

TP Automatismes

Mise en oeuvre de WinCC Flexible

Introduction à la mise en oeuvre d'une supervision

WinCC permet une supervision, par un ou plusieurs PC indépendants, ou écrans tactiles (HMI), nommés OS (station opérateur) d'un ou plusieurs automates (AS). La supervision consiste à « regarder » le process (via les E/S des AS et donc évidemment des capteurs, mais aussi les sorties et les mémoires internes), et de mettre les informations sous la forme la plus utile (graphique, tracés de courbes...)... Elle peut également influencer sur le process (via l'AS) en fixant des valeurs dans des mémoires internes (à condition que le programme écrit dans l'AS en tienne compte).

Il y a deux approches possibles pour mettre en oeuvre une supervision sous Simatic Step7:

a) Sur le PC, on appelle WinCC flexible, dans lequel on définit les vues (le graphisme) que l'on prévoit, ainsi que les variables (entrées/sorties/mémos et DB) des différents automates qu'on va superviser. Les automates, quand à eux, sont programmés indépendamment (en Step7 par exemple), il faut simplement que le programme automate prévoie de mettre à jour les variables utilisées par la supervision.

b) Sous STEP7, on crée un projet regroupant l'ensemble : programmes des automates et supervision. Cette deuxième approche me semble préférable, car on gère le problème globalement. Rien que dans le cas de mises à jour des programmes, on évitera le risque d'avoir des versions différentes sur différents matériels.

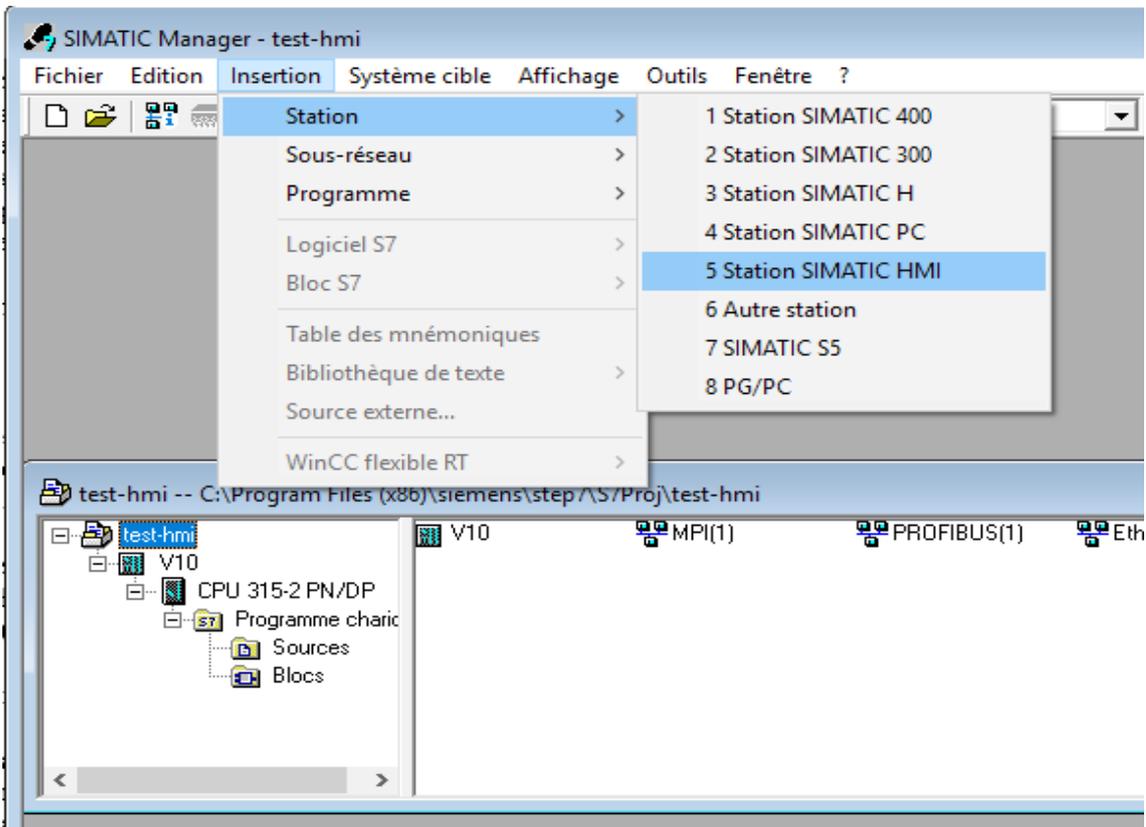
Choisissez donc plutôt le second cas. On crée sous STEP7 un projet. On y insère d'abord les différents matériels (AS : automates et leurs E/S, liaisons réseau). Dans chaque AS on définit les programmes (donnez un nom explicite, plutôt que « programme S7 », il servira pour définir les variables) et les mnémoniques. Dans ces mnémoniques, il faut en particulier définir les variables qui seront supervisées : sélection de la ou les lignes de mnémoniques concernés, clic droit -> propriétés spécifiques -> contrôle-commande puis cocher la case contrôle-commande). Rien ne vous interdit de faire un copier-coller (ou glissement de souris) pour récupérer des mnémoniques d'un projet antérieur.

