Sviluppo Oggetti per VTB

www.promax.it

Rev. 1.0.0

email:info@promax.it - internet:www.promax.it

1. Prefazione

Questo manuale, spiega come creare oggetti per VTB.
Un oggetto definisce una porzione di codice VTB riutilizzabile all' interno di altri progetti.
Un oggetto può essere duplicato e quindi anche tutto il codice all' interno di esso .

Utilizzare gli oggetti, semplifica la stesura di nuove applicazioni e garantisce la funzionalità di queste.
Un oggetto viene scritto in codice VTB, ma con alcuni accorgimenti che devono essere rispettati.

2. File della Classe

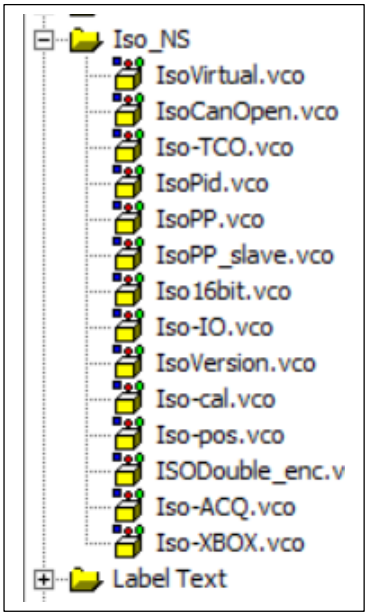
Gli oggetti vengono definiti in una file con estensione ***".vco"***
Per una corretta visualizzazione nel browser di VTB, gli oggetti devono essere raccolti in file con estensione ***".mco".***

File MCO
Contiene i file .VCO degli oggetti

$Rev 1.0.0	→ Versione del file MCO
$appPath$\Oggetti\nomefile1.vco	→ Definizione di un file .VCO visualizzato nel browser
$appPath$\Oggetti\nomefile2.vco	→ Definizione di un file .VCO visualizzato nel browser

Es: Iso_ns.mco

$Rev 1.0.9
$appPath$\Oggetti\IsoVirtual.vco
$appPath$\Oggetti\IsoCanOpen.vco
$appPath$\Oggetti\Iso-TCO.vco
$appPath$\Oggetti\IsoPid.vco
$appPath$\Oggetti\IsoPP.vco
$appPath$\Oggetti\IsoPP_slave.vco
$appPath$\Oggetti\Iso16bit.vco
$appPath$\Oggetti\Iso-IO.vco
$appPath$\Oggetti\IsoVersion.vco
$appPath$\Oggetti\Iso-cal.vco
$appPath$\Oggetti\Iso-pos.vco
$appPath$\Oggetti\ISODouble_enc.vco
$appPath$\Oggetti\Iso-ACQ.vco
$appPath$\Oggetti\Iso-XBOX.vco



Il file .MCO e i files .VCO devono trovarsi nella cartella:
Vtb\oggetti

Un file .VCO può contenere più oggetti al suo interno

3. Corpo di un Oggetto

L' oggetto viene definito in base a delle sezioni specifiche

HEADER OGGETTO
Codice Oggetto
Descrizione oggetto e revisione

HEADER PROPRIETA'
Definizione delle proprietà
FINE PROPRIETA'

HEADER GRAFICO
Definizione rappresentazione grafica
FINE HEADER GRAFICO

HEADER CODICE
Definizione Costanti
Elenco variabili
Fine definizione Costanti

Definizione Strutture
Elenco strutture
Fine definizione Strutture

Definizione variabili Globali
Elenco variabili
Fine definizione variabili Globali

Definizione variabili Bit
Elenco variabili
Fine definizione variabili Bit

Definizione variabili Oggetto
Elenco variabili
Fine definizione variabili Oggetto

Definizione variabili CanOpen SDO
Elenco variabili
Fine definizione variabili Canopen

INIT MAIN
Elenco codice
FINE INIT MAIN

HEADER TASK
TASK PLC
Elenco codice
FINE TASK PLC
Elenco codice in task main
Fine Header Task

