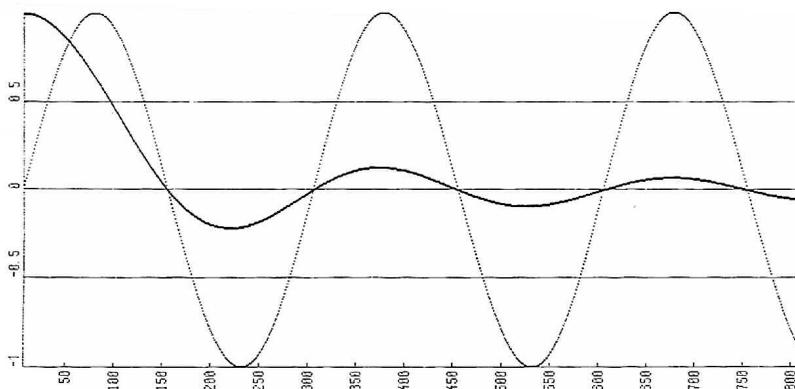


MODE GRAPHIQUE BITMAP POUR UN CONTROLEUR MATH-302x

Ce programme est une démonstration permettant de mettre en œuvre le mode graphique bitmap rapidement sur un contrôleur de la gamme MATH-302x PORTAPRINT.

Les sources sont fournis à titre d'exemple uniquement et n'engagent nullement notre responsabilité. Ils sont écrits sous Visual basic 6.0.



Le contrôleur MATH-302x permet l'utilisation de la tête MATH-3010 dans des applications graphiques jusque 200dpi, c'est à dire 384 points par ligne.

L'imprimante dispose de 5 modes graphiques :

- Mode bitmap de longueur fixe. Les 48 octets qui forment une ligne de 384 points ($48 \times 8 = 384$) sont envoyés intégralement.
- Mode Bitmap de longueur variable. Si le graphique n'occupe que la moitié de la feuille, alors 24 octets seront transmis.
- Mode Length encoded. Un octet et sa répétition sont transmis. Une ligne noire est donc codée FFh 30h, soit 2 octets.
- Mode delta Row. Seuls les octets qui diffèrent de la ligne précédente sont transmis.
- Mode TIFF. Mélange des deux modes précédents.

L'utilisation des modes graphiques compressés est avantageux dans le cas d'une liaison série car le volume de données transmis en mode graphique est important. L'imprimante peut alors fonctionner à sa vitesse maximale.

Les contrôleurs MATH-3021,2,3 disposent d'une EEPROM. Celle ci peut servir à stocker un logo ou un en-tête, qui ne sera alors plus transmis. Un mode graphique compressé permet alors de stocker une grande quantité de graphique dans l'eprom où la place est comptée (8K)

Le mode graphique Bitmap permet de mixer texte et graphique, ce qui est intéressant dans le cas d'un tracé de courbe. Le source suivant exploite cette caractéristique.

```

+-----+
| MEGATRON
| ZI noyer, BP1, 74200 ALLINGES
| tél : 04 50 70 54 54
| fax : 04 50 70 56 56
| info@megatron.fr
| http:\\www.megatron.fr
|
| Denis Stremplewski, 2002
|
| programme de démonstration en Visual Basic 6 de la mise en oeuvre
| d'une imprimante PORTAPRINT en mode graphique BITMAP
| Ce programme trace deux courbes avec un repère orthogonal gradué
+-----+
'
' Une ligne graphique fait 384 points
' la partie gauche est réservée aux abscisses, soit 3 caractères en police 7/16,
' arrondie à 24 points, soit 3 octets. La courbe commence donc en position 24
' 0 1 2 .. 24 25 26 27 28 ..... 384 : ligne physique
'      0 1 2 3 4 ..... 359 : translation nouveau repère
' Le tableau Buf contient l'image de la ligne à imprimer
' SetPix( n ) allume le point n dans le nouveau repère 0 ... 359
' ResPix( n ) éteint " " " "
' SendAndRAZ imprime la ligne, et réinitialise le tableau BUF avec le tableau
' reticule qui contient les lignes des ordonnées

Dim buf(46) As Byte
Dim reticule(46) As Byte

Dim commande As Integer
Const Cde_stop = 1
Const Cde_trace = 2

Private Sub Command1_Click()
Dim i As Integer
Dim j As Integer
Dim index As Integer

Const pi = 3.1415927

commande = Cde_trace
MSComm1.Output = Chr$(27) + "P" + Chr$(3) ' sélection du jeu n°3

' On souhaite un repère placé au quart, moitié et 3 quarts de la courbe, reticule
' est initialisé
SetPix (359 / 4)
SetPix (359 / 2)
SetPix (359 / 4 * 3)

For i = 1 To 45
    reticule(i) = buf(i)
Next i

' Impression des unités des ordonnées en utilisant des tabulations
i = 24
MSComm1.Output = Chr$(27) + "N" + Chr$(Int(i / 256)) + Chr$(i - Int(i / 256) * 256)
MSComm1.Output = "-1"
i = 359 / 4 + 24
MSComm1.Output = Chr$(27) + "N" + Chr$(Int(i / 256)) + Chr$(i - Int(i / 256) * 256)
MSComm1.Output = "-0.5"
i = 359 / 2 + 24
MSComm1.Output = Chr$(27) + "N" + Chr$(Int(i / 256)) + Chr$(i - Int(i / 256) * 256)
MSComm1.Output = "0"
i = 359 / 4 * 3 + 24
MSComm1.Output = Chr$(27) + "N" + Chr$(Int(i / 256)) + Chr$(i - Int(i / 256) * 256)
MSComm1.Output = "0.5"
MSComm1.Output = Chr$(13)

' Trace d'une ligne horizontale (axe des Y)
For i = 0 To 359
    SetPix (i)
Next i
SendAndRaz

' Boucle d'impression, les mesures doivent être entrées ici

