

Compte-rendu TP Supervision

Supervision d'un ascenseur



Nicolas BAËLE TP3

Sommaire

Table des matières

Introduction.....	3
Variables et programme.....	3
Fenêtre et objet générique	5
Synoptique global.....	7
Serveur OPC.....	8
Conclusion	9

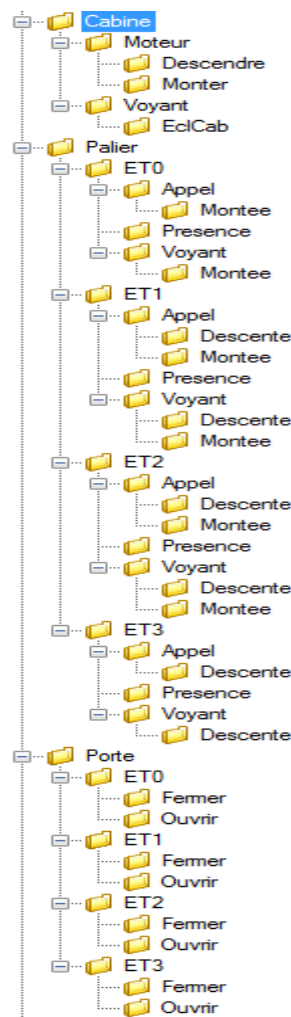
Introduction

Objectif : Superviser un ascenseur programmé en S2 en automatisme, avoir une vue d'ensemble et un contrôle sur les actions de l'ascenseur.

La supervision d'un système est complètement différente de la programmation d'un même système. En effet elle vient après la programmation et à un but plus de contrôle et d'information que la programmation pure.

Variables et programme

Tout d'abord, nous avons dû réaliser l'instanciation à plat des variables sous forme d'arborescence, avec comme domaines : RDC, ET1, ET2, ET3, Cabine, et pour les natures nous avons : Appel, Fermer, Ouvrir, Presence, Voyant.



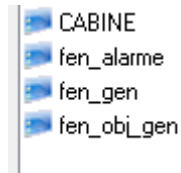
Pour que le programme puisse fonctionner, nous avons créer un programme qui relis les ETAT aux CMD :

```
sub main()  
end sub  
  
sub relation()  
ETAT=CMD;  
end sub  
  
sub relation1()  
@Palier.ET0.Appel.Montee.ETAT=@Palier.ET0.Voyant.Montee.CMD;  
@Palier.ET1.Appel.Montee.ETAT=@Palier.ET1.Voyant.Montee.CMD;  
@Palier.ET2.Appel.Montee.ETAT=@Palier.ET2.Voyant.Montee.CMD;  
@Palier.ET1.Appel.Descente.ETAT=@Palier.ET1.Voyant.Descente.CMD;  
@Palier.ET2.Appel.Descente.ETAT=@Palier.ET2.Voyant.Descente.CMD;  
@Palier.ET3.Appel.Descente.ETAT=@Palier.ET3.Voyant.Descente.CMD;  
@OBS = @OBS_ch;  
  
end sub  
sub relation2()  
@Bit02=@Bit01;  
end sub  
  
sub ouv_alarm()  
WINDOW("OPEN","fen_alarme","");  
end sub
```

Le sub relation1() relis de façon précise les variables écrites, mais le sub relation() permet de relier les ETAT aux CMD si on écrit la bonne branche dans l'édition de la variable, c'est un programme plus global.

Fenêtre et objet générique

Nous n'allons pas réexpliquer les bases de la supervision sur PCVUE, nous nous contenterons de parler des nouvelles fonctionnalités. Par exemple le système de synoptique générique. Nous avons plusieurs synoptiques :



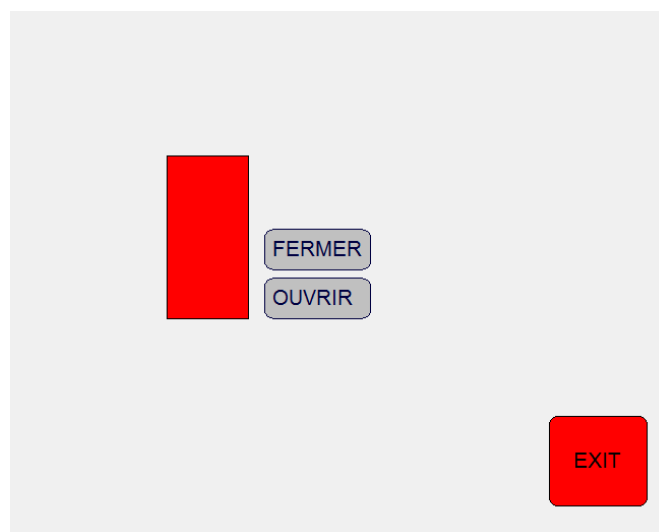
Nous avons tout d'abord le synoptique dit « global », CABINE, il contient tous les éléments qui composent notre TP.

