Access : créer des codes-barres avec Visual Basic

par Dominique KIRCHHOFER (http://domi2.developpez.com/)

Date de publication : 24 octobre 2010

Dernière mise à jour :

Créer des codes-barres avec Visual Basic. Niveau requis : intermédaire.

I - Introduction

Les questions touchant à la création de codes-barres sont relativement fréquentes sur le forum Access. Bien que plusieurs réponses aient été données, nombre de discussions demeurent non résolues. Dans la plupart des cas, ce n'est pas faute d'une bonne solution, mais plutôt l'absence de détails et de précisions qui ont fait que beaucoup de membres ont abandonné cette idée.

Cet article a pour but de vous expliquer comment créer des codes-barres, sans utiliser un contrôle ActiveX spécifique ni de police de caractères spéciale, mais uniquement avec Visual Basic.

Pour atteindre cet objectif, de bonnes connaissances en Visual Basic sont nécessaires, ce qui réserve plutôt cet article à des utilisateurs ayant déjà quelque expérience de la programmation avec ce language.

II - Généralités

Le premier système de codes-barres a été breveté en 1952. Il concernait un code à ligne verticale, mais aussi de forme concentrique (en forme de cible) ainsi que le système de lecture des données.

Leur première utilisation a été l'étiquetage des wagons de train, sans que le succès commercial ne soit toutefois au rendez-vous.

Il faudra attendre le début des années 1970 et l'invention du code UCP, *Universal Product Code* ou CUP en français, utilisé dans la grande distribution, pour que leur usage se généralise.

Aujourd'hui, ils sont utilisés dans de nombreux domaines d'activité, par exemple :

- *code 11* : télécommunications ;
- *code 39* : industrie automobile, pharmacode France, armée américaine ;
- *code 128* : transports, santé, pièces détachées pour le secteur automobile.

Cette liste est bien sûr loin d'être exhaustive.

II-A - Les codes-barres

À l'origine, les codes-barres sont la représentation, sous forme d'un ensemble de barres et d'espaces, d'une donnée numérique ou alphanumérique, l'épaisseur des barres et espaces variant selon la symbiologie utilisée et les caractères de la donnée à coder. Ces codes, linéaires, sont lus horizontalement.

Puis sont apparus des codes linéaires empilés, ainsi nommés car ils sont constitués de plusieurs codes-barres linéaires empilés les uns sur les autres. Ils sont lus verticalement.

Plus récemment, ils ont encore évolué avec l'apparition de codes-barres à deux dimensions (2D), lus horizontalement **et** verticalement. Ils permettent de coder un grand nombre de données sur une petite surface, jusqu'à plusieurs milliers de caractères pour certains d'entre eux.

Code-barres linéaire	Code-barres linéaire empilé	Code-barres à deux dimensions
		
Code-barres 128	Code-barres PDF 417	Code-barres Aztec

II-B - Lecture des codes-barres

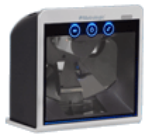
Pour lire les données ainsi codées, on utilise un lecteur de code-barres, qui n'est autre qu'un lecteur optique émettant de la lumière :

- le lecteur émet de la lumière qui est est absorbée par les barres sombres et réfléchie par les espaces clairs ;
- à l'intérieur du lecteur, une cellule photosensible reçoit la lumière réfléchie et la convertit en signal électrique ;
- le signal est faible pour les espaces clairs et fort pour les barres sombres ;
- la durée du signal détermine la largeur des barres et espaces ;
- un décodeur convertit le signal en caractères ;
- les caractères décodés sont transmis à l'ordinateur.

Différents types de lecteurs de codes-barres :

Stylo optique

Douchette



Scanner

III - Le code-barres 128

Comme nous l'avons vu dans le précédent chapitre, il existe de nombreux types de codes-barres, destinés pour la plupart à des domaines bien spécifiques. Dans le cadre de cet article, nous n'en étudierons qu'un seul, choisi non pas en fonction d'un domaine d'utilisation précis, mais parce qu'il permet de coder un maximum de caractères de manière simple et fiable.

Le choix du *code 128* s'est donc imposé presque naturellement. Il permet en effet de coder les 128 caractères ASCII de base, il est omnidirectionnel, il est très dense, notamment pour les valeurs numériques, et possède un caractère de contrôle.

Il ne faut pas confondre le code 128 avec le code EAN (UCC) 128. Ce dernier, basé sur la symbologie du code 128 n'est pas simplement un code-barres, c'est un standard qui définit la manière dont les données sont formatées.
Leur signification est précisée selon une liste d'Identificateurs d'Application, usuellement abrégé AI. Ces derniers, placés au début du code, indiquent ce que contiennent les données elle-mêmes (numéro de colis, date de production, d'emballage, etc.).

III-A - Les différents caractères du code

Il est constitué de 107 motifs différents, représentant 103 caractères de données, 3 caractères de démarrage et un caractère d'arrêt. Pour permettre de coder les 128 caractères de la table ASCII, il existe 3 ensembles de codes (A, B et C) qui peuvent être mélangés dans un même code-barres par l'utilisation de caractères de permutation (caractères 99, 100 et 101).

Contenu des tables :

- 128A : les chiffres, les lettres majuscules, des caractères de ponctuations, des codes de contrôle et des codes spéciaux ;
- 128B : les chiffres, les lettres majuscules et minuscules, des caractères de ponctuations et des codes spéciaux ;
- 128C : des paires de chiffres, permettant de doubler la densité des caractères numériques, et des codes spéciaux.

La table représentant les 3 ensembles de caractères du code est disponible en annexe.

Dans cet article, nous nous limiterons à l'usage des ensembles de caractères B et C.

III-B - La stucture du code

Le *code 128* est composé de 6 sections :

- une marge blanche à gauche ou "quiet zone" ;
- un caractère de démarrage ;
- les données elles-mêmes ;
- un caractère de contrôle ;
- un caractère d'arrêt ;
- une marge blanche à droite ou "quiet zone".

- Les marges ou "quiet zone" à gauche et à droite du code doivent avoir au moins 10 modules d'épaisseur ;
- le caractère de démarrage détermine la table utilisée au départ ;
- le caractère, ou clé, de contrôle est obligatoire. Le détail du calcul est exposé ci-après ;
- le caractère d'arrêt est identique, quelque soit les ensembles de caractères utilisés.

III-C - Les modules

Comme nous l'avons précédemment, le *code 128* est constitué de 107 motifs différents, composés chacun de 11 modules, groupés en 3 barres et 3 espaces.

Le caractère d'arrêt est complété par deux modules de barres à son extrémité droite permettant de fermer le code-barres. Il est donc composé de 13 modules au total.

Pour encoder les données, on peut considérer que le chiffre "1" représente des barres et le chiffre "0" des espaces. Ainsi la lettre "A" sera symbolisée par "10100011000", soit 3 barres et 3 espaces de différentes épaisseurs.

III-D - Optimisation du code

Un des intérêts du *code 128* réside dans la possibilité d'optimiser le codage de données numériques en utilisant un seul caractère pour des paires de chiffres, en recourant à l'ensemble de caractères C.

Ainsi, si "44" nécessite 2 fois le caractère "4" dans l'ensemble B, il peut être codé uniquement avec le caractère "L" en recourant à l'ensemble C.