	Etat	Mnémonique ▲	Opérande	Type de d	Corr
1	<input checked="" type="checkbox"/>	entree-capt	E 0.0	BOOL	
2	<input checked="" type="checkbox"/>	sortie	A 0.0	BOOL	
3	<input type="checkbox"/>	memo-local	M 10.0	BOOL	
4	<input checked="" type="checkbox"/>	memo-public	M 10.1	BOOL	
5	<input type="checkbox"/>				

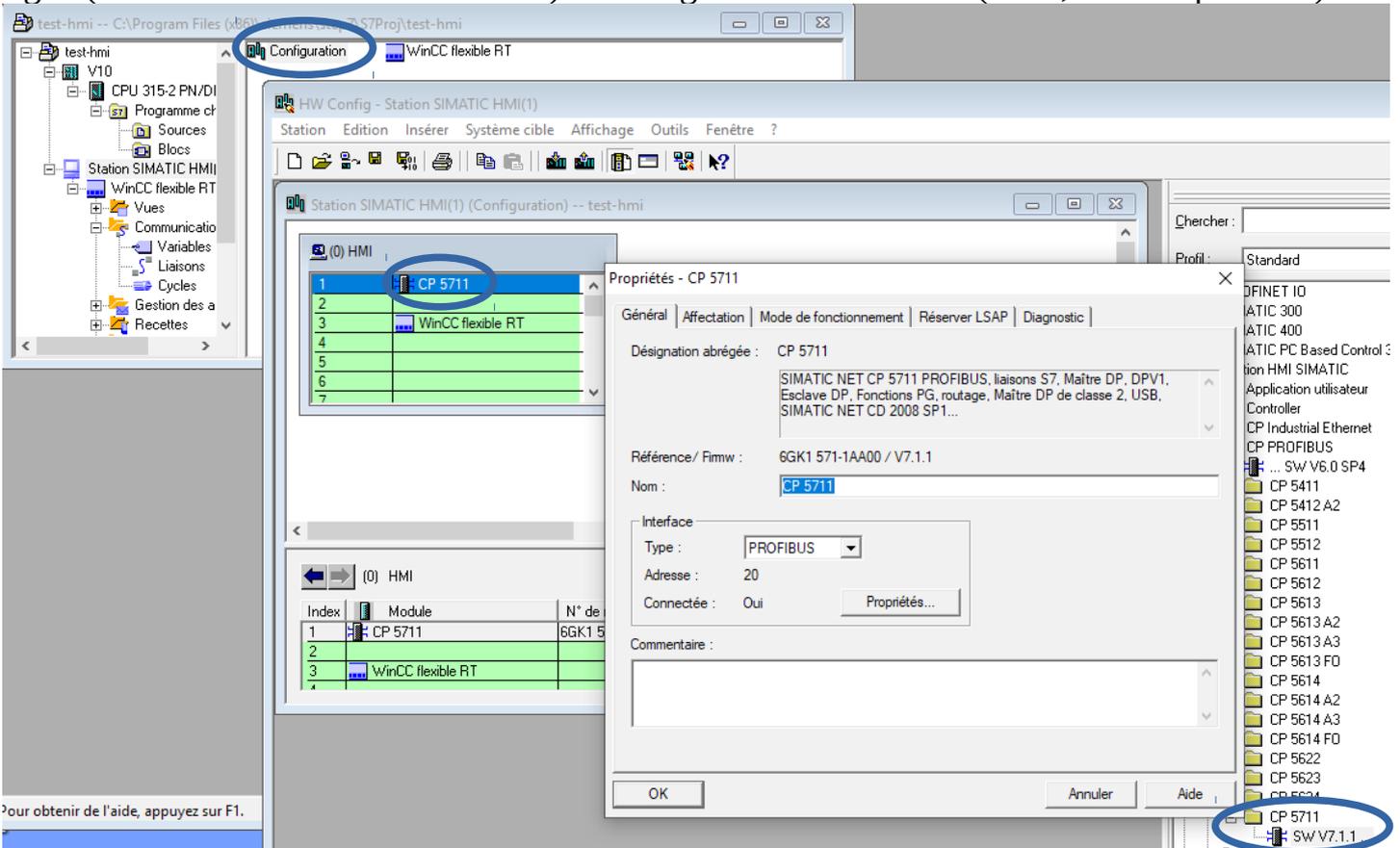
Avant de commencer, une précision : nos PC sont reliés au réseau Profibus soit par une carte interne (les plus vieux) soit via un boîtier externe via USB. Les automates sont aussi reliés à Profibus (câble violet). Ce n'est pas parce que l'endroit le plus proche pour relier le PC au réseau est au niveau d'un automate que le PC a une dialogue privilégié avec celui-ci. Tous les AS sont accessibles de la même manière, depuis tout OS, tant que Profibus fonctionne : bien configuré, bonne vitesse de transmission, jamais 2 appareils de même numéro (ne donnez pas au boîtier de l'OS le même numéro que l'AS).

