

I.U.T. de Limoges
Dpt G.E.I.I.

2^{ème} année
Option R.L.I.

AUTOMATISME

TP3-1

Langage FST pour IPC HC16 :

OM20+IO40

TP4 : Langage FST

But : Apprendre le langage FST des automates IPC HC16 en s'appuyant sur les cartes d'Entrées/Sorties TOR , module appelé OM20.

1) LES DIFFERENTES PHASES DE CONSTRUCTION D'UN PROJET.

1.1) Fermer l'ancien projet : (si nécessaire)

Aller dans le menu Project| Close

1.2) Créer un nouvelle application :

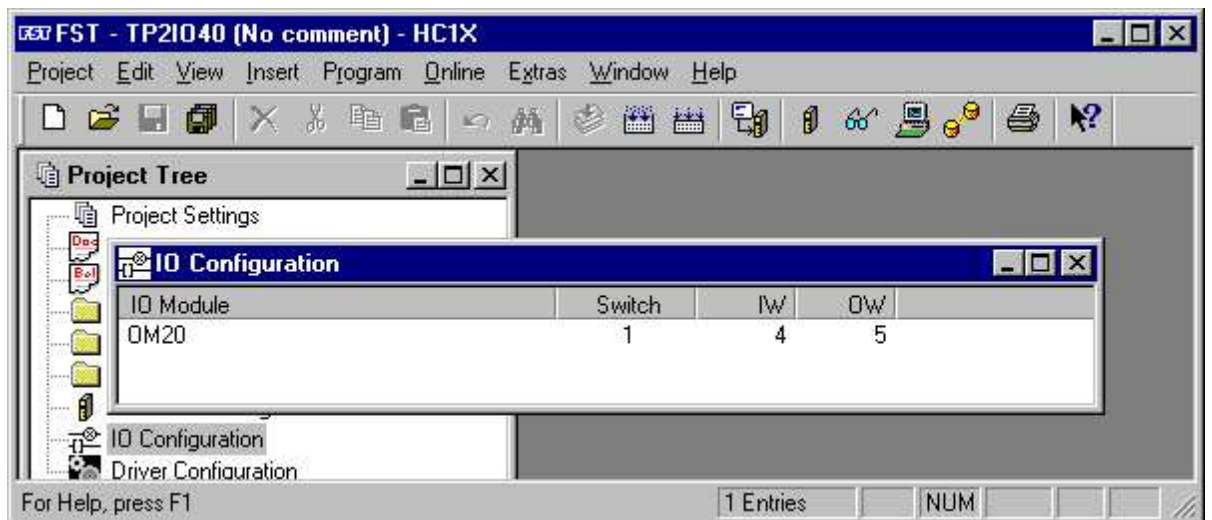
Aller dans le menu *Project | New*

Donner un nom de projet en 8 lettres maximun.

1.3) Mettre les Drivers des cartes insérées dans le BusBoard

Dans notre cas, on va insérer le driver de la carte **OM20** :

Aller dans l'arborescence *I/O Configuration* et double cliquer. La fenêtre *I/O Configuration* s'ouvre. Redouble Cliquer dans la fenêtre et vous pouvez choisir la carte d'Entrées/Sorties que vous voulez. Vous sélectionnez donc la carte **OM20**. Donner le numéro du switch (1 par défaut) et les numéros i et j des mots mémoires FST **IW_i** et **OW_j** où seront recopiés les valeurs des Entrées et Sorties TOR de la carte OM20.(Ex 4 et 5)



