DEVELOPPEMENT D'UNE IHM BASEE SUR LE STANDARD OPC

Serge Bouter[1] et Rachid MALTI[1,2]

[1]IUT Bordeaux 1, 15 rue Naudet, CS 10207, 33175 GRADIGNAN Cedex

[2]IMS UMR 5218 CNRS, Département LAPS – Université Bordeaux I
351, cours de la Libération – F 33405 Talence cedex – France
http://www.iut.u-bordeaux1.fr

Objectif

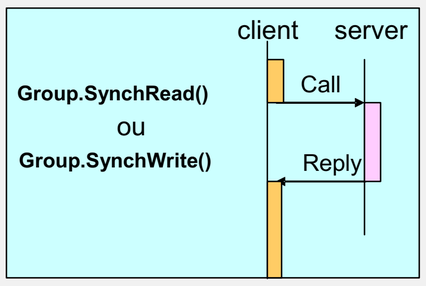
- ✓ Utilisation du standard OPC à travers le développement d'un client OPC à partir d'un langage de programmation événementielle (VB, VC++ ou C#) sous l'environnement Visual Studio .NET
- ✓Mise en œuvre rapide d'une interface homme/machine conforme au standard OPC
- ✓Montrer les avantages du standard OPC.
- ✓Montrer l'interactivité entre le monde des automates programmables industriels et celui des micro-ordinateurs de bureau.

Public visé

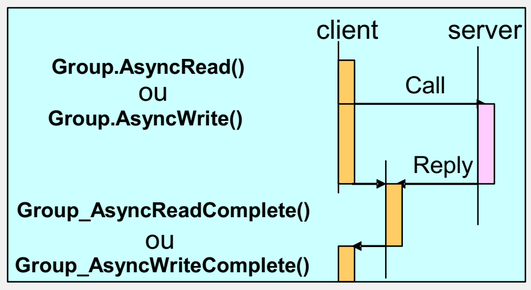
- ✓**Licence Professionnelle** Automatique et Informatique Industrielle – **spécialité** : Systèmes Automatisés & Réseaux Industriels (SARI) de l'IUT de Bordeaux 1
- ✓**Master** Génie des Systèmes Electroniques et Mécaniques pour l'Aéronautique et les Transports – **spécialité** : Génie des Systèmes Electroniques et Mécaniques pour l'Aéronautique et les Transports de l'université de Bordeaux 1.

Standard OPC DA

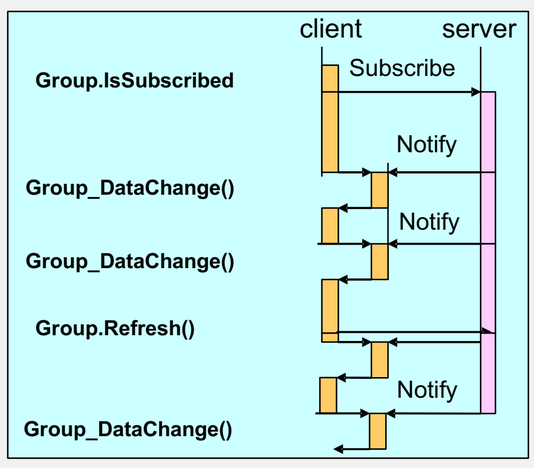
L'objectif des différentes normes OPC est de standardiser les échanges de flux entre équipements communicants sur des réseaux locaux industriels hétérogènes. Les différents modèles de Lecture/Ecriture sont schématisés ci-dessous



Lecture ou Ecriture Synchrone
Une opération synchrone est bloquante, c.-à-d. que le client attend la réponse du serveur



Lecture ou Ecriture Asynchrone
Une opération asynchrone n'est pas bloquante.

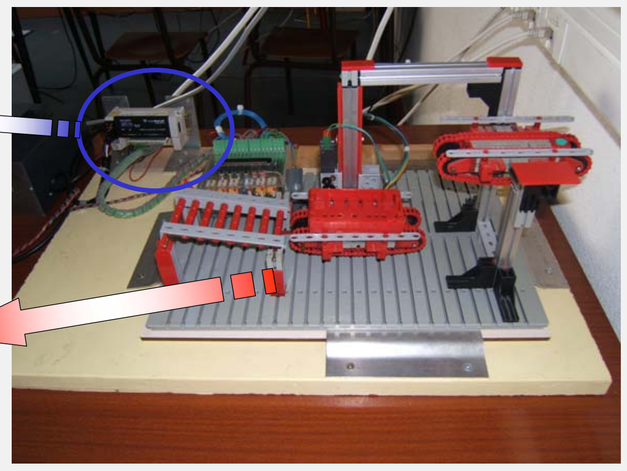


Lecture par notification. A chaque changement d'état d'une variable, le serveur OPC envoie une notification aux clients. Cet mode de lecture est nettement plus avantageux que les deux précédents.