Il faut tenir compte ici des caractères spéciaux permettant de passer d'un ensemble de caractères à l'autre. L'optimisation du code est donc possible pour autant que la donnée à coder :

- commence par 4 chiffres au moins ;
- se termine par 4 chiffres au moins ;
- comprenne 6 chiffres au moins au milieu de la donnée.

En effet, dans le premier cas, on commence en effet à utiliser l'ensemble C, puis on utilise **1** caractère de permutation pour passer à l'ensemble B pour coder la suite des données si nécessaire.

Dans le deuxième cas, on commence par l'ensemble B, puis **1** caractère est utilisé pour passer vers l'ensemble C.

Enfin, dans le troisième cas de figure, **2** caractères sont nécessaires pour passer de l'ensemble B vers l'ensemble C, puis à nouveau vers l'ensemble B.

A titre d'exemple, essayons de créer le code pour la donnée **ABC2011**, tout d'abord sans optimisation de l'encodage, c'est-à-dire en ayant recours uniquement à l'ensemble de caractères B.

Le résultat obtenu est le suivant :

Caractère ASCII	Ñ	A	B	C	2	0	1	1
Caractère	Start B	A	B	C	2	0	1	1
Valeur	104	33	34	35	18	16	17	17

Reprenons la même donnée, en optimisant le code cette fois-ci, c'est-à-dire en ayant recours aux ensembles de caractères B et C.

Le résultat est naturellement différent, les 4 caractères numériques étant codés avec 2 caractères uniquement :

Caractère ASCII	Ñ	A	B	C	Ì	4	+
Caractère	Start B	A	B	C	Table C	20	11
Valeur	104	33	34	35	99	20	11

On voit bien l'intérêt d'optimiser l'encodage, le recours à ce procédé permettant de produire des codes très denses s'agissant de données numériques.

⚠ *Dans les deux exemples ci-dessus, le caractère de contrôle et le caractère d'arrêt ne sont pas encore compris.*

III-E - Le caractère de contrôle

Le caractère de contrôle, obligatoire, se calcule selon un schéma bien précis et en utilisant un modulo 103.

En lisant le code de gauche à droite, on commence par prendre la valeur du code de démarrage (103, 104 ou 105) pour les ensembles de caractères A, B ou C respectivement.

Puis, pour chaque caractère suivant, on multiplie la valeur du caractère par la valeur de sa position dans la chaîne, le caractère suivant immédiatement le caractère de démarrage ayant le rang 1.

Caractère ASCII	Ñ	A	B	C	Ì	4	+
Caractère	Start B	A	B	C	Table C	20	11
Valeur	104	33	34	35	99	20	11
Position	-	1	2	3	4	5	6
Multiplication	104	33*1	34*2	35*3	99*4	20*5	11*6
Produit	104	33	68	105	396	100	66

Le résultat de l'addition des produits, 872 dans cet exemple, est divisé par 103, le **reste** de la division étant la valeur du caractère de contrôle, soit 48.

Ensuite, cette valeur doit être convertie en caractère ASCII en appliquant la méthode suivante :

- si elle est comprise entre 0 et 94, on ajoute 32 ;
- si elle est supérieure à 94, on ajoute 105.

Ici, la valeur étant inférieure à 95, nous ajoutons 32. Le résultat, soit 80, donne le caractère ASCII "P".

Il ne reste alors plus qu'à ajouter ce caractère de contrôle et le caractère d'arrêt pour compléter le code :

Caractère ASCII	Ñ	A	B	C	Ì	4	+	P	Ó
Caractère	Start B	A	B	C	Table C	20	11	P	Stop

*Nous connaissons maintenant l'intégralité des caractères ASCII que nous devons convertir en barres et espaces pour obtenir le code-barres de la donnée **ABC2011**.*

III-F - Les spécifications

Il s'est avéré difficile de trouver des spécifications pour le *code 128*. La plupart de celles trouvées sur Internet sont le fait de fournisseurs de polices de caractères, de lecteurs de codes-barres ou encore d'imprimantes.

Il existe également des spécifications plus officielles, par exemple la norme française et européenne NF EN 799, applicable entre autre à l'impression des N° RPPS et N° AM des ordonnances médicales.

Le but de cette article n'est cependant pas de produire des codes-barres pour un usage spécifique. Aussi, les indications fournies ci-après sont plutôt une synthèse des divers renseignements trouvés, qui devraient permettre d'imprimer des codes-barres lisibles par la grande majorité des lecteurs disponibles sur le marché.

Il n'est non plus pas nécessaire de disposer d'une imprimante haut de gamme pour imprimer vos codes-barres. Laser ou jet d'encre, n'importe quelle imprimante suffit aujourd'hui à cette tâche.

III-F-1 - Légende du code-barres

Le code-barres peut être légendé ou non.

III-F-2 - Largeur des modules et du code-barre

La dimension X d'un module devrait être de 7.5 millièmes de pouce au minimum.

Sachant qu'un pouce est égal à 1'440 twips ou 2.54 cm, un millième de pouce est égal à 1.44 twips. La dimension X minimum d'un module est donc d'environ 10.8 twips.

Bien qu'il soit tout à fait possible de lire des codes-barres ayant des modules de 10 twips, dans l'application, nous utiliserons une valeur par défaut de 12 twips, ce qui devrait garantir une lecture par tout type de matériel.

Ceci est en principe largement suffisant. Pour une donnée de 25 caractères par exemple, soit 28 caractères en tenant compte des caractères de démarrage, de contrôle et d'arrêt, la largeur du code-barres n'excède pas 6.56 cm.

La largeur des codes-barres n'est en théorie pas limitée. Vous devez cependant être attentif aux caractéristique techniques de votre lecteur de codes-barres.
Certaines douchettes sont par exemple dans l'incapacté de lire des codes-barres de plus de 7.5 cm de largeur.

III-F-3 - Hauteur des modules

La symbologie du *code 128* ne spécifie également pas de hauteur maximum pour les modules. Il serait toutefois faux de croire que plus la hauteur d'un code-barres est importante, plus celui-ci est lisible.

La hauteur minimum du code-barres devrait être de 15 % de la longueur de ce dernier ou 0.25 pouces, soit environ 0.6cm, la valeur la plus élevée étant retenue.

Pour la hauteur maximum, on peut se limiter à 0.5 pouces, soit environ 1.3 cm. Dans l'application exemple, elle est fixée au plus à 1 cm.

III-F-4 - Specifications de vos propres codes-barres

Comme cela a été précisé, les spécifications ci-dessus n'ont rien d'officiel. Il vous faudra donc faire vos propres tests pour obtenir les meilleurs résultats possibles.

Les lecteurs de codes-barres sont généralement fournis avec une documentation contenant des codes témoins, permettant soit de calibrer le lecteur, soit de faire des simples tests de lecture.
Si un code 128 est fourni, l'étude de celui-ci vous fournira de précieuses indications (hauteur, largeur des modules) pour la création de vos propres codes-barres.

IV - L'application de création de code-barres

Après cette intoduction théorique, nous pouvons passer à la création de l'application !