Codice Funzioni globali

FINE HEADER CODICE

HEADER OGGETTO

Definisce l' inizio di un oggetto valore fisso:
#CODICE#

Codice Oggetto

Valore numerico è riferito solo agli oggetti all' interno del file.vco
Questo deve essere diverso per ogni oggetto es:
610

Descrizione oggetto e revisione

Descrizione dell' oggetto e revisione:
La revisione deve iniziare con il TAG **$Rev** e comprendere 3 cifre numeriche separate da un punto ***x.x.x***
- ISOVirtual - $Rev 2.3.1

HEADER PROPRIETA'

Definisce l' inizio delle proprietà oggetto valore fisso:
#PROPRIETA#

p1="Name",NomeOggetto,V,T
p2="Left",0,F,N → Posizione in X del frame grafico
p3="Top",0,F,N → Posizione in Y del frame grafico
p4="",40,F,N → Larghezza del frame grafico
p5="",40,F,N → Altezza del frame grafico
p6="",0,F,N → Posizione in X del Box grafico
p7="",0,F,N → Posizione in Y del Box grafico
p8="",40,F,N → Larghezza del box grafico
p9="",40,F,N → Altezza del box grafico
p10="",0,F,C → ForeColor del box grafico
p11="",6,F,C → BackColor del box grafico
p12="",2,F,N → Posizione in X dell' immagine
p13="",2,F,N → Posizione in Y dell' immagine
p14="",37,F,N → Larghezza dell' immagine
p15="",37,F,N → Altezza dell' immagine
p16="",nomebitmap.bmp,F,B → Bitmap di rappresentazione
la bitmap deve trovarsi nella cartella VTB\SystemBmp
Dimensione tipica 37x37 pixel

Tutti i valori sono fissi
Cambiare solo la proprietà P1 NomeOggetto
Inserire un nome valido senza spazi e caratteri strani.
Es: MyIso
Questo è il nome che identificherà tutte le proprietà.

p17… Da qui iniziano le Proprietà Custom e sono definite nel seguente modo:

Pn=”Descr”,Valdef,Vis,Type

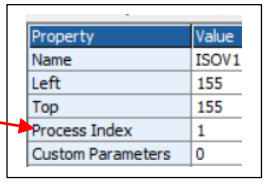
Descr descrizione della proprietà solo se visibile
La descrizione è quella che viene rappresentata nel browser

Valdef Valore di default della proprietà

Vis **V** Visibile nel Browser
F Invisibile

Type **N** Valore numerico
V Variabile di sistema (variabile di VTB tipo bit long ecc.)
T Testo

Es:
p17="Process Index",1,V,N



In pratica le proprietà possono sostituire il proprio valore nel codice VTB
Per far questo è sufficiente inserire il numero della proprietà tra due codice ? ..?.
Es: **?p17?**
In questo modo il TAG **?p17?** viene sostituito con il valore della proprietà P17

1) Assegnazione di una variabile con valore proprietà
   VarVtb=?pn? dove n è il numero della proprietà
   Es:
   P17=”Prop 17”,102,V,N
   VarVtb=?p17? var VTB assume il valore 102

2) Utilizzo del valore per la dichiarazione di una variabile
   In genere si utilizza il nome dell’oggetto, cioè la proprietà p1

```
BEGIN VAR
?p1?.enable as char
?p1?.kp as long
?p1?.ki as long
?p1?.kv as long
?p1?.err_sum as long
?p1?.err_sat as long
END VAR
```

In questo modo tutte le variabili dichiarate nell' oggetto, verranno legate al nome dell' oggetto stesso, e quindi diventeranno univoche per l' oggetto.
In pratica se il valore di p1 è:
p1="Name",MyObj,V,T

Tutte le variabili sopra elencate saranno nominate nel seguente modo
MyObj.*enable as char*
MyObj.*kp as long*
MyObj.*ki as long*
MyObj.*kv as long*

Pertanto sono legate al nome dell' oggetto.
Le proprietà non possono essere cambiate in ***run time***, pertanto se è necessario questo occorre passare da una variabile di appoggio.