Insertion d'une station opérateur (OS) dans le projet :

Au niveau supérieur du projet, insérer (clic droit ou menus) un OS (ici : de type HMI -> PC -> Runtime).

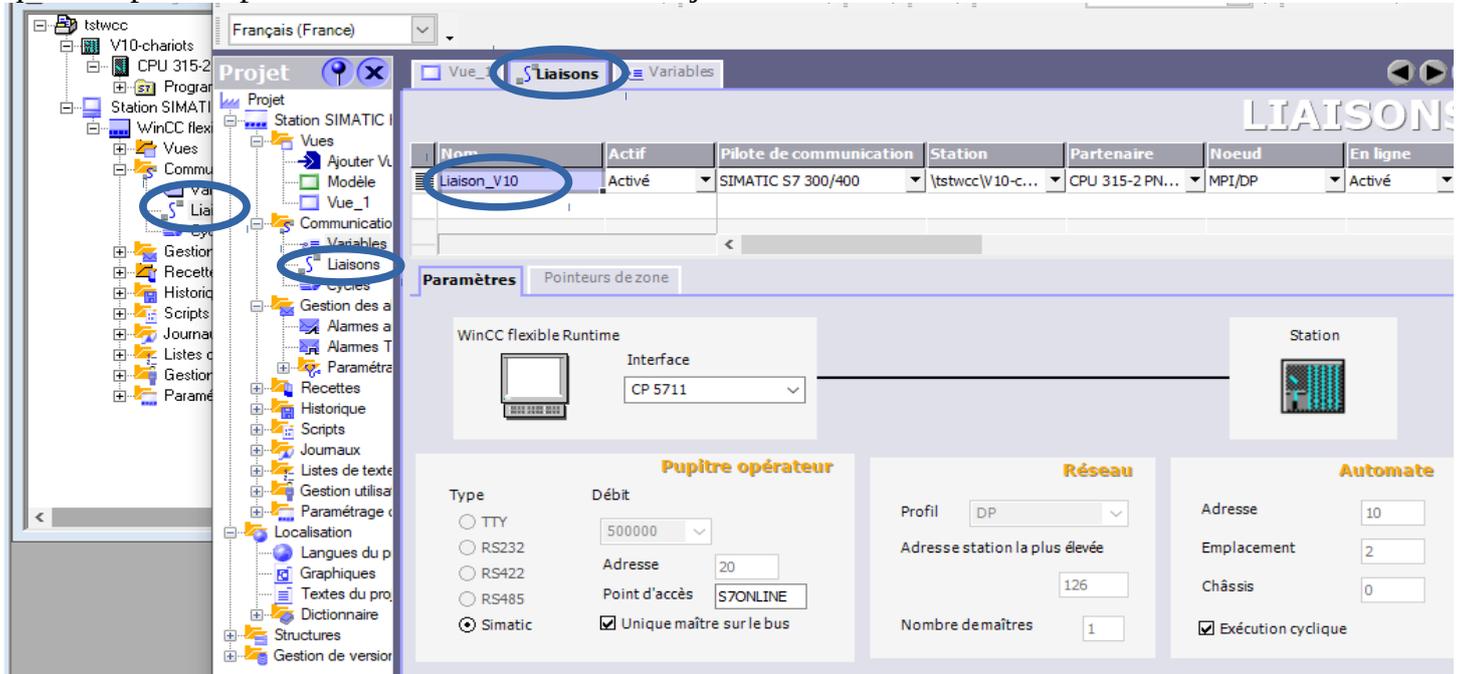


Puis il faut définir comment votre PC est relié au réseau. Ici nous passons (via USB) sur Profibus via un boîtier qui a, lui aussi, un numéro unique sur Profibus (marqué dessus, souvent le n° de valise+10). Donc cliquez sur l'OS → configuration, et insérez le boîtier en première ligne (HMI → CP Profibus → CP5711) et configurez bien le réseau (500k, adresse profibus).



Il faut aussi définir quel est l'automate supervisé (il peut y en avoir plusieurs !). Sur la station, cherchez « communication → liaisons ». Il devrait l'avoir créé automatiquement (vérifiez les paramètres), sinon créez une nouvelle liaison (de type MPI/DP ou Ethernet suivant le mode de raccordement du PC au réseau des automates, et définissez les paramètres). Activez la liaison s'il ne l'a pas fait (vous pouvez la renommer avec le nom de votre valise). Enregistrez, je crois

qu'il en profite pour mettre diverses choses à jour.