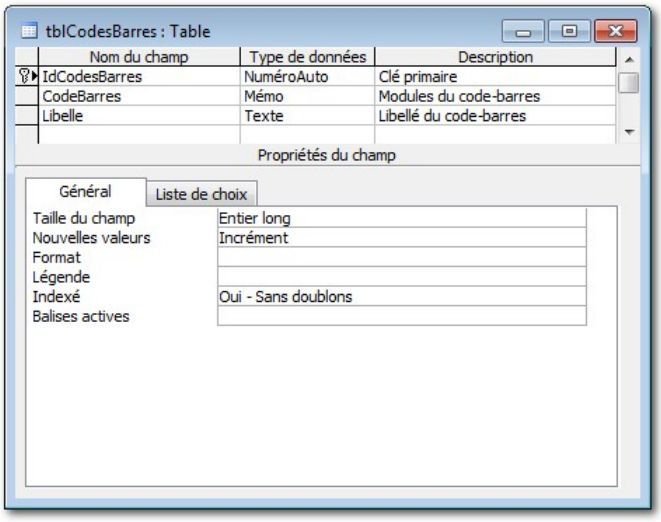
Celle-ci est assez modulaire, de sorte que vous devriez pouvoir l'intégrer facilement dans une application existante.

Elle est composée :

- d'une table ;
- d'un formulaire et de son module (4 procédures) ;
- d'un état et de son module (2 procédures) ;
- d'un module de code contenant 3 fonctions nécessaires à l'encodage et au tracage des codes-barres.

IV-A - Table

On commence par créer une table **tblCodesBarres** avec 3 champs :

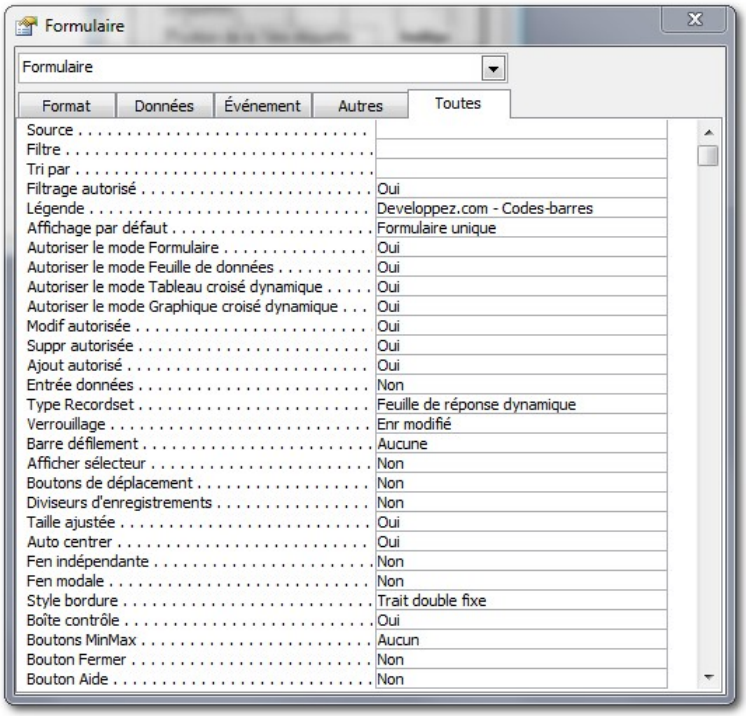


- **IdCodesBarres**, de type **NumérotAuto**, qui est la clé primaire de la table ;
- **CodesBarres**, de type **Mémo**, pour recevoir les modules du code-barres ;
- **Libelle**, de type **Texte** (taille 50), pour le libellé du code.

*Si vous vous êtes certains que le nombre de modules constituant vos codes-barres sera toujours égal ou inférieur à 255, vous pouvez bien sûr utiliser un champ de type **Texte** pour le champ **CodesBarres**.*

IV-B - Formulaire

On peut ensuite passer à la création du formulaire **frmCodeBarres** et modifier les propriétés qui nous intéressent.



On notera principalement qu'il est indépendant, qu'il n'a pas de barres de défilement, ni de sélecteur et de diviseurs d'enregistrements. Les boutons ne sont également pas affichés.

On y place 4 zones de texte, indépendantes elles aussi :

- **txtChaineCaractere** va nous permettre de saisir la donnée du code-barres ;
- **txtEtiquetteDebut** indique la position de la 1ère étiquette à imprimer. La propriété **Valeur par défaut** de cette zone de texte est fixée à 1 ;
- **txtNbEtiquettes** définit le nombre d'étiquettes désirés. La propriété **Valeur par défaut** de cette zone de texte est fixée à 16 ;
- **txtLargeurModules** fixe la dimension X ou largeur des modules, exprimée en twips. La propriété **Valeur par défaut** de cette zone de texte est fixée à 12.

Puis on ajoute 2 images et un bouton avec sa propriété **Tranparent** à **Oui**, qu'on superpose les uns sur les autres dans cet ordre, en commençant par l'arrière-plan :

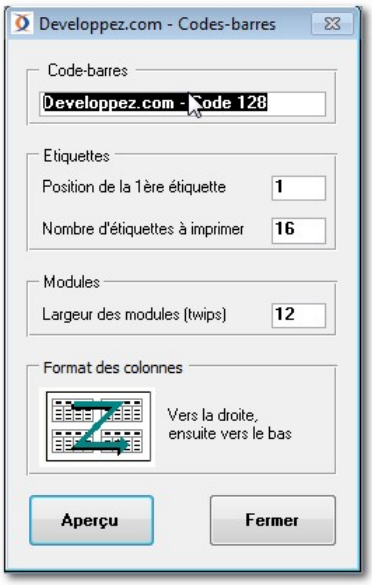
- **imgTracageVertical** ;
- **imgTracageHorizontal** ;
- **cmdTracage**.

Ces 3 trois éléments vont nous permettre, sur clic du bouton **cmdTracage**, de sélectionner le mode de **Traçage** des étiquettes.

Enfin, on rajoute 2 boutons :

- **cmdAperculmpression** ouvre l'état **rptEtiquettes** en mode **Aperçu avant impression** ;
- **cmdFermer** ferme le formulaire.

Ci-dessous, l'aspect définitif du formulaire :



IV-C - Etat

Enfin, on crée un état **rptEtiquettes**. Il ne présente pas de particularité et on peut sans autre utiliser l'**Assistant Étiquettes** en sélectionnant la table **tblCodesBarres** comme **Source**.

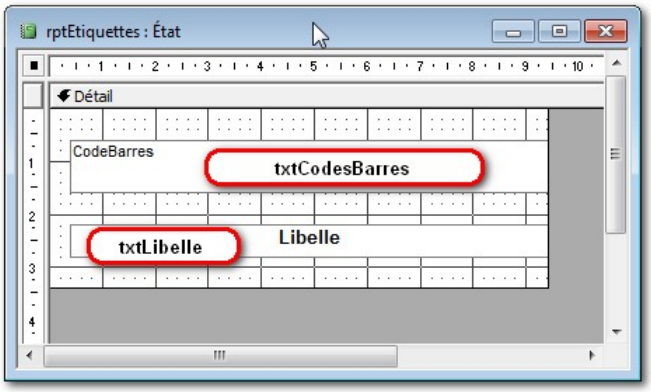
Pour l'application, le choix s'est porté sur un modèle de 2 étiquettes de front. Après ajustement manuel des dimensions, celles-ci sont d'environ 94 x 34 mm, soit 16 étiquettes au total.

On y insère ensuite les champs **CodeBarres** et **Libelle**.

*Si le champ **CodeBarres** et de type **Mémo**, il est possible que celui ne soit pas proposé par l'assistant. Il est donc nécessaire de l'ajouter manuellement par la suite en modifiant votre état.*

Ensuite que quoi nous renommons les deux zones de texte, respectivement **txtCodeBarres** et **txtLibelle**.