FINE PROPRIETA'

Definisce la fine delle proprietà.
#END PROPRIETA#

HEADER GRAFICO

Definisce la rappresentazione grafica dell' oggetto nell' I.D.E. di VTB
Ovviamente questa rappresenta solamente una visualizzazione grafica.
E' consigliabile utilizzare sempre la ***rappresentazione classica***
#OGGETTI#

Rappresentazione Classica
frame(p2,p3,p4,p5)
box(p6,p7,p8,p9,p10,p11)
bmp(p12,p13,p14,p15,p16)

Questo tipo di rappresentazione, identifica un Oggetto con un quadrato con all' interno una BitMap.

FINE HEADER GRAFICO

Fine della rappresentazione Grafica
#END OGGETTI#

HEADER CODICE

Inizio dell' area del codice oggetto
#BEGIN CODE#

Definizione Costanti

Inizio della definizione delle costanti (DEFINE) utilizzate nel codice
Generalmente le DEFINE vengono legate al nome dell' oggetto
BEGIN DEFINE

Es: **?P1?_DEF_NASSI as 3**
?P1?_LEN_PROCESSO as 256

?P1? viene sostituita con il nome dell' oggetto

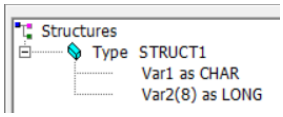
Fine definizione Costanti

Fine dell' area definizione costanti
END DEFINE

Definizione Strutture

Inizio della definizione delle strutture utilizzate nel codice VTB

BEGIN STRUCT



Es:

```
Type ?p1?.STRUCT1
Var1 as char
Var2(8) as long
endtype
Type ?p1?.STRUCT2
Var1(?P1?_DEF_NASSI) as LONG
Var2 as LONG
Var3 as FLOAT
endtype
```

Questa è una struttura

Questa è una struttura
?P1?_DEF_NASSI è una DEFINE

Fine definizione Strutture

Fine area dichiarazione strutture

END STRUCT

Definizione variabili Globali

Inizio area di definizione delle variabili globali utilizzate da VTB

BEGIN VAR

Es:

```
?p1?.var1 as char
?p1?.var2 as long
?p1?.var3 as float
?p1?.var3 as ?p1?.STRUCT1
```

Fine definizione variabili Globali

Fine della definizione variabili globali

END VAR

Definizione variabili Bit

Inizio area di definizione delle variabili BIT

BEGIN VAR BIT

Es:

?p1?.bit0 as ?P1?.var.0 → bit **0** della variabile **?P1?.var**

?p1?.bit1 as ?P1?.var.1 → bit **1** della variabile **?P1?.var**

?p1?.bit2 as ?P1?.var.2 → bit **2** della variabile **?P1?.var**

Fine definizione variabili Bit

Fine della definizione variabili BIT

END VAR BIT

Definizione variabili Oggetto

Inizio area di definizione variabili oggetto. **Sezione Obsoleta**

BEGIN VAR OBJ

Fine definizione variabili Oggetto

Fine della definizione variabili oggetto

END VAR OBJ

Definizione variabili CanOpen SDO

Inizio area di definizione variabili CanOpen (vedi il manuale vtb)
BEGIN VAR OBJ (questa sezione non è presente nell' IDE di VTB)

Es: è stata inserita una proprietà P17 che contiene il nodo dello slave CanOpen

```
p17="Nodo",1,F,N
```

?p1?.acc as LONG at 0x608300 id ?p17? → **Setta Accelerazione**
?p1?.dec as LONG at 0x608400 id ?p17? → **Setta Decelerazione**

Usare:
?p1?.acc=100 'Setta Accelerazione
?p1?.dec=50 'Setta Decelerazione

Fine definizione variabili Canopen

Fine area di definizione variabili CanOpen
END VAR OBJ

INIT MAIN

Inizio codice inserito nella ***"Init del TASK MAIN" (inizializzazioni)***
BEGIN $INIT

