



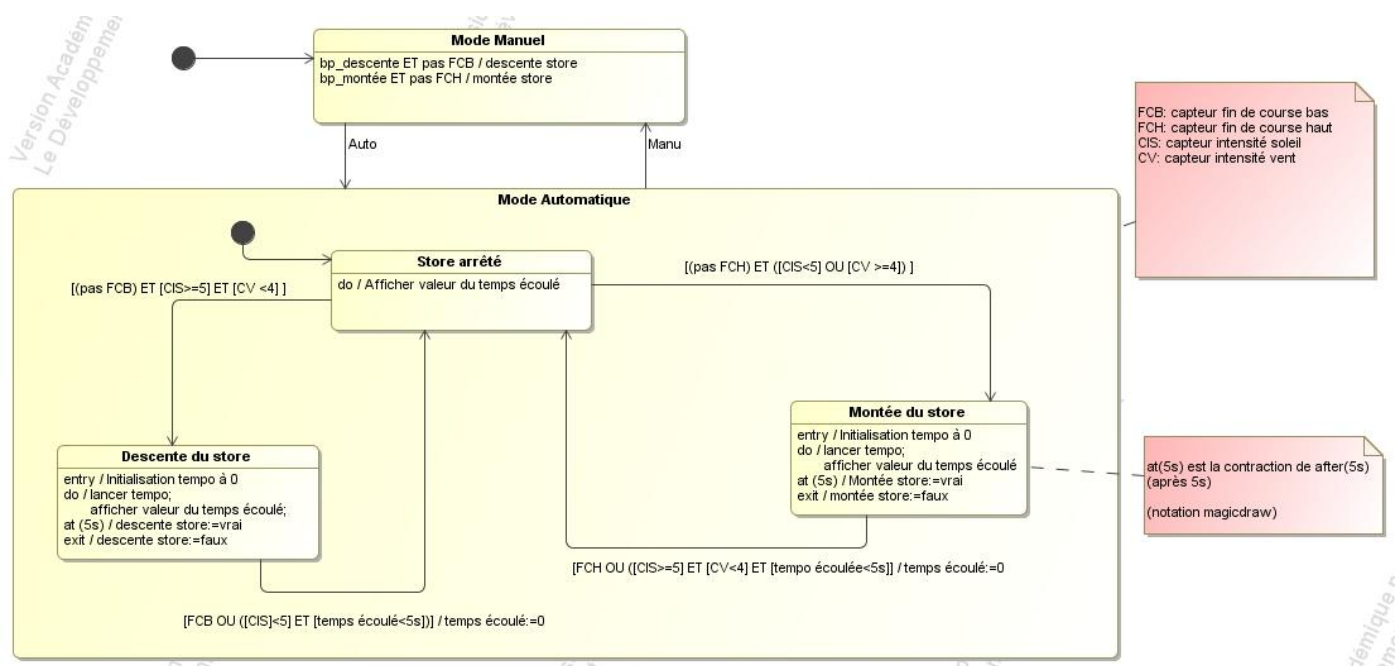
1) OBJECTIFS

- ✚ Spécifier deux modes de marche de fonctionnement d'un système automatisé par un diagramme états transitions de la norme Sysml.
- ✚ Utiliser la notion « d'état composite » d'un point de vue Sysml et l'implémenter sous Labview.
- ✚ Introduire la notion de temps (temporisation) d'un point de vue Sysml et l'implémenter sous Labview.

2) SYSTEME SUPPORT DE CE TUTORIEL : STORE TYPE SOMFY (Que tout le monde connaît)



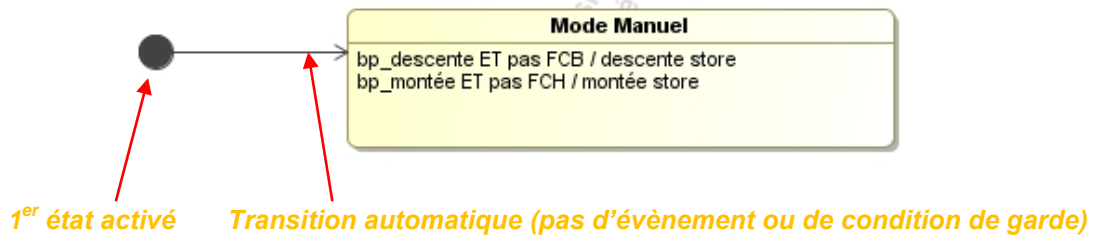
3) DIAGRAMME ETATS-TRANSITIONS PROPOSE





4) DECODAGE, INTERPRETATION DE CE DIAGRAMME

4.1) Etat initial général



Donc à la mise en « RUN » l'état initial général atteint sera le « Mode Manuel »