Maintenant, on peut récupérer directement les contrôle-commande (variables) définis dans les AS : dans « variables », créez vos variables, en récupérant les données dans les symboles de votre AS (choisissez le mnémonique en déroulant l'arborescence dans le projet, il rajoute tout seul les autres données). Ici aussi, je crois qu'il vaut mieux enregistrer.

Insertion de la vue de supervision

Dans notre OS, on doit insérer une vue (ou utiliser Vue_1 si elle existe). Ce sera l'écran graphique que l'on verra sur l'OS lorsqu'il supervisera les AS). Puis on double clique dessus, on entre dans la partie graphique. On peut y placer tous objets graphiques plus ou moins simples.

Insertion d'un objet simple :

Entrer par exemple un texte statique (champ de texte) : on le place dans la page, on remplit le texte (un titre par exemple). Puis on va définir plus précisément ses propriétés : par exemple représentation → couleur arrière plan, texte → alignement horizontal - > centré (idem en Y). Certaines propriétés peuvent dépendre de variables, par exemple la couleur, le fond, la position,...

Créons un texte donnant le nom d'une variable que l'on désire superviser. Puis mettre en face un petit cercle. Animations → représentation : on choisit la variable (dans la station HMI, pas dans le projet Step7). Si c'est un Tout ou Rien, choisir Bit, puis définir une couleur (d'arrière plan) pour la valeur 0, une autre pour 1.

