TRAVAUX PRATIQUES
LIAISON DDE

I – Utilisation des fonctions DDE Excel

1) Lien DDE entre 2 applications d'une même machine

Ouvrir deux instances d'Excel

Ecrire dans la cellule A1 de l'instance 1 d'Excel la fonction de liaison DDE (lecture) avec la cellule A1 de l'instance 2.

Vérifier le fonctionnement avec divers types de données: valeur numérique, chaîne , absence de données. Y a-t-il transmission du formatage de la données (taille, style...)?

Sauvegarder les deux instances, les fermer. Rouvrir l'instance 1 et observer le fonctionnement à l'ouverture. Supprimer la sauvegarde de l'instance 2 et rouvrir l'instance 1. Conclure sur l'ouverture automatique du lien DDE sous Excel.

2) Liens croisés entre 2 applications d'une même machine

Ouvrir deux instances d'Excel et placer lune valeur numérique quelconque (entière ou fractionnaire) dans la cellule A1 de l'instance 2.

Ecrire dans la cellule A1 de l'instance 1 d'Excel la fonction de liaison DDE (lecture) avec la cellule A1 de l'instance 2.

Ecrire dans la cellule A2 de l'instance 2 d'Excel la fonction de liaison DDE (lecture) avec la cellule A2 de l'instance 1.

Pour l'instance 1, écrire la formule permettant le calcul A2 = A1 +1 .

Pour l'instance 2, écrire la formule permettant le calcul A1= A2 + 1 .

Expliquer le fonctionnement constaté d'Excel en client et serveur. En déduire la vitesse de rafraîchissement des formules sous Excel.

Sauvegarder les deux instances, les fermer puis rouvrir l'instance 1 . Expliquez les problèmes rencontrés.

3) Lien entre 2 applications sur 2 machines distantes

La manipulation se fait deux par deux, un binôme tentant de se connecter sur la machine de son voisin et réciproquement.

Ouvrir une instance d'Excel et intituler le classeur avec le nom de votre choix. Placer une valeur numérique quelconque dans la cellule A1 .

Paramétrer le service NetDDE pour que votre binôme voisin puisse se connecter (lui fournir le nom d'application que vous avez choisi).

Ecrire dans A2 de votre instance "client" la formule permettant de lire la cellule A1 de l'instance "serveur" de votre voisin.

Vérifier le bon fonctionnement réciproque client/serveur.

Placer une valeur numérique dans la cellule A1, écrire la formule permettant A1 = A2 + 1 et observer le fonctionnement. Conclure sur le fonctionnement en réseau.

II – Utilisation du langage VBA

1) Prise en main

Ecrire une fonction [macro] permettant d'afficher le message "bonjour" à l'écran. Déclencher son exécution directe et vérifier son bon fonctionnement.

Insérer un bouton permettant d'exécuter la fonction.

2) Lien DDE déclenché

Ouvrir deux instances d'Excel. Insérer dans l'instance 1 un bouton associé à une macro permettant de lire la cellule A1 de l'instance 2.

3) Lien DDE cyclique

Mettre en place le mécanisme de Timer VBA, avec pour tâche cyclique la lecture de la cellule A1 de l'instance 2. Le résultat sera placé dans la cellule A1. Le timer sera contrôlé par un bouton "Start" et un bouton "Stop"

Visualiser la valeur lue par un graphique du type "barre verticale" . Vérifier le fonctionnement dynamique.

Modifier le programme pour que les valeurs soient écrites en A1, puis A2... Associer à la plage de cellules un graphique du type "courbe" . Vérifier le fonctionnement dynamique de l'ensemble. Compléter en implantant un bouton de réinitialisation

4) Mini projet serveur Modbus

Objectif: réaliser un serveur DDE pour une carte E/S Modbus, du type utilisé en enseignement à l'USTL.

Pour disposer de la liaison série sous VBA, il faut implanter l'OCX MScomm32 programme fourni avec le VB.

Mettre en place l'écriture puis la lecture sur le port série de façon à lire les bits d'entrée ou les registres internes de la carte Modbus. Les données seront disposés sur la feuille Excel afin d'être à disposition d'applications externes par lien DDE.