Ensuite fen_alarme est une fenêtre l'alarme qui va s'ouvrir quand il y a une modification sur la variable d'alarme, grâce au sub ouv_alarm() du programme précédent.

Puis fen_gen qui est simplement la fenêtre générique des portes, en effet par une action de chaînage qui sera expliquée par la suite, on peut ouvrir une même fenêtre avec une branche différente, ceci permet de réduire le nombre de synoptiques.

Et enfin fen_obj_gen, ce synoptique a été utilisé pour créer les différents symboles génériques dont nous avons besoin dans ce TP.

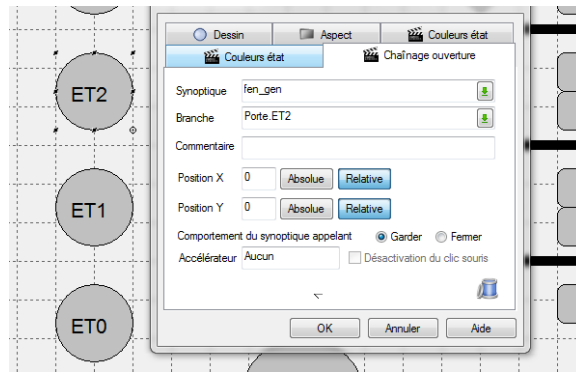
Le synoptique fen_gen est donc un synoptique générique car il permet d'ouvrir chaque porte dans un seul synoptique :



Ce que l'on voit ici est appelé un objet générique, en effet il a été créé au préalable dans fen_obj_gen et peut maintenant être instancié où on le souhaite en un simple « clic ».

On peut donc observer que la porte est fermée et que l'on peut décider soit de l'ouvrir soit de la fermer. Le bouton EXIT étant un chaînage de fermeture.

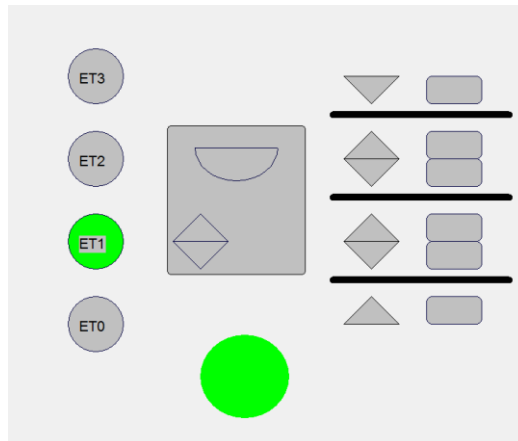
Or, ceci n'est possible que grâce à une action de chaînage depuis le synoptique global.



On peut voir ci-dessus un chaînage d'ouverture où l'on renseigne le synoptique que l'on souhaite ouvrir et quand quelle branche. Ici on aura pour effet de préremplir les variables de fen_gen avec Porte.ET2 et donc avoir une fenêtre branchée. Pour fermer un synoptique on utilisera un chaînage de fermeture ou cette fois ci on notera dans la branche « * ».

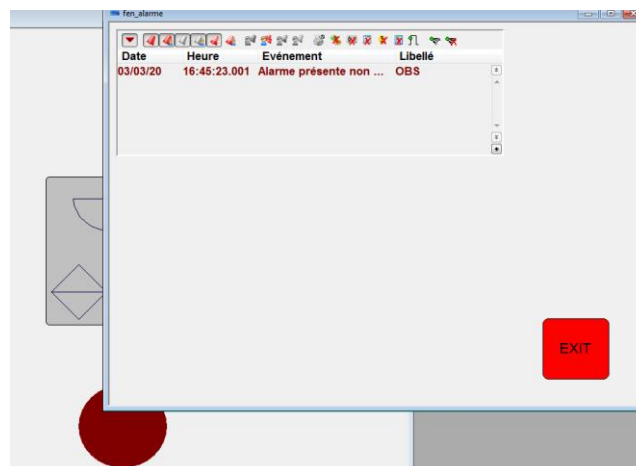
Synoptique global

Tout ceci nous amène donc a notre synoptique global, CABINE :



On voit donc les boutons ET0, ET1, ET2, ET3 qui sont les chaînage d'ouverture vers les différentes portes dans la fen_gen. On observe au centre l'objet générique CABINE qui n'a pas réellement d'utilité car il n'a été instancié qu'ici. Cependant sur cette cabine on peut noter une commande pour la lumière, et deux boutons monter et descendre. Sur la droite, on observe des boutons de commandes pour piloter la cabine en montée ou en descente depuis n'importe quel étage, ainsi que les témoins lumineux qui leurs sont attirés.

