

Description des fonctions de la librairie XVB32.DLL

©1999 DELEAU Jean-Marie e-mail : djm@euronet .be

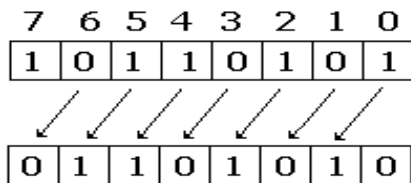
Fonctions de manipulations de bits

Dans les fonctions décrites ci-dessous, la numérotation des bits se fait de droite à gauche.
Le bit le plus à droite a pour index Zero.

Pour un octet : b7 b6 b5 b4 b3 b2 b1 b0

xShiftLeftB - xShiftLeftI - sShiftLeftL

Cette fonction deplace les octets du nombre de n octets vers la gauche .



Syntaxe : R = xShiftLeftB (nombre , déplacement)

Le déplacement est le nombre qui spécifie le nombre de décalage vers la gauche.

xShiftLeftB s'applique au type Byte

xShiftLeftI s'applique au type Integer

xShiftLeftL s'applique au type Long

L'octet le plus à droite est remplacé par un zéro.

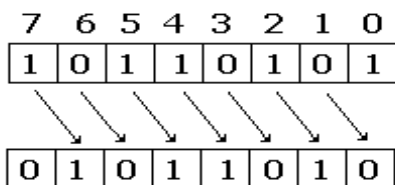
Exemple :

Nombre = &h95

R = xShiftLeftB (Nombre , 1) -> R sera égal à &h6A

xShiftRightB - xShiftRightI - sShiftRightL

Cette fonction deplace les octets du nombre de n octets vers la droite .



Syntaxe : R = xShiftRightB (nombre , déplacement)

Le déplacement est le nombre qui spécifie le nombre de décalage vers la droite.

xShiftRightB s'applique au type Byte

xShiftRightI s'applique au type Integer

xShiftRightL s'applique au type Long

L'octet le plus à gauche est remplacé par un zéro.

Exemple :

Nombre = &h95

R = xShiftRightB (Nombre , 1) -> R sera égal à &h5A

xTestBitB - xTestBitI - xTestBitL

Cette fonction teste un bit et retourne True (Vrai) si le bit est à 1 et False (Faux) si le bit est à 0.

Syntaxe :

R = xTestBitB (Nombre , position)

Position est l'index du bit à tester. La numérotation de l'index commence à 0

Exemple :

Nombre = 5

R = xTestBitB (5 , 0) → R = True car le bit0 est égal à 1

xTestBitB s'applique au type Byte

xTestBitI s'applique au type Integer

xTestBitL s'applique au type Long

xSetBitB - xSetBitI - xSetBitL

Cette fonction positionne à 1 un bit du nombre et retourne le nombre modifié.

Syntaxe :

R = xSetBitB (Nombre , Position)

Position est l'index du bit à positionner

Exemple :

R = xSetBitB (4 , 0) → R est égal à 5

xSetBitB s'applique au type Byte

xSetBitI s'applique au type Integer

xSetBitL s'applique au type Long

xResetBitB - xResetBitI - xResetBitL

Cette fonction positionne à 0 un bit du nombre et retourne le nombre modifié.

Syntaxe :

R = xResetBitB (Nombre , Position)

Position est l'index du bit à positionner.

Exemple :

R = xResetBitB (5 , 0) → R est égal à 4

xResetBitB s'applique au type Byte

xResetBitI s'applique au type Integer

xResetBitL s'applique au type Long

xLowByte - xHighByte

Ces fonctions extraient d'un Nombre de type Integer , l'octet Low et l'octet High

Syntaxe :

bLow = xLowByte (Nombre)

Nombre est un entier (Integer)

bLow est un octet

Exemple :

Nombre = &h1234

bLow = xLowByte (Nombre) → bLow est égal à &h34

bHigh = xHighByte (Nombre) → bHigh est égal à &h12

xLowInteger - xHighInteger

Ces fonctions extraient d'un Nombre de type LONG , l'integer Low et l'integer High

Syntaxe :

iLow = xLowInteger (Nombre)

Nombre est un LONG

iLow est un integer

Exemple :

Nombre = &h12347890

iLow = xLowInteger (Nombre) → iLow est égal à &h7890

iHigh = xHighInteger (Nombre) → iHigh est égal à &h1234

xHiLoByte - xHiLoInteger

Ces fonctions reconstituent un entier à partir de deux octets Low et High (fonction xHiLoByte)

Ou bien reconstitue un LONG à partir de deux integer.(xHiLoInteger)

Syntaxe :

R = xHiLoByte (Hi , Lo)

Hi est un byte et représente l'octet High

Lo est un byte et représente l'octet Low

R est un Integer

RL = xHiLoInteger (Hi , Lo)

Hi est integer et représente l'integer High

Lo est un integer et représente l'integer Low

R est un LONG

xRed - xGreen - xBlue

Ces fonctions extraient les composantes Rouge, Verte et Bleue d'une couleur de type RGB.

xRed extrait la composante Rouge
xGreen extrait la composante Verte
xBlue extrait la composante Bleue

Syntaxe :

C = xRed (Couleur)

Couleur est du type Long

C est du type integer

Exemple :

Col& = RGB (128 , 255 , 255)

R = xRed (Col&) → R est égal à 128

FONCTIONS D'ACCES MEMOIRE

Les fonctions décrites ci-dessous permettent de lire ou d'écrire dans la mémoire. Attention, une mauvaise utilisation de ces fonctions peut planter la machine. Certaines adresses ne sont pas accessibles car utilisées par windows. Ces fonctions ont été testées sous Win95/98, elles ne fonctionnent pas sous winNT. Ces fonctions sont données pour pouvoir les expérimenter , par exemple avec vos propres cartes d'extension. Ils n'est pas conseillé de les utiliser dans un programme que vous distribuez.

xPeekB - xPeekI - xPeekL

Ces fonctions permettent de lire respectivement un octet , un Integer et un entier de type Long à partir d'une Adresse mémoire.

Syntaxe :

B = xPeekB (Adresse)
Adresse est de type Long

xPokeB - xPokeI - xPokeL

Ces routines permettent d'écrire respectivement un octet, un integer ou un entier de type Long à une adresse mémoire.

Syntaxe :

xPokeB adresse , octet

Remarque : Pour obtenir l'adresse d'une variable, vous pouvez utiliser la fonction VarPTR de Visual Basic. Cette fonction n'est pas documentée dans VB.

Exemple :

Dim I As Integer
Adresse& = VarPtr (I)

xOut - xInp

Ces fonctions permettent d'écrire un octet et de lire un octet sur un port

Syntaxe :

xOut port , octet

octet = xInp (port)