PROGRAMMATION VBA

La programmation VBA a pour objet de développer des applications complexes dans l'environnement Office. Le langage de base est le Visual Basic complété par de nombreuses fonctions relatives aux applications Excel, Word ou Access.

A la différence du VB classique qui génère une application autonome, le VBA s'exécute dans l'environnement pour lequel il a été développé.

Dans ce cours de supervision, le VBA est développé sous Excel, qui permet de manipuler facilement des listes de données et d'en donner une représentation graphique.

Le code VBA s'écrit à l'aide du Visual Basic Editor intégré à Office. Son activation se fait depuis Excel par alt+F11 ou par le menu "Outils"→"Macros"→"Visual Basic Editor" .

L'éditeur possède une aide à la construction du programme ainsi que des navigateurs intégrés (projet, objets…).

Fonction/Sous-programme VBA

Dans une application Excel, l'utilisateur peut définir des sous-programmes (pas de retour) ou des fonctions (valeur en retour) .

Une fonction/sous-programme est associée à une *feuille* ou un *classeur*. La portée des variables déclarées et fonctions est limitée à l'environnement choisi.

Exemple de sous-programme:

```
Sub bonjour()
' Première macro: Hello World (version2)
'   MsgBox ("Hello, world !")      'INCORRECT: MsgBox() est une  fonction (réponse =  MsgBox…)
   MsgBox "Hello, world !",,"Première macro"  MsgBox est ici une action (voir aide)
End Sub
```

Exemple de fonction

```
Function repOuiNon(ByVal message As String) As Integer
' Première macro fonction avec arguments
   repOuiNon = MsgBox(Prompt:=message, Buttons:=vbYesNoCancel)
End Function
```

Bouton de commande dans Excel

Pour créer un bouton de commande (ou un autre objet graphique) dans une feuille, sélectionner dans le menu "Barre d'Outils" le module "Commandes" qui se présente comme suit :

Dans cet écran, sélectionner la fonction "Bouton de commande" et tracer le bouton de commande à l'emplacement souhaité.

Un "click" sur ce bouton activera un sous-programme intitulé "Nom_du_bouton_Click()"dans le module associé à la feuille courante. Cette fonction n'est pas accessible depuis les autres feuilles (à moins de préciser dans l'appel qu'elle se trouve dans cette feuille).

Macro-Commandes

Depuis la version 97 d'Office, les macro-commandes sont des sous-programmes particuliers. La syntaxe est donc exactement celle du VBA. Le code des macros est enregistré dans les "modules". Les macros sont construites par un *générateur de code automatique.*

Pour construire une macro, sélectionner dans le menu "Outils" →Macros"→"Nouvelle macro. Excel passe en mode "Générateur de code" une fois le nom de la macro défini. Toute action sur la feuille Excel en cours (ou dans une autre feuille) est enregistrée sous forme d'une suite de commandes VBA. Lorsque l'utilisateur a terminé la construction, choisir "Fin d'enregistrement" dans le menu "Macros".

Le code construit est visualisable dans l'éditeur VBA. L'utilisateur a la liberté de modifier le code construit par le générateur (sauf le nom de la procédure).

