	Page Functions
1			'*****
2			' Controlla switch Assi homing
3			'*****
4			function AssiHoming() as void
5			ISOV1_ext_fcx_x=!ISOV1.inp0 ' Homing switch X
6			ISOV1_ext_fcx_y=!ISOV1.inp1 ' Homing switch Y
7			ISOV1_ext_fcx_z=!ISOV1.inp2 ' Homing switch Z
8			endfunction

```

'*****
' Controlla switch Assi homing
'*****
function AssiHoming() as void
    ISOV1_ext_fcx_x=!ISOV1.inp0 ' Homing switch X
    ISOV1_ext_fcx_y=!ISOV1.inp1 ' Homing switch Y
    ISOV1_ext_fcx_z=!ISOV1.inp2 ' Homing switch Z
endfunction
'*****
' Controlla EMERGENZA Generale
'*****
function GetEMCY() as void
    ISOV1_ext_emcy=!ISOV1.inp3 ' EMERGENZA Generale
endfunction
'*****
' Controlla JOG
'*****
function AssiManualJog() as void
    if ISOV1.inp16 ' Set Asse X
        ISOV1_asse_man=0
    endif
    if ISOV1.inp17 ' Set Asse Y
        ISOV1_asse_man=1
    endif
    if ISOV1.inp18 'Set Asse Z
    
```

```

        ISOV1_asse_man=2
    endif
    ISOV1_ext_jogp=ISOV1.inp20 'Aggiorna Jog Input +
    ISOV1_ext_jogm=ISOV1.inp19 'Aggiorna Jog Input -
endfunction
*****
' Controlla Manuale JOG
' L' aggiornamento encoder volante è in Task PLC
*****
function SpeedHeWheel() as void
    if ISOV1.inp21          'Moltiplicatore x1
        ISOV1_msofv=1
    endif
    if ISOV1.inp22          'Moltiplicatore x10
        ISOV1_msofv=10
    endif
    if ISOV1.inp23          'Moltiplicatore x100
        ISOV1_msofv=100
    endif
endfunction
*****
' Controlla Potenzimetri di Override su PDO
*****
function GetOverride()as void
    ISOV1_vper=AnalInput
endfunction
*****
' Set digital outputs
*****
function SetOutputs()as void
    ISOV1.out0=ISOV1_status_enable_x      ' X Abilitato
    ISOV1.out1=ISOV1_status_enable_y      ' Y Abilitato
    ISOV1.out2=ISOV1_status_enable_z      ' Z Abilitato
    ISOV1.out3=ISOV1_status_error         ' CNC errore
endfunction

*****
' Control Override potentiometer
' from PDO declare in configurator CanOpen
*****
function GetOverride()as void
    ISOV1_vper=AnalInput
endfunction

```

## 21) Inserire nel Task PLC l'aggiornamento encoder volante

Questa funzione deve essere sincrona con il Task PLC

TASK PLC Code	
Init Task PLC	Task PLC
1	EncoderInput=EncX      'read the HandWheel encoder from NGQx PDO EncX
2	ISOV1_qvola=EncoderOut 'Update the Handwheel from FiltroVol Object
3	

```

EncoderInput=EncX      'read the HeWheel encoder from NGQx PDO EncX
ISOV1_qvola=EncoderOut 'Update the Hewheel from FiltroVol Oggetto

```

## Sommario

1	PREFAZIONE .....	3
2	COMPONENTI BASE .....	3
2.1	Selezione Hardware .....	3
2.2	Inserire Oggetto <i>cobjinterpola</i> .....	3
2.3	Proprietà dell' oggetto <i>cobjinterpola</i> .....	4
	.....	4
2.4	Inserire Oggetto Iso Virtual.....	4
2.5	Proprietà di IsoVirtual .....	4
2.6	Inserire la Variabile Fixed0 .....	5
3	Inserire il tipo di Assi.....	5
3.1	Scelta del tipo di Assi.....	5
4	Inserire I/O.....	9
4.1	Scelta del tipo di I/O .....	9
5	Utilizzo delle funzioni M interne al CNC.....	10
5.1	Inizializzare in VTB l' entry point alla funzione ISOV1_start_m .....	10
5.2	Leggere i parametri della funzione M.....	11
5.3	Scrittura dei Parametri M da Applicazione VTB .....	11
5.4	M flags .....	12
5.5	Esempio M3 M4 M5 start/stop Mandrino.....	12
6	Standard I/O.....	13
7	Status Word.....	21
8	Gestione PLC I/O .....	22
8.1	Definizione bit Ingressi Digitali.....	22
8.2	Definizione bit Uscite Digitali .....	22
9	FEED Override Esterno .....	23
10	Volantino Elettronico .....	23
11	Parametri Macchina .....	27
11.1	Parametri Generali .....	27
11.2	Parametri Asse X.....	28
11.3	Parametri Custom.....	29
12	Allarmi mappati a Bit.....	30
13	Allarmi CanOpen .....	31
14	Esempi.....	32
14.1	NGQ-NGMEVO 3 Assi Step/Dir.....	32
14.2	NG35+2xNGIO 3 Assi Analogici +/- 10V e Volantino Elettronico.....	36

14.3	NG35+1xNGIO 3 Assi CanOpen.....	43
14.4	NGMEVO+NGQx (CanOpen) 3 Assi Step/Dir, Mandrino e Volantino .....	51