On peut aussi faire clignoter le texte si la variable est vraie (clignotement du texte activé).

On peut également associer la position d'un objet à une variable numérique : une valeur calculée dans l'automate, mais aussi l'entrée analogique (potentiomètre de notre console ou la roue codeuse).

Insertion d'un objet complexe :

Insérons maintenant un objet complexe : champ d'E/S. Il est associé à une variable (numérique). Dans les propriétés, on définit couleurs, police... mais surtout dans Général → Entrée/Sortie il faut définir la base (décimal, binaire, hexa) et le format (par exemple 999,99 pour 3 chiffres maxi avant la virgule et 2 après, 000,00 pour toujours 3 et 2 chiffres, 9999 pour des entiers, 99,99e99 pour une notation scientifique). Un champ de sortie nous affiche l'état d'une variable, une entrée permet de forcer une valeur dans l'automate.

Une E/S est généralement associée à un mémoire (M0.0 ou MW0...), dont la valeur est affichée et peut être modifiée depuis l'OS. Supervisez par exemple un timer : dans le programme automate, mettre la valeur de la tempo dans un mot double (MD, 32 bits non signé). Du côté supervision, dans le champ d'E/S, choisir le format décimal 99999. Le temps sera donné en ms, mais on peut

facilement le changer : dans les propriétés de la variable, on peut utiliser la mise à l'échelle pour par exemple donner les temps en s.

Au lieu d'afficher la valeur, on peut aussi insérer un Bargraphe, une flèche dont l'extrémité bouge...

Il est possible d'insérer des objets encore plus complexes. Dans les Graphiques, une quantité énorme d'objets est disponible : cadrans à aiguilles, tachymètres, thermomètres, horloges, moteurs et vérin. Également une bibliothèque de symboles (vannes, relais...). Voir un camion ou un avion, si vous supervisez également la logistique. Regardez également les bibliothèques (clic droit, ouvrir, biblio système...).

On peut également allumer un bit (M0.0 par ex) sur l'automate via la supervision, par un champs d'E/S bien sûr (format binaire), mais également avec des objets graphiques comme des commutateurs (ouvert/fermé), un bouton (evenements → presser → AssignerValeur → nomDeLaVariable → 1). Il faut évidemment que votre programme prenne en compte ce bit si vous voulez que cela serve à quelque chose (démarrer un cycle par exemple).

Sur la version précédente du logiciel, on pouvait aussi faire des calculs sur les variables (il connaît toutes les instructions du langage C) : par ex $243 * (\text{variable_1} - 10)$, ou $(1 - \text{variable})$ si on veut inverser un booléen. Tout calcul en C devrait être possible, par exemple $10 * \text{variable_200}$, voire des if, des fonctions (il a inclus par défaut math.h, string.h, stdlib.h...). Mais je n'ai pas encore réussi à le refaire.

Activation de la supervision

Une fois tout prêt, enregistrer la vue ou (Projet → compilateur → générer). Puis lancer le runtime (Projet → compilateur → runtime, ou l'icône correspondante) Pour quitter le runtime, si l'on n'a pas prévu de bouton spécifique pour cela (normalement, la supervision n'a pas à être interrompue), appuyez Alt F4 ou la touche windows et fermez la fenêtre. Ou alors, dans « paramétrage du pupitre » décochez « plein écran ».

exemple de vue pour nos deux chariots

The screenshot displays the WinCC Runtime environment. At the top, a file explorer shows the project structure. The main window, titled 'WinCC-Runtime -', shows a graphical interface for two cranes, 'Chariot A' and 'Chariot B'. Each crane has a corresponding set of control buttons: 'Repos', 'Chargt', 'Attente', 'Travail', and 'Retour'. Below these buttons are numerical displays for 'valeur Cptra' (11,00) and 'valeur CptraB' (70,00). A central display shows 'commande / valeur Tempo' with values 0 and 0. The background features a blue circular graphic. The bottom of the window shows the status bar with 'vue-chariots.pdl' and 'MAJ NUM'.

P Trau