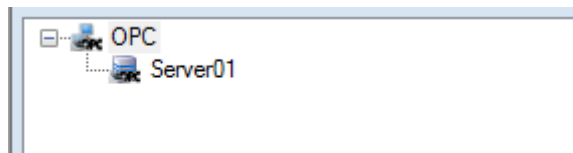
Le bouton du bas quant à lui, sert de simulation pour un éventuel obstacle, et déclenche donc un changement de la variable d'alarme et donc cela ouvre la fenêtre d'alarme suivante :



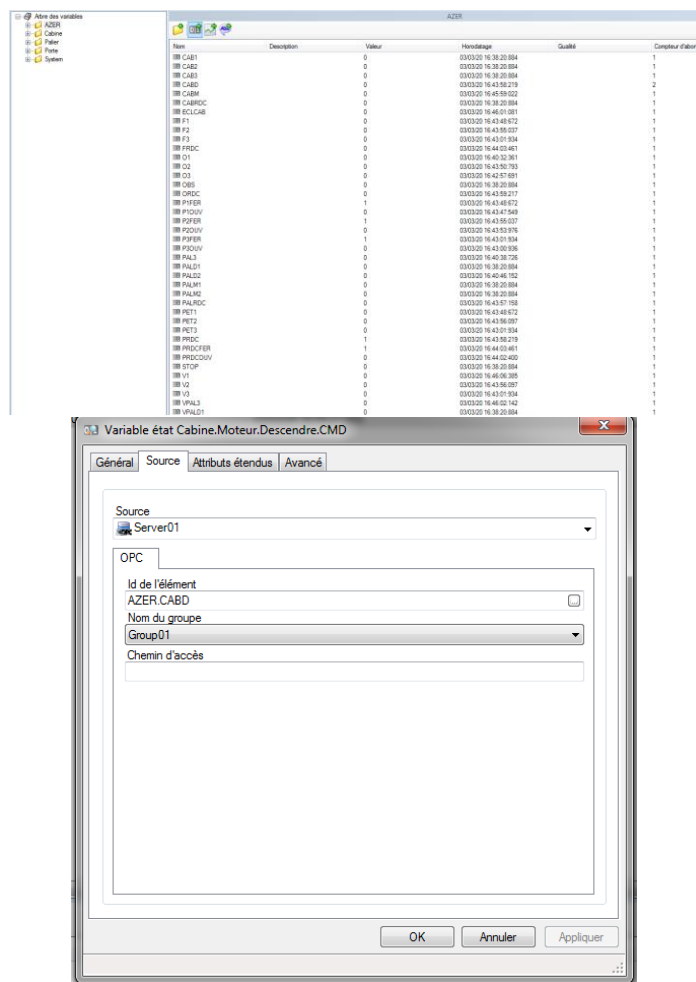
Ici, l'alarme est présente et non acquittée.

Serveur OPC

Une fois cela fait en simulation, on doit utiliser notre supervision pour superviser l'automate de l'ascenseur, pour ceci nous allons faire ce qui s'appelle un serveur OPC qui va permettre la communication entre la supervision et les différents clients.



Nous avons donc récupéré toutes les variables prises par l'automates. Pour que notre système fonctionne sur l'automate, il faut relier les variables avec celles de notre supervision, pour cela nous allons changer le source des variables pour qu'elles viennent du serveur. Nous pourrons ainsi spécifier la variable à attribuer.



Une fois ceci fait, nous simulons un client OPC qui va venir se connecter sur notre serveur pour tester le bon fonctionnement de notre supervision.

Conclusion

En conclusion, grâce a PCVUE nous avons appris a superviser un système déjà existant et complexe. Mais dans ce TP particulièrement nous avons réussis à créer un serveur OPC et a créer un client pour tester notre supervision. Ce serveur OPC est d'autant plus important qu'il simule le fonctionnement réel si nous avons étés en entreprise.