Aperçu de l'état en mode **Création** :



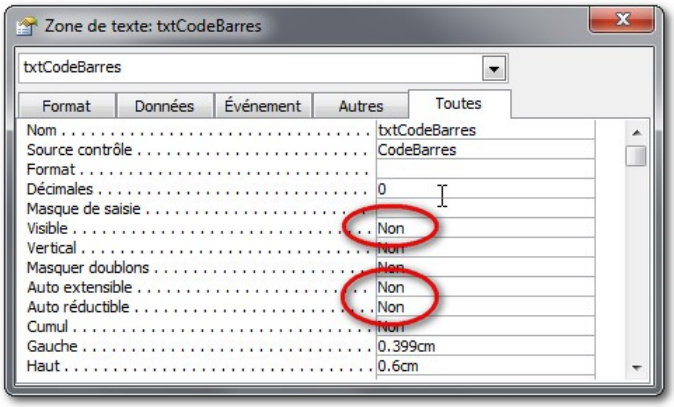
Il est utile de commenter 2 propriétés de la zone de texte **txtCodeBarres**.

Tout d'abord, celle-ci n'est pas visible. En effet, on ne va pas tracer le code-barres **dans** la zone de texte, mais bien **par-dessus** celle-ci, en utilisant le coin supérieur gauche comme point d'origine. La propriété **Visible** est fixée à **Non**.

D'autre part, il est possible de tracer des codes-barres de hauteur proportionnelle à la longueur du code ou de taille fixe.

Cette dernière option implique que les propriétés **Auto extensible** et **Auto réductible** soient fixées à **Non** également.

Propriétés de la zone de texte :



IV-D - Modules de code

IV-D-1 - Le module du formulaire

Code

```
Private Sub Form_Open(cancel As Integer)

    On Error GoTo GestionErreurs

    'Vérification du mode de traçage des colonnes renvoyé par défaut par l'imprimante
    'Traçage horizontal = acPRHorizontalColumnLayout = 1953
    'Traçage vertical = acPRVerticalColumnLayout = 1954
    lngTracageColonne = Printer.ItemLayout
    'Affichage de l'image correspondant au mode de traçage
    If lngTracageColonne = 1953 Then
        Me.imgTracageHorizontal.Visible = True
        Me.imgTracageVertical.Visible = False
        Me.lblTracageColonnes.Caption = "Vers la droite, ensuite vers le bas"
    Else
        Me.imgTracageVertical.Visible = True
        Me.imgTracageHorizontal.Visible = False
        Me.lblTracageColonnes.Caption = "Vers le bas, ensuite vers la droite"
    End If

    'Sortie de la procédure
    Exit Sub

GestionErreurs:

    'Affichage d'un message
    MsgBox "Une erreur inattendue s'est produite !" & vbNewLine & "Erreur no : " _
    & Err.Number & vbNewLine & Err.Description, vbCritical, "Erreur !"

End Sub
```

Code

```
Private Sub cmdTracage_Click()

    On Error GoTo GestionErreurs

    'Modification du mode de traçage des étiquettes avec affichage de l'image correspondante
    If lngTracageColonne = 1953 Then
        Me.imgTracageVertical.Visible = True
        Me.imgTracageHorizontal.Visible = False
        Me.lblTracageColonnes.Caption = "Vers le bas, ensuite vers la droite"
        lngTracageColonne = 1954
    Else
        Me.imgTracageHorizontal.Visible = True
        Me.imgTracageVertical.Visible = False
        Me.lblTracageColonnes.Caption = "Vers la droite, ensuite vers le bas"
        lngTracageColonne = 1953
    End If

    'Sortie de la procédure
    Exit Sub

GestionErreurs:

    'Affichage d'un message
    MsgBox "Une erreur inattendue s'est produite !" & vbNewLine & "Erreur no : " _
    & Err.Number & vbNewLine & Err.Description, vbCritical, "Erreur !"

End Sub
```

Code

```
Private Sub cmdApercuImpression_Click()

    'Déclaration des objets et des variables
    Dim db As DAO.Database
    Dim rst As DAO.Recordset
    Set db = CurrentDb
    Set rst = db.OpenRecordset("select * from tblCodesBarres")

    Dim strChaine As String
    Dim strCaractere As String
    Dim strBarres
    Dim strCodeBarres As String
    Dim i As Long

    On Error GoTo GestionErreurs

    'La largeur des modules est assignée à la variable lngTailleModule
    lngTailleModule = Me.txtLargeurModules

    'Suppression des enregistrements de la table tblCodesBarres
    CurrentDb.Execute "DELETE tblCodesBarres.CodeBarres FROM tblCodesBarres;", dbFailOnError

    'Codage des données
    strChaine = Code128(Me.txtChaineCaractere)

    For i = 1 To Len(strChaine)
        strCaractere = Mid(strChaine, i, 1)
        strBarres = MotifCodeBarres128(strCaractere)
        strCodeBarres = strCodeBarres & strBarres
    Next i

    'Calcul de la hauteur des modules par rapport à la largeur du code-barres
    lngLargeurCodeBarres = Len(strCodeBarres) * Me.txtLargeurModules
    Select Case lngLargeurCodeBarres
        '15 % de la longueur excède 567 twips (1 cm)
        Case Is > 3780
            'Hauteur maximale des modules égale 567 twips
            lngHauteurModule = 567
        '15 % de la longeur est inférieur à 340 twips (0.6 cm)
```

Code

```
        Case Is < 2268
            'Hauteur minimale des modules, 340 twips
            lngHauteurModule = 340
        Case Else
            'La hauteur des modules est égale à 15 % de la longueur du code-barres
            lngHauteurModule = Int((lngLargeurCodeBarres / 100 * 15) + 1 * 0.5)
    End Select

    MsgBox lngHauteurModule

    'Ajout des "Quiet zone" en début et en fin du code-barres
    strCodeBarres = "00000000000" & strCodeBarres & "00000000000"
    'Largeur du code-barres, "Quiet zone" comprises
    lngLargeurCodeBarres = Len(strCodeBarres) * Me.txtLargeurModules

    i = 0

    'Ajout d'enregistrements vides en fonction de l'étiquette de début sélectionnée
    For i = 1 To Me.txtEtiquetteDebut - 1
        rst.AddNew
        rst("CodeBarres") = Null
        rst("Libelle") = Null
        rst.Update
    Next i

    'Ajout du nombre d'enregistrements correspondants au nombre d'étiquettes désirées
    i = 0
    For i = 1 To Me.txtNbEtiquettes
    rst.AddNew
        rst("CodeBarres") = strCodeBarres
        rst("Libelle") = Me.txtChaineCaractere
        rst.Update
    Next i

    'Ouverture de l'état rptEtiquettes en le maximisant et en l'ajustant à la fenêtre Access
    DoCmd.OpenReport "rptEtiquettes", acViewPreview
    DoCmd.Maximize
    DoCmd.RunCommand acCmdFitToWindow

SortieApercuImpression:

    'Libération des objets
    rst.Close
    db.Close
    Set rst = Nothing
    Set db = Nothing

    'Sortie de la procédure
    Exit Sub

GestionErreurs:

    Select Case Err.Number
       'L'ouverture de l'état à été annulée
        Case 2501

            'Sortie de la procédure
            Resume SortieApercuImpression
        'La chaîne de caractère est nulle
        Case 513
            'Affichage d'un message
            MsgBox "La chaîne de caractères n'est pas valide (Null) !" & vbNewLine & vbNewLine & _
            "Veuillez vérifiez votre saisie.", vbCritical, "Erreur !"

            'Sortie de la procédure
            Resume SortieApercuImpression
        'Le code contient des caractères n'appartenant pas à la table ASCII de base
        Case 514
            'Affichage d'un message
            MsgBox "La chaîne de caractères n'est pas valide !" & vbNewLine & vbNewLine & _
            "Elle contient des caractères n'appartenant pas à la table" & vbNewLine & _
```