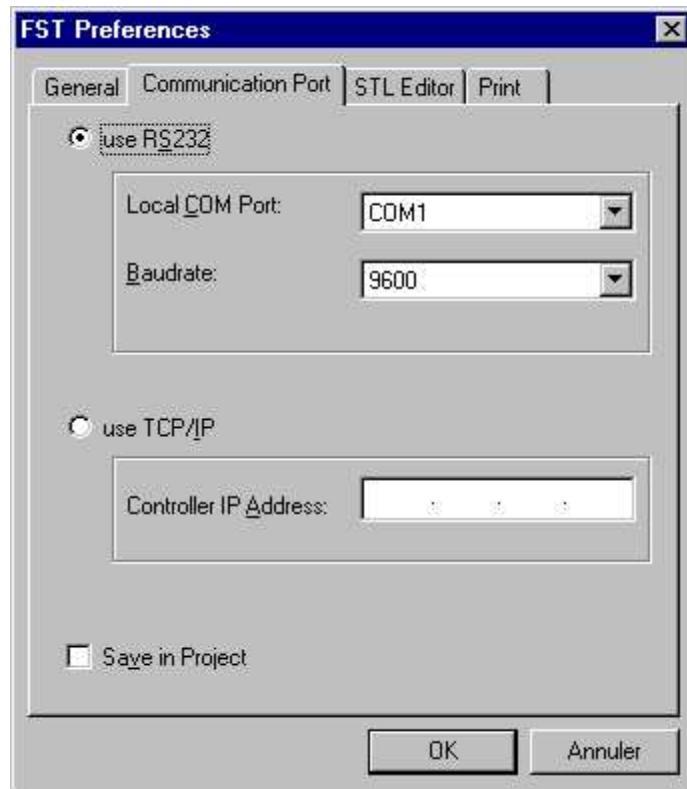
Fermer la fenêtre.

1.4) Tester la carte OM20 (sans aucun programme)

1.4.1)Après vérification de la liaison IPC-PC par le menu :

Extra|Preference|Communication Port

Vous devez établir une communication entre le PC et l'IPC par le COM1 avec une vitesse de 9600 bauds



1.4.2) Compiler le projet

Make Project (F7)

Rq : il n'y a pas de programme, on compile uniquement le driver de la carte OM20, d'où le warning qui apparaît.

1.4.3)Charger le projet afin que le driver OM20 soit dans l'IPC :

Download Project(F5)

Ainsi le projet est placé sur l'unité C:\ de l'IPC qui est celle où il y a le plus de place. On chargera les drivers sur l'unité A:\, pour qu'ils soient à accès rapide.

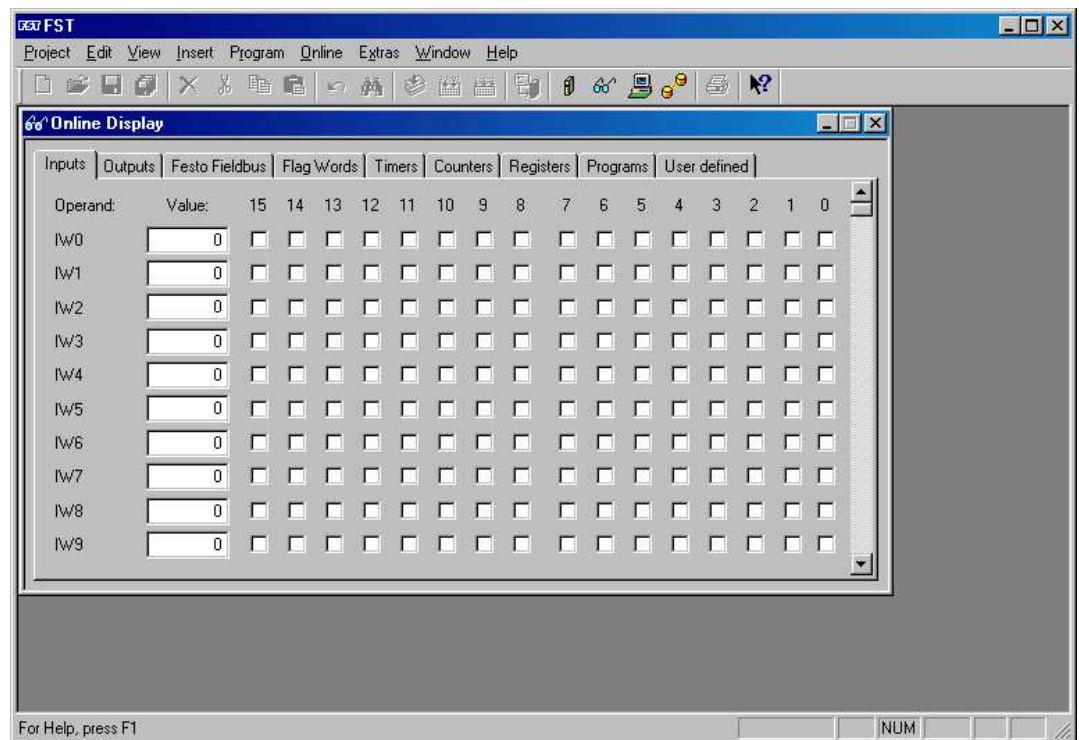
1.4.4) Rebooter

Une fois la transmission finie, L'IPC reboote et l'indique par un bip sonore.