```

```

' Toutes les 50 mesures, on imprime une unité
index = 1
Do
For j = 1 To 50
SetPix (0) ' trace un point ; ce sera l'axe X
' Il faut imprimer le texte de l'unité un peu plus tôt que le tick afin
' qu'ils soient centrés
If j = 37 Then MSCComm1.Output = Right$(Str$(index + 13), 3)
' impression d'un tick horizontal de 4 pixels
If j = 43 Then
SetPix (1)
SetPix (2)
SetPix (3)
SetPix (4)
End If
'
' Remplacer la formule par la routine d'acquisition, par exemple
'
x = Int((359 / 2 * (Sin(2 * pi / 300 * index) / (2 * pi / 300 * index) + 1)))
SetPix (x - 1) ' cette courbe sera imprimée en gras
SetPix (x) ' on allume donc des points contigus
SetPix (x + 1)

x = Int((359 / 2 * (Sin(2 * pi / 300 * index) + 1)))
SetPix (x)
SetPix (x + 1)
'
'
SendAndRaz ' la ligne est terminée, on l'imprime
index = index + 1
DoEvents
Next j
Loop Until (commande = Cde_stop)

End Sub

Private Sub Command2_Click()
commande = Cde_stop
End Sub

Private Sub Command3_Click()
MSCComm1.Output = Chr$(13)
MSCComm1.PortOpen = False
End
End Sub

Private Sub Form_Load()
MSCComm1.CommPort = 1
' 9600 bauds, pas de parité, 8 bits de données et 1 bit d'arrêt.
MSCComm1.Settings = "9600,N,8,1"
' Ouvre le port.
MSCComm1.PortOpen = True

For i = 1 To 45
buf(i) = 0
Next i

End Sub

Private Sub Form_Unload(Cancel As Integer)
MSCComm1.Output = Chr$(13)
MSCComm1.PortOpen = False
End Sub

Public Sub SetPix(n As Integer)
' Accède le tableau de 45 octets linéairement avec un adressage de 0 à 359 pixels
' et allume un point
Dim p As Integer
Dim b As Integer
p = Int(n / 8) + 1
b = 7 - Int((n / 8 - Int(n / 8)) * 8)
buf(p) = buf(p) Or power(b)
End Sub

Public Function power(n As Integer) As Integer

```

```
' retourne 2 puissance n
Dim p As Integer
p = 1
For i = 1 To n
    p = p * 2
Next i
power = p
End Function

Public Sub ResPix(n As Integer)
' Accède le tableau de 45 octets linéairement avec un adressage de 0 à 359 pixels
' et éteint un point
Dim p As Integer
Dim b As Integer
    p = Int(n / 8) + 1
    b = 7 - Int((n / 8 - Int(n / 8)) * 8)
    buf(p) = buf(p) And Not (power(b))
End Sub

Public Sub SendAndRaz()
' expédie la ligne à l'imprimante
' et recharge BUF avec le "fond d'écran " contenu dans reticule
MSComm1.Output = Chr$(27) + "G" + Chr$(0) + Chr$(0) + Chr$(0)
For i = 1 To 45
    MSComm1.Output = Chr$(buf(i))
    buf(i) = reticule(i)
Next i

End Sub
```