Code

```
            "ASCII de base, par exemple des caractères accentués.", vbCritical, "Erreur !"

            'Sortie de la procédure
            Resume SortieApercuImpression
        Case Else
            'Affichage d'un message
            MsgBox "Une erreur inattendue s'est produite !" & vbNewLine & vbNewLine & _
            "Source : " & Err.Source & vbNewLine & _
            "Erreur no : " & Err.Number & vbNewLine & _
            "Description : " & Err.Description, vbCritical, "Erreur !"

            'Sortie de la procédure
            Resume SortieApercuImpression
    End Select

End Sub
```

Code

```
Private Sub cmdFermer_Click()

    On Error GoTo GestionErreurs

    'Fermeture du formulaire
    DoCmd.Close

    'Sortie de la procédure
    Exit Sub

GestionErreurs:

    'Affichage d'un message
    MsgBox "Une erreur inattendue s'est produite !" & vbNewLine & "Erreur no : " _
    & Err.Number & vbNewLine & Err.Description, vbCritical, "Erreur !"

End Sub
```

IV-D-2 - Le module de l'état

Code

```
Private Sub Report_Open(cancel As Integer)

    On Error GoTo GestionErreurs

    'Mode de traçage des étiquettes : 1953 = Traçage horizontal : 1954 = Traçage vertical
    Me.Printer.ItemLayout = lngTracageColonne

    If lngLargeurCodeBarres > Me.txtCodeBarres.Width Then

        'Affichage d'un message
        MsgBox "La largeur des codes-barres est supérieure à celle des étiquettes !", _
        vbCritical, "Erreur !"

        'Annulation de l'ouverture de l'état. L'erreur est gérée dans la procédure appelante
        cancel = True

    End If

    'Sortie de la procédure
    Exit Sub

GestionErreurs:

    'Affichage d'un message
    MsgBox "Une erreur inattendue s'est produite !" & vbNewLine & "Erreur no : " _
    & Err.Number & vbNewLine & Err.Description, vbCritical, "Erreur !"
```

Code
```
End Sub
```

Code
```
Private Sub Detail_Format(cancel As Integer, FormatCount As Integer)

    On Error GoTo GestionErreurs

    'Appel de la fonction de traçage des codes-barres
    TracerMotifCodeBarres128 Me.txtCodeBarres, Me

    'Sortie de la procédure
    Exit Sub

GestionErreurs:

    'Affichage d'un message
    MsgBox "Une erreur inattendue s'est produite !" & vbNewLine & "Erreur no : " _
    & Err.Number & vbNewLine & Err.Description, vbCritical, "Erreur !"

End Sub
```

IV-D-3 - Les fonctions de codage et de dessin du code-barres

Blablablablablabla

Code
```
Option Compare Database
Option Explicit

'Déclaration des variables publiques
'Mode de traçage des colonnes de l'état
Public lngTracageColonne As Long
'Taille d'un module de barre ou d'espace (twips)
Public lngTailleModule As Long
'Taille d'un module de barre ou d'espace (twips)
Public lngHauteurModule As Long
'Largeur du code-barres (twips)
Public lngLargeurCodeBarres As Long

'Déclaration des constantes utilisées dans le module
'Couleur des barres
Private Const CouleurBarre As Long = 0
'Couleur des espaces
Private Const CouleurEspace As Long = 16777215
```

Code
```
Public Function Code128(strChaine)

    'Déclaration des variables
    Dim i As Long
    Dim lngCaractereControle As Long
    Dim strMini As String
    Dim strDummy As String
    Dim blnTableB As Boolean

    On Error GoTo GestionErreurs

    'Provoque une erreur transmise à la procédure appelante si la chaîne est nulle
    If IsNull(strChaine) Then Err.Raise 513

    'Calculer la chaine de code en optimisant l'usage des tables B et C
    Code128 = Null
    blnTableB = True
```

Code

```
    i = 1 'i devient l'index sur la chaîne
    Do While i <= Len(strChaine)
        If blnTableB Then
            'Voir si intéressant de passer en table C
            'Oui pour 4 chiffres au début ou à la fin, sinon pour 6 chiffres
            strMini = IIf(i = 1 Or i + 3 = Len(strChaine), 4, 6)
            GoSub TestNum
            If strMini < 0 Then 'Choix table C
                If i = 1 Then 'Débuter sur table C
                    Code128 = Chr(210)
                Else 'Commuter sur table C
                    Code128 = Code128 & Chr(204)
                End If
                blnTableB = False
            Else
                If i = 1 Then Code128 = Chr(209) 'Débuter sur table B
            End If
        End If
        If Not blnTableB Then
            'On est sur la table C, essayer de traiter 2 chiffres
            strMini = 2
            GoSub TestNum
            If strMini < 0 Then 'OK pour 2 chiffres, les traiter
                strDummy = Val(Mid(strChaine, i, 2))
                strDummy = IIf(strDummy < 95, strDummy + 32, strDummy + 105)
                Code128 = Code128 & Chr(strDummy)
                i = i + 2
            Else 'On n'a pas 2 chiffres, repasser en table B
                Code128 = Code128 & Chr(205)
                blnTableB = True
            End If
        End If
        If blnTableB Then
            'Traiter 1 caractère en table B
            Code128 = Code128 & Mid$(strChaine, i, 1)
            i = i + 1
        End If
    Loop
    'Calcul de la clé de contrôle
    For i = 1 To Len(Code128)
        strDummy = Asc(Mid(Code128, i, 1))
        strDummy = IIf(strDummy < 127, strDummy - 32, strDummy - 105)
        If i = 1 Then lngCaractereControle = strDummy
        lngCaractereControle = (lngCaractereControle + (i - 1) * strDummy) Mod 103
    Next
    'Calcul du code ASCII de la clé
    lngCaractereControle = IIf(lngCaractereControle < 95, lngCaractereControle + 32,
 lngCaractereControle + 105)
    'Ajout de la clé et du STOP
    Code128 = Code128 & Chr(lngCaractereControle) & Chr(211)

    'Sortie de la fonction
    Exit Function

TestNum:
    'Si les strMini caractères à partir de i sont numériques, alors strMini=0
    strMini = strMini - 1
    If i + strMini <= Len(strChaine) Then
        Do While strMini >= 0
            If Asc(Mid$(strChaine, i + strMini, 1)) < 48 Or Asc(Mid$(strChaine, i +
 strMini, 1)) > 57 Then Exit Do
            strMini = strMini - 1
        Loop
    End If
    Return

GestionErreurs:

    'Transmet l'erreur à la procédure appelante
```