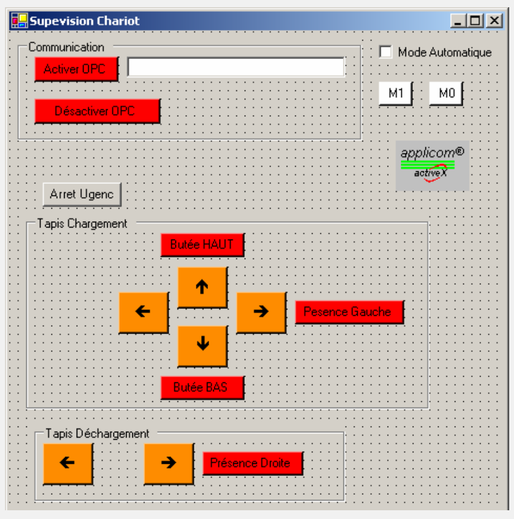
Description de la maquette

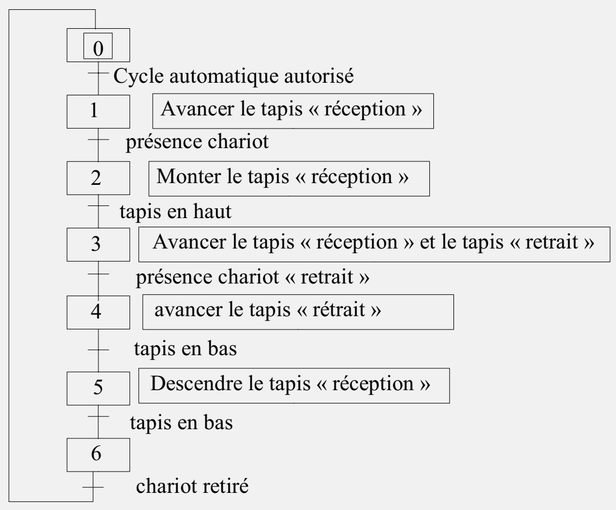
Module de 8 entrées et 8 sorties TOR déportées fonctionnant sous DeviceNet

Maquette "Chariot élévateur" comprenant 4 capteurs et 4 actionneurs



Développement de l'IHM

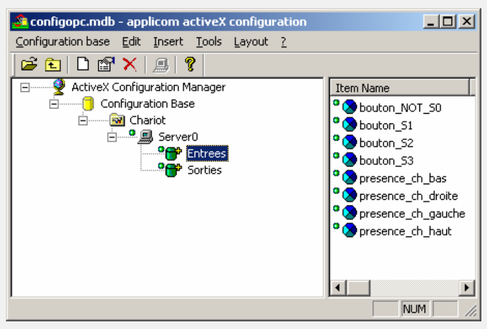




Programmation événementielle en relation avec l'IHM

1ère étape

Configuration du serveur OPC



2ème étape – Initialisation du serveur OPC – sous .NET

```
Retour = Client.ActiveConfig
RefServer = Client.GetServerRef("Server0")
    RefGSorties = Client.GetGroupRef(RefServer, "Sorties")
Name = "ch_droite_vers_droite"
RefItemS_d_d = Client.GetItemRef(RefGSorties,Name)
```

3ème étape – Exécution des fonctions de lecture et/ou écriture asynchrones

```
Dim tabStatus As Object
Dim retour As Long
retour = Client.Write(1, RefItemS_d_d, valeur, tabStatus)
If retour = 0 Then
status.Text = "Write ok"
Else
status.Text = "Write bad"
End If
```

✓

4ème étape – Fonction de notification

```
Private Overloads Sub Client_EventNewValue(...) ...
...
For i = 1 To e.nbItems  Name = "ch_droite_vers_droite"
    status.Text = e.tabItemName(i)
    Select Case e.tabItemName(i)
    ...
        Case "Server0.Sorties.M0«
          flagM0 = e.value(i)
          If flagM0 Then
              M0.BackColor = Color.Red
          Else
              M0.BackColor = Color.White
          End If
        Case "Server0.Sorties.M1«
            flagM1 = e.value(i)
            If flagM1 Then
                M1.BackColor = Color.Red
            Else
                M1.BackColor = Color.White
            End If
    ...
        End Select
    ...
End For
```