VTB Objects Development

www.promax.it

Rev. 1.0.0

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

1. Preface

This document explain, how create objects for VTB.
An Object defining a portion of VTB code. It can be reused in others projects.
An Object can be duplicated, and automatically all the internal code is duplicated.

Use objects, simplifies the applications development.
An object, is writes by VTB code

2. Class File

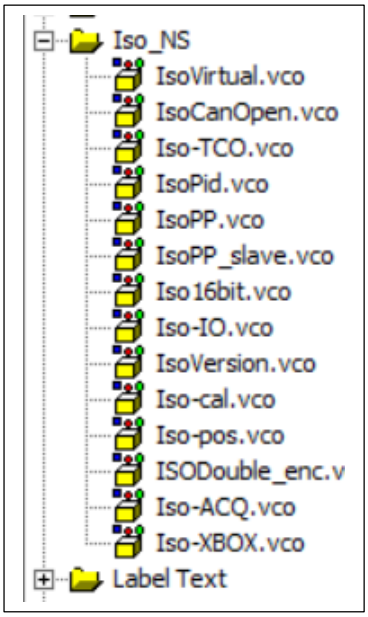
The objects are defined in a file with extension ***".vco"***
For a correct showing in the VTB browser, the objects, must be included in a file with extension ***".mco".***

File MCO
It contain, the VCO files

$Rev 1.0.0	→ Version MCO file
$appPath$\Oggetti\namefile1.vco	→ VCO File showed in the browser
$appPath$\Oggetti\namefile2.vco	→ VCO File showed in the browser

Ex: 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



The MCO files and the VCO files must be in the folder:
Vtb\oggetti

The VCO file can contain more objects

3. Object Body

The object is defined with following specifics

OBJECT HEADER
Object Code
Object Description and revision

PROPERTY HEADER
Property defining
END PROPERTY

GRAPHIC HEADER
Graphic defining
END GRAPHIC

VTB CODE HEADER
Constants Defining
Costants list
End Constants Defining

Structures Defining
Structures list
End Structures Defining

Global Variables Defining
Variables list
End Global Variables Defining

Bit Variables Defining
Bit list
End Bit Variables Defining

Objects Variables Defining
Objects list
End Objects Variables Defining

CanOpen SDO Variables Defining
SDO list
End CanOpen SDO Variables Defining

INIT MAIN
Code list
END INIT MAIN

HEADER TASK
TASK PLC
Code List
END TASK PLC
List code in Task Main
End Header Task

Global Functions List

END HEADER CODE

OBJECT HEADER

Object Start:
#CODICE#

Object Code
Numerical value. This value must be different for any object in the VCO ex:
610

Object Description and Revision
Object description
The revision must started with tag $Rev and must contain 3 numerical values separated with decimal point
- ISOVirtual - $Rev 2.3.1

PROPERTY HEADER

Defining properties:
#PROPRIETA#

p1="Name",ObjectName,V,T
p2="Left",0,F,N → Frame X position in the IDE (image)
p3="Top",0,F,N → Frame Y position in the IDE (image)
p4="",40,F,N → Frame width
p5="",40,F,N → Frame Height
p6="",0,F,N → Box X position
p7="",0,F,N → Box Y position
p8="",40,F,N → Box Width
p9="",40,F,N → Box Height
p10="",0,F,C → Box ForeColor
p11="",6,F,C → Box BackColor
p12="",2,F,N → Image X position
p13="",2,F,N → Image Y position
p14="",37,F,N → Image Width
p15="",37,F,N → Image Height
p16="",nomebitmap.bmp,F,B → Bitmap name
The BitMap must be in the folder VTB\SystemBmp
Tipical Dimension 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… Start Custom Properties
See following definition:

Pn=”Descr”,Valdef,Vis,Type

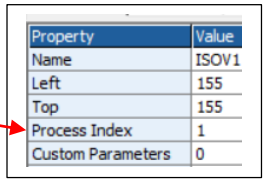
Descr Property description. Only if it is visible. This property is displayed in the Browser

Valdef Default value

Vis **V** Visible in the Browser
F Not visible in the Browser

Type **N** Numerical Value
V System variable (VTB global variable type bit long etc)
T Text (string type)

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



In the VTB code, is possible substitute the code, from the properties value.
You must insert the code internal the ***TAG ? property number?.***
Ex: **?p17?**
The TAG ?p17? is substitute to P17 value