Code

```
    Err.Raise Err.Number, "Code128"

End Function
```

Code

```
Public Function MotifCodeBarres128(strChaine As String) As String

    On Error GoTo GestionErreurs

    Select Case Asc(strChaine)
        Case 32:
 MotifCodeBarres128 = "11011001100" ' Caractère = Espace / Table B = Espace / Table C = 00
        Case 33: MotifCodeBarres128 = "11001101100" ' Caractère = ! / Table B = ! / Table C = 01
        Case 34: MotifCodeBarres128 = "11001100110" ' Caractère = " / Table B = " / Table C = 02
        Case 35: MotifCodeBarres128 = "10010011000" ' Caractère = # / Table B = # / Table C = 03
        Case 36: MotifCodeBarres128 = "10010001100" ' Caractère = $ / Table B = $ / Table C = 04
        Case 37: MotifCodeBarres128 = "10001001100" ' Caractère = % / Table B = % / Table C = 05
        Case 38: MotifCodeBarres128 = "10011001000" ' Caractère = & / Table B = & / Table C = 06
        Case 39: MotifCodeBarres128 = "10011000100" ' Caractère = ' / Table B = ' / Table C = 07
        Case 40: MotifCodeBarres128 = "10001100100" ' Caractère = ( / Table B = ( / Table C = 08
        Case 41: MotifCodeBarres128 = "11001001000" ' Caractère = ) / Table B = ) / Table C = 09
        Case 42: MotifCodeBarres128 = "11001000100" ' Caractère = * / Table B = * / Table C = 10
        Case 43: MotifCodeBarres128 = "11000100100" ' Caractère = + / Table B = + / Table C = 11
        Case 44: MotifCodeBarres128 = "10110011100" ' Caractère = , / Table B = , / Table C = 12
        Case 45: MotifCodeBarres128 = "10011011100" ' Caractère = - / Table B = - / Table C = 13
        Case 46: MotifCodeBarres128 = "10011001110" ' Caractère = . / Table B = . / Table C = 14
        Case 47: MotifCodeBarres128 = "10111001100" ' Caractère = / / Table B = / / Table C = 15
        Case 48: MotifCodeBarres128 = "10011101100" ' Caractère = 0 / Table B = 0 / Table C = 16
        Case 49: MotifCodeBarres128 = "10011100110" ' Caractère = 1 / Table B = 1 / Table C = 17
        Case 50: MotifCodeBarres128 = "11001110010" ' Caractère = 2 / Table B = 2 / Table C = 18
        Case 51: MotifCodeBarres128 = "11001011100" ' Caractère = 3 / Table B = 3 / Table C = 19
        Case 52: MotifCodeBarres128 = "11001001110" ' Caractère = 4 / Table B = 4 / Table C = 20
        Case 53: MotifCodeBarres128 = "11011100100" ' Caractère = 5 / Table B = 5 / Table C = 21
        Case 54: MotifCodeBarres128 = "11001110100" ' Caractère = 6 / Table B = 6 / Table C = 22
        Case 55: MotifCodeBarres128 = "11101101110" ' Caractère = 7 / Table B = 7 / Table C = 23
        Case 56: MotifCodeBarres128 = "11101001100" ' Caractère = 8 / Table B = 8 / Table C = 24
        Case 57: MotifCodeBarres128 = "11100101100" ' Caractère = 9 / Table B = 9 / Table C = 25
        Case 58: MotifCodeBarres128 = "11100100110" ' Caractère = : / Table B = : / Table C = 26
        Case 59: MotifCodeBarres128 = "11101100100" ' Caractère = ; / Table B = ; / Table C = 27
        Case 60: MotifCodeBarres128 = "11100110100" ' Caractère = < / Table B = < / Table C = 28
        Case 61: MotifCodeBarres128 = "11100110010" ' Caractère = = / Table B = = / Table C = 29
        Case 62: MotifCodeBarres128 = "11011011000" ' Caractère = > / Table B = > / Table C = 30
        Case 63: MotifCodeBarres128 = "11011000110" ' Caractère = ? / Table B = ? / Table C = 31
        Case 64: MotifCodeBarres128 = "11000110110" ' Caractère = @ / Table B = @ / Table C = 32
        Case 65: MotifCodeBarres128 = "10100011000" ' Caractère = A / Table B = A / Table C = 33
        Case 66: MotifCodeBarres128 = "10001011000" ' Caractère = B / Table B = B / Table C = 34
        Case 67: MotifCodeBarres128 = "10001000110" ' Caractère = C / Table B = C / Table C = 35
        Case 68: MotifCodeBarres128 = "10110001000" ' Caractère = D / Table B = D / Table C = 36
        Case 69: MotifCodeBarres128 = "10001101000" ' Caractère = E / Table B = E / Table C = 37
        Case 70: MotifCodeBarres128 = "10001100010" ' Caractère = F / Table B = F / Table C = 38
        Case 71: MotifCodeBarres128 = "11010001000" ' Caractère = G / Table B = G / Table C = 39
        Case 72: MotifCodeBarres128 = "11000101000" ' Caractère = H / Table B = H / Table C = 40
        Case 73: MotifCodeBarres128 = "11000100010" ' Caractère = I / Table B = I / Table C = 41
        Case 74: MotifCodeBarres128 = "10110111000" ' Caractère = J / Table B = J / Table C = 42
        Case 75: MotifCodeBarres128 = "10110001110" ' Caractère = K / Table B = K / Table C = 43
        Case 76: MotifCodeBarres128 = "10001101110" ' Caractère = L / Table B = L / Table C = 44
        Case 77: MotifCodeBarres128 = "10111011000" ' Caractère = M / Table B = M / Table C = 45
        Case 78: MotifCodeBarres128 = "10111000110" ' Caractère = N / Table B = N / Table C = 46
        Case 79: MotifCodeBarres128 = "10001110110" ' Caractère = O / Table B = O / Table C = 47
        Case 80: MotifCodeBarres128 = "11101110110" ' Caractère = P / Table B = P / Table C = 48
        Case 81: MotifCodeBarres128 = "11010001110" ' Caractère = Q / Table B = Q / Table C = 49
        Case 82: MotifCodeBarres128 = "11000101110" ' Caractère = R / Table B = R / Table C = 50
        Case 83: MotifCodeBarres128 = "11011101000" ' Caractère = S / Table B = S / Table C = 51
        Case 84: MotifCodeBarres128 = "11011100010" ' Caractère = T / Table B = T / Table C = 52
        Case 85: MotifCodeBarres128 = "11011101110" ' Caractère = U / Table B = U / Table C = 53
        Case 86: MotifCodeBarres128 = "11101011000" ' Caractère = V / Table B = V / Table C = 54
        Case 87: MotifCodeBarres128 = "11101000110" ' Caractère = W / Table B = W / Table C = 55
```

