

Pouvoir utiliser le port série en VB.Net

Introduction

En Visual Basic 6, il existe un contrôle qui permet d'utiliser et de modifier les paramètres du port série de l'ordinateur.

En VB.Net, on ne sait pas utiliser le port série sans faire la petite modification qui suit, car ce composant n'a pas été ajouté par défaut dans le Framework 1.1.

Cependant, Microsoft annonce l'implémentation de ce composant dans le Framework 2.0.

Modifications système

Premièrement, il faut aller chercher le composant MSCOMM32.OCX sur le cd d'installation de Visual Basic 6 ou bien le chercher sur le net.

Sur le cd de VB6, il se trouve dans le dossier **OS\SYSTEM**.

Il faut alors le copier dans **c:\windows\system32**.

Une fois que c'est fait, il faut modifier une valeur dans la base de registre en utilisant le fichier **vb6controls.reg** qui se trouve sur le cd 2 d'installation de Visual Studio.Net dans le dossier **Extras\VB6Controls**.

Ce fichier permet en fait d'avoir la licence pour pouvoir utiliser ce contrôle dans le .Net.

Si vous ne faites pas la modification dans la base de registre, un message d'erreur sera affiché en vous disant que vous n'avez pas la licence.

Fermez alors votre projet .Net et rouvrez le.

Utilisation du contrôle

Pour pouvoir utiliser ce contrôle dans le Visual Studio.Net, il faut cliquer droit sur la barre d'outils et choisir le sous-menu **Ajouter/Supprimer des éléments**.

Ensuite, dans l'onglet **Composants COM**, cliquez sur Parcourir et allez chercher le contrôle **MSCOMM32.OCX** que vous avez placé précédemment dans le dossier **c:\windows\system32**. L'icône du composant est un téléphone jaune.

Et enfin, il suffit de glisser le composant de la barre d'outils dans la form de votre application.

Code simple pour une communication bidirectionnelle (machine – pc, pc – machine)

Avant de mettre en place le code que vous devez utiliser pour votre programme, vérifiez les paramètres de communication par le port COMM. On peut modifier ceux-ci par la propriété **Settings** du contrôle. Par défaut, les paramètres sont : *9600,n,8,1*.

Je ne dois pas expliquer ce que ça signifie, tout le monde le sait :
9600 Bauds, pas de parité, 8 bits de données, 1 bit de stop.

Prenons un cas concret :

Admettons que nous voulons faire communiquer le pc avec un projet d'électronique qui renvoie des données au pc via le port série.

Il faut envoyer une commande bien spécifique au projet pour qu'il renvoie le résultat de cette commande à l'ordinateur.

Voici comment ça fonctionne :

On teste si le port est ouvert. S'il ne l'est pas, on l'ouvre. (« comm » est le nom que j'ai donné au composant : son nom par défaut est AxMsComm1)

```
If comm.PortOpen = False Then
    Try
        comm.PortOpen = True
    Catch ex As Exception
        MsgBox("Problème de communication.", "Connection")
    End Try
End If
```

Le port est bien ouvert. On peut alors envoyer une commande depuis l'ordinateur vers le projet.

```
comm.Output = "*G01$C0$ R" & vbCrLf
```

Dès que la commande est envoyée, le projet fait son travail et renvoie le résultat de cette commande à l'ordinateur. Il faut donc récupérer ce résultat. Ceci est possible en utilisant la fonction **input** du composant.

```
Textbox1.Text = comm.Input
```

Ces deux instructions (Output et Input) sont généralement les seules instructions qu'on utilisera pour communiquer avec le port RS232. Cependant, selon les besoins, on peut avoir besoin d'utiliser d'autres fonctions et propriétés du composant.