Es:
?P1?.var1=0
?P1?.var1=100

FINE INIT MAIN

Fine codice inserito nella init del TASK MAIN
END $INIT

HEADER TASK

Header inziale delle TASK Main e PLC
BEGIN $TASK
EVENTI_MASTER

TASK PLC

Inizio codice inserito nel ***TASK PLC***
Qui può essere scritto tutto il codice che poi verrà
Inserito nel TASK PLC
#BEGIN PLC#

Es:

```
if ?p1?.enable=1
        ng_enc(?p23?,?p1?.posr())
        ?p1?.tmp=(?p22?_qi(?p21?)-?p1?.posr)
        ?p22?_qr(?p21?)=?p1?.posr/?p22?_rap(?p21?)
        ?p22?_qerr(?p21?)=?p1?.tmp/?p22?_rap(?p21?)
        if ?p36?=1
                ?p1?.err_p=?p1?.kp*?p1?.tmp
        Endif
Endif
```

FINE TASK PLC

Fine codice inserito nel TASK PLC
Sotto questa sezione può essere inserito il codice nel Task Main
#END PLC#

ATTENZIONE IL CODICE DEL TASK MAIN NON HA UN HEADER PARTICOLARE, MA UTILIZZA IL TAG #END PLC#

Es:

```
if ISOV1_cfgerr(?p17?).n=10              'Test EMCY
        ISOV1_cfgerr(?p17?).n=0
        ?p1?.err_mot=(ISOV1_cfgerr(?p17?).code(7)&0xff)
        ?p1?.err_mot=?p1?.err_mot<<8
        ?p1?.err_mot=?p1?.err_mot|(ISOV1_cfgerr(?p17?).code(6)&0xff)
        ?p1?.err_mot=?p1?.err_mot<<8
        ?p1?.err_mot=?p1?.err_mot|(ISOV1_cfgerr(?p17?).code(5)&0xff)
        .
        .
Endif
```

Fine Header Task

Fine della sezione TASK
Dopo questa sezione possono essere inserite tutte le funzioni globali
END_EVENTI_MASTER
END $TASK

Codice Funzioni globali

Le funzioni globali vengono inserite dopo il tag END TASK

Es:

```
'-----------------------------------------------
'start_home
'
function ?p1?.start_home(hm as char) as void
?P1?.start=0
?P1?.modo=0
?P1?.homemode=hm
'?P1?.homeacc=1000000
?P1?.mode=6
?p1?.ctrl=?P1?.ctrl|0x10
?P1?_timer=get_timer()
while test_timer(?P1?_timer,100/TAU)=0
loop
endfunction
```

FINE HEADER CODICE

Fine dell' oggetto, sotto questa sezione può essere dichiarato un nuovo oggetto ripartendo dall' HEADER OGGETTO
#END CODE#

4. File base per creazione Oggetti

E' possibile utilizzare questo file come base per creare oggetti - Download File

```
#CODICE#
600
- OBJName - $Rev 1.0.0
#PROPRIETA#
p1="Nome",MyObj,V,T
p2="Left",0,F,N
p3="Top",0,F,N
p4="",40,F,N
p5="",40,F,N
p6="",0,F,N
p7="",0,F,N
p8="",40,F,N
p9="",40,F,N
p10="",0,F,C
p11="",6,F,C
p12="",2,F,N
p13="",2,F,N
p14="",37,F,N
p15="",37,F,N
p16="",MyObj.bmp,F,B
#END PROPRIETA#
#OGGETTI#
frame(p2,p3,p4,p5)
box(p6,p7,p8,p9,p10,p11)
bmp(p12,p13,p14,p15,p16)
#END OGGETTI#
#BEGIN CODE#
BEGIN DEFINE
END DEFINE
BEGIN STRUCT
END STRUCT
BEGIN VAR
END VAR
BEGIN VAR OBJ
END VAR OBJ
BEGIN VAR SDO
END VAR SDO
BEGIN VAR BIT
END VAR BIT
BEGIN $INIT
'**********************************
' Insert Here Task Main Init Code
'**********************************
END $INIT
BEGIN $TASK
EVENTI_MASTER
```

```
#BEGIN PLC#
'**********************************
' Insert Here Task PLC  Init Code
'**********************************
#END PLC#
'**********************************
' Insert Here Task Main code
'**********************************
END_EVENTI_MASTER
EVENTI_TASTI
END_EVENTI_TASTI
END $TASK
#EVENTI OGGETTO#
#END EVENTI OGGETTO#
'**********************************
' Insert Here VTB Functions
'**********************************

'==================================
#END CODE#
```