Code

```
        Case 88: MotifCodeBarres128 = "11100010110" ' Caractère = X / Table B = X / Table C = 56
        Case 89: MotifCodeBarres128 = "11101101000" ' Caractère = Y / Table B = Y / Table C = 57
        Case 90: MotifCodeBarres128 = "11101100010" ' Caractère = Z / Table B = Z / Table C = 58
        Case 91: MotifCodeBarres128 = "11100011010" ' Caractère = [ / Table B = [ / Table C = 59
        Case 92: MotifCodeBarres128 = "11101111010" ' Caractère = \ / Table B = \ / Table C = 60
        Case 93: MotifCodeBarres128 = "11001000010" ' Caractère = ] / Table B = ] / Table C = 61
        Case 94: MotifCodeBarres128 = "11110001010" ' Caractère = ^ / Table B = ^ / Table C = 62
        Case 95: MotifCodeBarres128 = "10100110000" ' Caractère = _ / Table B = _ / Table C = 63
        Case 96: MotifCodeBarres128 = "10100001100" ' Caractère = ` / Table B = ` / Table C = 64
        Case 97: MotifCodeBarres128 = "10010110000" ' Caractère = a / Table B = a / Table C = 65
        Case 98: MotifCodeBarres128 = "10010000110" ' Caractère = b / Table B = b / Table C = 66
        Case 99: MotifCodeBarres128 = "10000101100" ' Caractère = c / Table B = c / Table C = 67
        Case 100: MotifCodeBarres128 = "10000100110" ' Caractère = d / Table B = d / Table C = 68
        Case 101: MotifCodeBarres128 = "10110010000" ' Caractère = e / Table B = e / Table C = 69
        Case 102: MotifCodeBarres128 = "10110000100" ' Caractère = f / Table B = f / Table C = 70
        Case 103: MotifCodeBarres128 = "10011010000" ' Caractère = g / Table B = g / Table C = 71
        Case 104: MotifCodeBarres128 = "10011000010" ' Caractère = h / Table B = h / Table C = 72
        Case 105: MotifCodeBarres128 = "10000110100" ' Caractère = i / Table B = i / Table C = 73
        Case 106: MotifCodeBarres128 = "10000110010" ' Caractère = j / Table B = j / Table C = 74
        Case 107: MotifCodeBarres128 = "11000010010" ' Caractère = k / Table B = k / Table C = 75
        Case 108: MotifCodeBarres128 = "11001010000" ' Caractère = l / Table B = l / Table C = 76
        Case 109: MotifCodeBarres128 = "11110111010" ' Caractère = m / Table B = m / Table C = 77
        Case 110: MotifCodeBarres128 = "11000010100" ' Caractère = n / Table B = n / Table C = 78
        Case 111: MotifCodeBarres128 = "10001111010" ' Caractère = o / Table B = o / Table C = 79
        Case 112: MotifCodeBarres128 = "10100111100" ' Caractère = p / Table B = p / Table C = 80
        Case 113: MotifCodeBarres128 = "10010111100" ' Caractère = q / Table B = q / Table C = 81
        Case 114: MotifCodeBarres128 = "10010011110" ' Caractère = r / Table B = r / Table C = 82
        Case 115: MotifCodeBarres128 = "10111100100" ' Caractère = s / Table B = s / Table C = 83
        Case 116: MotifCodeBarres128 = "10011110100" ' Caractère = t / Table B = t / Table C = 84
        Case 117: MotifCodeBarres128 = "10011110010" ' Caractère = u / Table B = u / Table C = 85
        Case 118: MotifCodeBarres128 = "11110100100" ' Caractère = v / Table B = v / Table C = 86
        Case 119: MotifCodeBarres128 = "11110010100" ' Caractère = w / Table B = w / Table C = 87
        Case 120: MotifCodeBarres128 = "11110010010" ' Caractère = x / Table B = x / Table C = 88
        Case 121: MotifCodeBarres128 = "11011011110" ' Caractère = y / Table B = y / Table C = 89
        Case 122: MotifCodeBarres128 = "11011110110" ' Caractère = z / Table B = z / Table C = 90
        Case 123: MotifCodeBarres128 = "11110110110" ' Caractère = { / Table B = { / Table C = 91
        Case 124: MotifCodeBarres128 = "10101111000" ' Caractère = | / Table B = | / Table C = 92
        Case 125: MotifCodeBarres128 = "10100011110" ' Caractère = } / Table B = } / Table C = 93
        Case 126: MotifCodeBarres128 = "10001011110" ' Caractère = ~ / Table B = ~ / Table C = 94
        Case 200: MotifCodeBarres128 = "10111101000" ' Caractère = È - Table B = Del / Table C = 95
        Case 201: MotifCodeBarres128 = "10111100010" ' Caractère = É / Table B = Fnc 3 / Table C = 96
        Case 202: MotifCodeBarres128 = "11110101000" ' Caractère = Ê / Table B = Fnc 2 / Table C = 97
        Case 203: MotifCodeBarres128 = "11110100010" ' Caractère = Ë / Table B = Shift / Table C = 98
        Case 204: MotifCodeBarres128 = "10111011110" ' Caractère = Ì - Table B = Table C / Table C = 99
        Case 205:
 MotifCodeBarres128 = "10111101110" ' Caractère = Í - Table B = Fnc 4 / Table C = Table B
        Case 206:
 MotifCodeBarres128 = "11101011110" ' Caractère = Î - Table B = Table A / Table C = Table A
        Case 207:
 MotifCodeBarres128 = "11110101110" ' Caractère = Ï - Table B = Fnc 1 / Table C = Fnc 1
        Case 208:
 MotifCodeBarres128 = "11010000100" ' Caractère = Ð - Table B = Start A / Table C = Start A
        Case 209:
 MotifCodeBarres128 = "11010010000" ' Caractère = Ñ - Table B = Start B / Table C = Start B
        Case 210:
 MotifCodeBarres128 = "11010011100" ' Caractère = Ò - Table B = Start C / Table C = Start C
        Case 211:
 MotifCodeBarres128 = "1100011101011" ' Caractère = Ó - Table B = Stop / Table C = Stop
        Case Else: Err.Raise 514
    End Select

    'Sortie de la fonction
    Exit Function

GestionErreurs:

    'Transmet l'erreur à la procédure appelante
    Err.Raise Err.Number, "MotifCodeBarres128"
```

Code

```
End Function
```

Code

```
Public Function TracerMotifCodeBarres128(Ctrl As Control, rpt As Report)

    'Déclaration des variables

    'Chaîne complète des modules du code-barres
    Dim strCodeBarres As String
    'Type d'un module : 1 = barre / 0 = espace
    Dim strTypeModule As String
    'Variable de compteur
    Dim i As Long
    'Largeur de la zone de texte
    Dim sngLargeurZoneTexte As Single
    'Valeur de la marge gauche (pour centrage) en twips
    Dim sngMargeGauche As Single
    'Coordonnée (X1) du point de départ des barres et des espaces
    Dim sngX1 As Single
    'Coordonnée (X2) du point de départ des barres et des espaces
    Dim sngY1 As Single
    'Coordonnée (Y2) du point d'arrivée des barres et des espaces
    Dim sngY2 As Single

    On Error GoTo GestionErreurs

    'Assignation de la chaîne de caractères du code-barres à la variable
    If Not IsNull(Ctrl) Then
        strCodeBarres = Ctrl
    Else
        Exit Function
    End If

    'Initialisation des coordonnées X1, Y1, Y2
    sngX1 = Ctrl.Left
    sngY1 = Ctrl.Top
    'La hauteur des modules est définie par la hauteur de la zone de texte txtCodeBarres
    sngY2 = Ctrl.Height
    'La hauteur des modules est calculée en fonction de la largeur du code-barres
    'sngY2 = lngHauteurModule

    'Calcul de la valeur de la marge à appliquer à gauche pour le centrage du code-barres
    sngLargeurZoneTexte = Ctrl.Width
    sngMargeGauche = Int((sngLargeurZoneTexte - (Len(strCodeBarres) * lngTailleModule)) / 2)

    'Décalage de la valeur de la coordonnée X1 pour le centrage du code-barres
    sngX1 = sngX1 + sngMargeGauche

    'Traçage du code-barres
    For i = 1 To Len(strCodeBarres)
        'Type de module, barre ou espace, à tracer
        strTypeModule = Mid(strCodeBarres, i, 1)
        Select Case strTypeModule
            Case "1"
                'Traçage d'une barre
                rpt.Line (sngX1, sngY1)-Step(lngTailleModule, sngY2), CouleurBarre, BF
                sngX1 = sngX1 + lngTailleModule
            Case "0"
                'Traçage d'un espace
                rpt.Line (sngX1, sngY1)-Step(lngTailleModule, sngY2), CouleurEspace, BF
                sngX1 = sngX1 + lngTailleModule
        End Select
    Next i

    'Sortie de la fonction
    Exit Function

GestionErreurs:
```

Code

```
    'Affichage d'un message
    MsgBox "Une erreur inattendue s'est produite !" & vbNewLine & "Erreur no : " _
    & Err.Number & vbNewLine & Err.Description, vbCritical, "Erreur !"

End Function
```

V - Conclusion

Il ne s'avère donc pas très compliqué de créer des codes-barres sans avoir recours à des composants externes, que ce soit une police de caractères ou un contrôle ActiveX.