1) Variable set from property value
   VarVtb=?pn? where n is the property number
   Ex:
   P17=”Prop 17”,102,V,N
   VarVtb=?p17? VarVTB = 102

2) Use the property value for variable declaration
   Generally is used the object name “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
```

All variables declared in the object, depende from object name, and they will be unique
Ex::
p1="Name",MyObj,V,T

All variables above wil become:
MyObj.*enable as char*
MyObj.*kp as long*
MyObj.*ki as long*
MyObj.*kv as long*

The properties can not be changed in ***run time***, If necessary, you must use a intermediate variable

END PROPERTY

End property section.
#END PROPRIETA#

GRAPHIC HEADER

Definines the image of the object showed in VTB the I.D.E.
This is only graphic rappresentation. Is recomended use the ***standard graphic***
#OGGETTI#

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

This rappresentation, identifies an Object, with a Box and a Bitmap inside

END GRAPHIC HEADER

End Graphic
#END OGGETTI#

VTB CODE HEADER

Beginning Vtb Code
#BEGIN CODE#

Constant Defining

Beginning section constant (DEFINE) used in the CODE
Generally, the DEFINE, are object dependent
BEGIN DEFINE

Ex: **?P1?_DEF_NAXISI as 3**
?P1?_LEN_PROCESS as 256

?P1? is replaced with object name

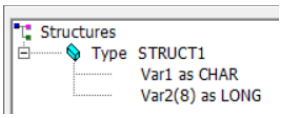
End Constant Defining

End define section
END DEFINE

Structures Defining

Beginning section Structures used in the VTB Code

BEGIN STRUCT



Ex:

```
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
```

This is a defining of structure

This is a defining of structure
?P1? DEF NASSI is a DEFINE

End Structures Defining

End structures section

END STRUCT

Global Variables Defining

Beginning section Gloabal Variables used in the VTB Code

BEGIN VAR

Ex:

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

End Global Variables

End Global variables section

END VAR

BIT Variables Defining

Beginning section BIT Variables used in the VTB Code

BEGIN VAR BIT

Ex:

?p1?.bit0 as ?P1?.var.0 → bit **0** of **?P1?.var** variable
?p1?.bit1 as ?P1?.var.1 → bit **1** of **?P1?.var** variable
?p1?.bit2 as ?P1?.var.2 → bit **2** of **?P1?.var** variable

End Bit Variables

End BIT variables section

END VAR BIT

Object Variables Defining

This section, normally is not used

BEGIN VAR OBJ

End Object Variables

End Section

END VAR OBJ

CanOpen SDO Variables Defining

Beginning section SDO Variables used in the VTB Code (See VTB manual)
BEGIN VAR OBJ (this section is not present in VTB IDE)

Ex: Following, has been inserted a P17 property ***"CanOpen Node Number"***

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

?p1?.acc as LONG at 0x608300 id ?p17? → **Acceleartion setting**
?p1?.dec as LONG at 0x608400 id ?p17? → **Deceleration setting**

Use:

```
?p1?.acc=100   'Acceleartion setting
?p1?.dec=50    'Deceleration setting
```

End SDO Variables

End SDO section
END VAR OBJ

INIT MAIN

Beginning VTB code insert in ***"init del TASK MAIN" (system init)***
BEGIN $INIT

Ex:

```
?P1?.var1=0
?P1?.var1=100
```

END INIT MAIN

End Init Main section
END $INIT

HEADER TASK

Header TASK Main and PLC
BEGIN $TASK
EVENTI_MASTER

TASK PLC

VTB code inserted in ***"Task Plc"***
All code inserted in this section, will writen in the TASK PLC
#BEGIN PLC#

Ex:

```
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
```

END TASK PLC End Task Plc section

All code Written below END TASK PLC, Is in Master Cycle of Main

#END PLC#

WARNING

TASK MAIN CODE MASTER CYCLE, DOES NOT HAVE A PARTICULAR HEADER, BUT USES #END PLC# TAG

Ex:

```
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
```

End Header Task End TASK

After this section, can be inserted, all global VTB Functions

END_EVENTI_MASTER

END $TASK

Global Functions List All Global Functins, must be inserted after END TASK Tag

Ex:

```
'-------------------------------------------
'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
```

END CODE HEADER OBJECT DECLARATION END Below you can declare a new object

From Tag OBJECT HEADER .- Uses a different OBJECT CODE

#END CODE#

4. Base File for Object create

Uses the fiollowing BASE, for declare a new object (or download) - 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. Object example

Following a VTB Object example

```
#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#
```

Index