Activation d'une macro par un bouton

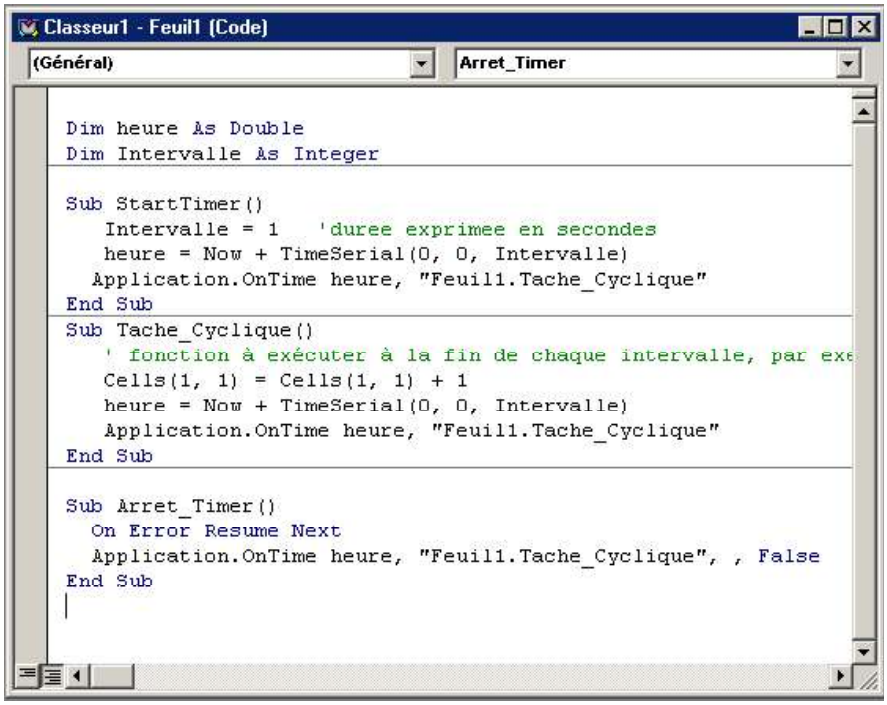
L'exécution d'une macro peut être déclenchée par un événement lié à un objet graphique.

Pour déclencher une macro par un bouton, sélectionner "Formulaires" dans le menu "Barre d'outils".

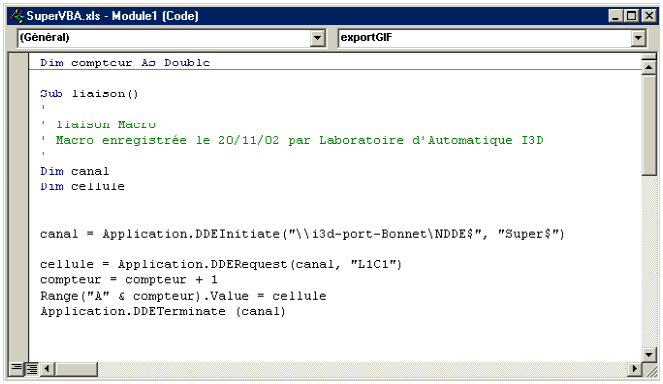
Sélectionner la fonction "Bouton de commande" et tracer le bouton à l'emplacement souhaité.

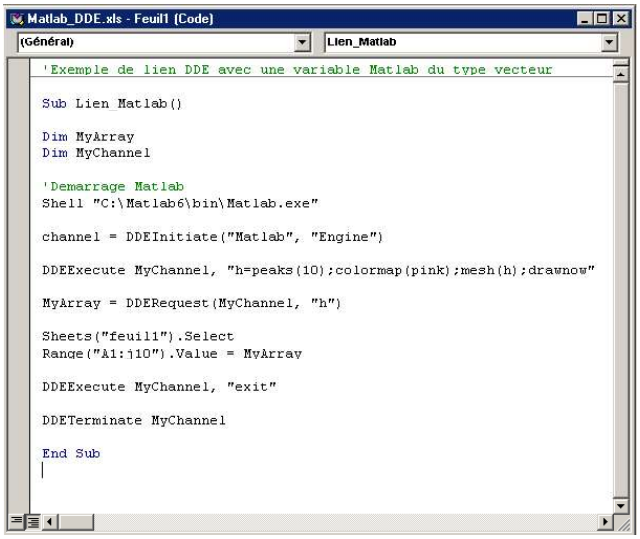
L'affectation du bouton à une macro se fait par le click droit de la souris.

Exécution d'une tâche cyclique sous Excel.



Exemples de lien DDE sous Excel





Un site intéressant pour la programmtion VBA : http://www.cathyastuce.com/vba.htm

Utilisation du port COM

Le port série n'est pas directement accessible depuis une application; il nécessite d'utiliser une bibliothèque spécifique de fonctions:

- méthode "createfile" depuis NT4 (ne fonctionne pas sous 95) pour le langage C
- bibliothèque OCX MSComm, faisant partie de la fourniture VB.