5. Esempio di un oggetto

Di seguito viene riportato un esempio di Oggetto di VTB

```
#CODICE#
609
- TEST CANOPEN - $Rev 1.0.0
#PROPRIETA#
p1="Nome",TCO,V,T
p2="Left",0,F,N
p3="Top",0,F,N
p4="",40,F,N
p5="",40,F,N
p6="",0,F,N
p7="",0,F,N
p8="",40,F,N
p9="",40,F,N
p10="",0,F,C
p11="",6,F,C
p12="",2,F,N
p13="",2,F,N
p14="",37,F,N
p15="",37,F,N
p16="",tco.bmp,F,B
p17="Nome processo",ISOV1,F,V
p18="LastNode",16,F,N
p19="Tscan",1000,F,N
p20="Index",0x6064,F,V
p21="Sub-Index",0,F,V
p22="",0x00,F,V
p23="",0,F,N
p24="",0,F,N
p25="",0,F,N
#END PROPRIETA#
#OGGETTI#
frame(p2,p3,p4,p5)
box(p6,p7,p8,p9,p10,p11)
bmp(p12,p13,p14,p15,p16)
#END OGGETTI#
#BEGIN CODE#
BEGIN DEFINE
ISOTCO as 0
END DEFINE
BEGIN VAR
?p1?.tscan as LONG
?p1?_tempo as LONG
?p1?_ax as INT
?p1?_val as LONG
?p1?.val(?p18?) as LONG
?p1?.stato(?p18?) as CHAR
?p1?_j as INT
END VAR
BEGIN VAR OBJ
END VAR OBJ
BEGIN VAR SDO
END VAR SDO
```

```
BEGIN $INIT
?p1?.tscan=?p19?/TAU
?p1?_tempo=?p1?.tscan
for ?p1?_j=0 to ?p1?_j<?p18?
	if ISOV1_cfgerr(?p1?_j+1).n
		?p1?.stato(?p1?_j)=1
	endif
next ?p1?_j
END $INIT
BEGIN $TASK
EVENTI_MASTER
#BEGIN PLC#
if ?p1?_tempo
	dec ?p1?_tempo
endif
#END PLC#
?p1?_scan()
END_EVENTI_MASTER
EVENTI_TASTI
END_EVENTI_TASTI
END $TASK
#EVENTI OGGETTO#
#END EVENTI OGGETTO#
function ?p1?_scan() as void
dim j as int
if ?p1?_tempo=0
	?p1?_tempo=?p1?.tscan
	j=0
	while ?p1?.stato(?p1?_ax)=0
		inc ?p1?_ax
		inc j
		if ?p1?_ax>=?p18?
			?p1?_ax=0
		endif
		if j>=?p18?
			return
		endif
	loop
	if pxco_sdoul(?p1?_ax+1,?p20?,?p21?,?p1?_val())
		?p17?_allarm(1)=?p17?_allarm(1)|(1<<?p1?_ax)
		?p17?_stop_emcy=1
	else
		?p1?.val(?p1?_ax)=?p1?_val
	endif
	inc ?p1?_ax
endif
endfunction
#END CODE#
```

Sommario

1. Prefazione 3
2. File della Classe 3
3. Corpo di un Oggetto 4
4. File base per creazione Oggetti 12
5. Esempio di un oggetto 14