Rq : Le reboot n'est automatique qu'au premier chargement du projet. Donc par la suite, après un nouveau chargement du projet, on devra rebooter manuellement l'IPC en appuyant sur le bouton de reset vert.

1.4.5) Test d'une Entrée

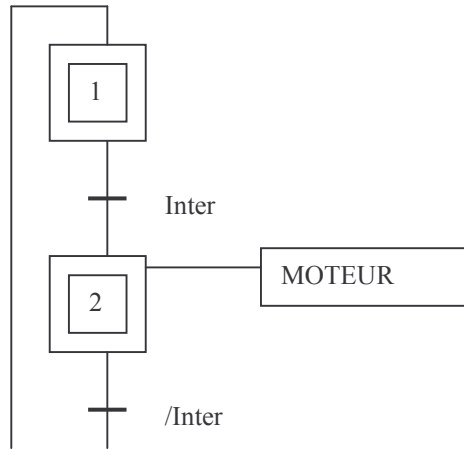
On teste le changement d'état d'une entrée en mettant l'IPC en **OnLine** : *Online | Online Display*. L'écran suivant apparaît, sinon la liaison est mal configurée.



Vérifier alors le bon fonctionnement de la carte : le mot **IW4** doit se modifier lors du changement d'état d'un interrupteur mis à une entrée de OM20.

1.5) Programmer l'exercice

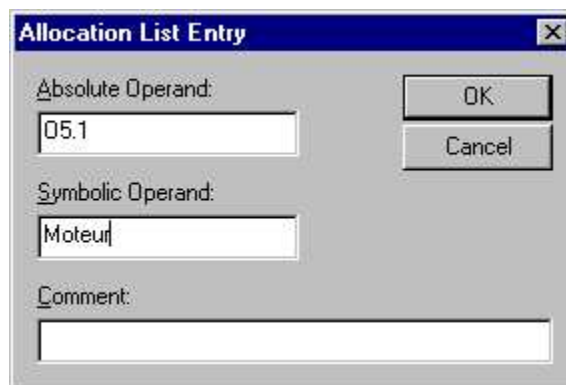
Ecrire en STL par l'intermédiaire de l'éditeur *STL Editor* l'exercice simple ci-dessous :



Déclarer les variables utilisées dans l'arborescence **Allocation List** :

Double cliquer sur l'arborescence Allocation List, puis dans la fenêtre sélectionner *Insert Operand* avec le clic droit de la souris.

Définir Moteur en O5.1



Puis Inter en I4.0

Ecrire le Grafcet précédent en STL dans le programme **P0**, qui est le programme immédiatement lancé après reboote de l'IPC.

Compiler et charger le projet.

ATTENTION : le programme **P0** doit être coché dans l'arborescence pour qu'il soit pris en compte.

1.6) Visualisation Online de l'application.

Vérifier le bon fonctionnement du programme en visualisant l'exécution. Pour cela, dans la fenêtre du programme P0 faire clic droit et sélectionner Online. On voit ainsi le STEP où est bloqué le programme et les valeurs des variables.

2) Application

Réaliser le TD : Régulation d'un niveau dans un réservoir dont l'énoncé est donné page suivante, avec son grafcet donné ci-dessous.

Les capteurs de niveau de la cuve **h**, **m**, **b** seront remplacés par un capteur analogique appelé **niveau** qui donnera le niveau de la cuve sous la forme d'une tension comprise entre 0 et 10v :

0v : cuve vide

10v : cuve pleine

On utilisera pour cela une entrée de la carte IO40

h correspondra à une tension de 8v

m correspondra à une tension de 5v

b correspondra à une tension de 2v

Les pompes **Pompe1**, **Pompe2**, et **Pompe3** seront des sorties TOR de la carte OM20. Ajouter un inter **Marche** et un inter **Stop**.