Les nombreux tests effectués sur ces codes-barres conçus uniquement en utilisant Visual Basic ont démontré qu'ils sont parfaitement lisibles et soutiennent aisément la comparaison avec des codes-barres imprimés au moyen de police de caractères spéciales.

Cette solution offre en outre l'avantage de pouvoir calculer très précisemment la largeur et la hauteur de chaque module si nécessaire.

VI - Remerciements

Je tiens à remercier tous les membres de **l'équipe Office** pour leurs conseils avisés. Remerciements tout particuliers à **GAYOT** pour ses nombreux tests de lecture des codes-barres, ainsi qu'à **xxx** pour la relecture attentive de cet article.

VII - Liens

Wikipédia, code-barres 128
Code original de la création de codes-barres 39 avec Visual Basic
Code d'optimisation des codes-barres 128

VIII - Téléchargement

Base exemple (format Access 2000) (Miroir HTTP)

IX - Annexe - composition des tables du code 128

Valeur	Table A	Table B	Table C	ASCII	Caractère	Motif
0	Espace	Espace	00	0032	Espace	11011001100
1	!	!	01	0033	!	11001101100
2	"	"	02	0034	"	11001100110
3	#	#	03	0035	#	10010011000
4	$	$	04	0036	$	10010001100
5	%	%	05	0037	%	10001001100
6	&	&	06	0038	&	10011001000
7	'	'	07	0039	'	10011000100
8	(	(	08	0040	(	10001100100
9	)	)	09	0041	)	11001001000
10	*	*	10	0042	*	11001000100
11	+	+	11	0043	+	11000100100
12	,	,	12	0044	,	10110011100
13	-	-	13	0045	-	10011011100
14	.	.	14	0046	.	10011001110
15	/	/	15	0047	/	10111001100
16	0	0	16	0048	0	10011101100
17	1	1	17	0049	1	10011100110
18	2	2	18	0050	2	11001110010
19	3	3	19	0051	3	11001011100
20	4	4	20	0052	4	11001001110
21	5	5	21	0053	5	11011100100
22	6	6	22	0054	6	11001110100
23	7	7	23	0055	7	11101101110
24	8	8	24	0056	8	11101001100
25	9	9	25	0057	9	11100101100
26	:	:	26	0058	:	11100100110
27	;	;	27	0059	;	11101100100
28	<	<	28	0060	<	11100110100
29	=	=	29	0061	=	11100110010
30	>	>	30	0062	>	11011011000
31	?	?	31	0063	?	11011000110
32	@	@	32	0064	@	11000110110
33	A	A	33	0065	A	10100011000
34	B	B	34	0066	B	10001011000
35	C	C	35	0067	C	10001000110
36	D	D	36	0068	D	10110001000
37	E	E	37	0069	E	10001101000
38	F	F	38	0070	F	10001100010
39	G	G	39	0071	G	11010001000
40	H	H	40	0072	H	11000101000
41	I	I	41	0073	I	11000100010
42	J	J	42	0074	J	10110111000
43	K	K	43	0075	K	10110001110
44	L	L	44	0076	L	10001101110
45	M	M	45	0077	M	10111011000
46	N	N	46	0078	N	10111000110

Valeur	Table A	Table B	Table C	ASCII	Caractère	Motif
47	O	O	47	0079	O	10001110110
48	P	P	48	0080	P	11101110110
49	Q	Q	49	0081	Q	11010001110
50	R	R	50	0082	R	11000101110
51	S	S	51	0083	S	11011101000
52	T	T	52	0084	T	11011100010
53	U	U	53	0085	U	11011101110
54	V	V	54	0086	V	11101011000
55	W	W	55	0087	W	11101000110
56	X	X	56	0088	X	11100010110
57	Y	Y	57	0089	Y	11101101000
58	Z	Z	58	0090	Z	11101100010
59	[	[	59	0091	[	11100011010
60	\	\	60	0092	\	11101111010
61	]	]	61	0093	]	11001000010
62	^	^	62	0094	^	11110001010
63	_	_	63	0095	_	10100110000
64	NUL	`	64	0096	`	10100001100
65	SOH	a	65	0097	a	10010110000
66	STX	b	66	0098	b	10010000110
67	ETX	c	67	0099	c	10000101100
68	EOT	d	68	0100	d	10000100110
69	ENQ	e	69	0101	e	10110010000
70	ACK	f	70	0102	f	10110000100
71	BEL	g	71	0103	g	10011010000
72	BS	h	72	0104	h	10011000010
73	HT	i	73	0105	i	10000110100
74	LF	j	74	0106	j	10000110010
75	VT	k	75	0107	k	11000010010
76	FF	l	76	0108	l	11001010000
77	CR	m	77	0109	m	11110111010
78	SO	n	78	0110	n	11000010100
79	SI	o	79	0111	o	10001111010
80	DLE	p	80	0112	p	10100111100
81	DC1	q	81	0113	q	10010111100
82	DC2	r	82	0114	r	10010011110
83	DC3	s	83	0115	s	10111100100
84	DC4	t	84	0116	t	10011110100
85	NAK	u	85	0117	u	10011110010
86	SYN	v	86	0118	v	11110100100
87	ETB	w	87	0119	w	11110010100
88	CAN	x	88	0120	x	11110010010
89	EM	y	89	0121	y	11011011110
90	SUB	z	90	0122	z	11011110110
91	ESC	{	91	0123	{	11110110110
92	FS	|	92	0124	|	10101111000
93	GS	}	93	0125	}	10100011110
94	RS	~	94	0126	~	10001011110
95	US	Del	95	0200	È	10111101000
96	Fnc 3	Fnc 3	96	0201	É	10111100010
97	Fnc 2	Fnc 2	97	0202	Ê	11110101000

Valeur	Table A	Table B	Table C	ASCII	Caractère	Motif
98	Shift	Shift	98	0203	Ë	11110100010
99	Table C	Table C	99	0204	Ì	10111011110
100	Table B	Fnc 4	Table B	0205	Í	10111101110
101	Fnc 4	Table A	Table A	0206	Î	11101011110
102	Fnc 1	Fnc 1	Fnc 1	0207	Ï	11110101110
103	Start A	Start A	Start A	0208	Ð	11010000100
104	Start B	Start B	Start B	0209	Ñ	11010010000
105	Start C	Start C	Start C	0210	Ò	11010011100
106	Stop	Stop	Stop	0211	Ó	1100011101011