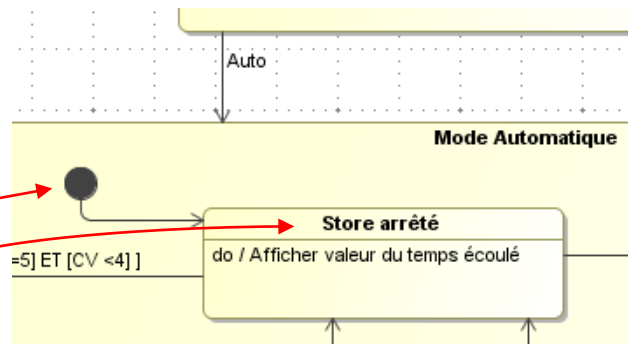
4.2) Etat initial du « Mode Automatique »

A l'appui sur le bouton « Auto » le système passe de l'état « Mode Manuel » à l'état « Mode Automatique ».

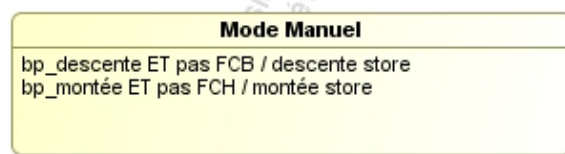
C'est un état composite car état composé d'états.

D'après le rond noir (état initial)

Le 1^{er} état atteint sera « Store arrêté »



4.3) Actions événementielles de l'état « Mode Manuel »



- ✚ Si l'opérateur appuie sur le bouton « bp_descente » et que le store n'est pas en position basse alors l'action « descente store » s'effectue.
- ✚ Si l'opérateur appuie sur le bouton « bp_montée » et que le store n'est pas en position haute alors l'action « montée store » s'effectue.

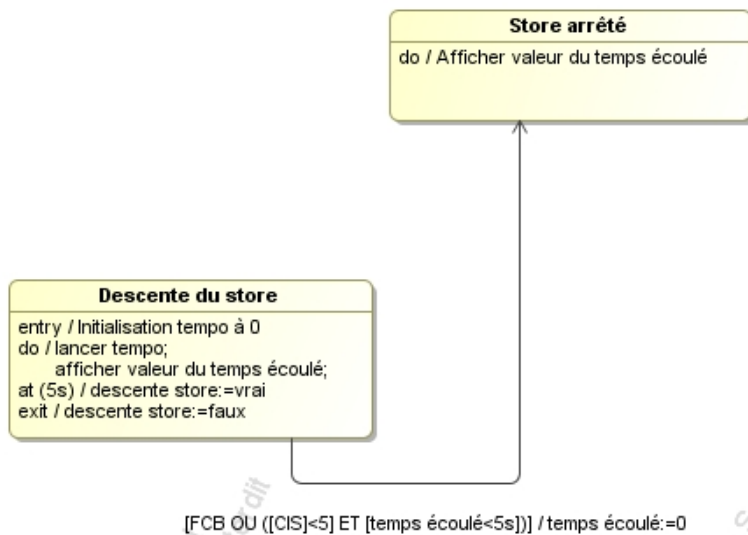


4.4) Séquencement des actions spécifiées dans l'état « Descente store »



1. Mise à 0 de la tempo (initialisation) à l'entrée (entry) dans l'état.
2. Lancement de la tempo et affichage du temps écoulé
3. Si « temps écoulé » arrivé à 5 secondes alors exécution de l'action « descente store »
4. A la sortie (exit) de l'état (transition suivante tirée) la tempo ainsi que la « descente store » s'arrêtent.

4.5) Transition entre l'état « Descente store » (considérée comme actif) et l'état « Store arrêté »



1) Si le capteur fin de course bas (FCB) est atteint alors le système passe dans l'état « Store arrêté »

OU

2) Si l'intensité du soleil (CIS) a baissé sous le niveau de 5 (nuage) avant que le temps écoulé soit arrivé à 5s (dans la réalité ce serait plusieurs mn) alors le store passe dans l'état « store arrêté ».

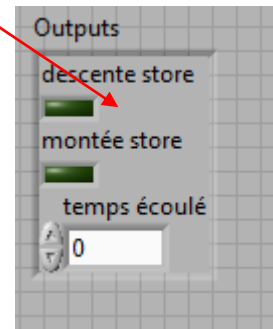
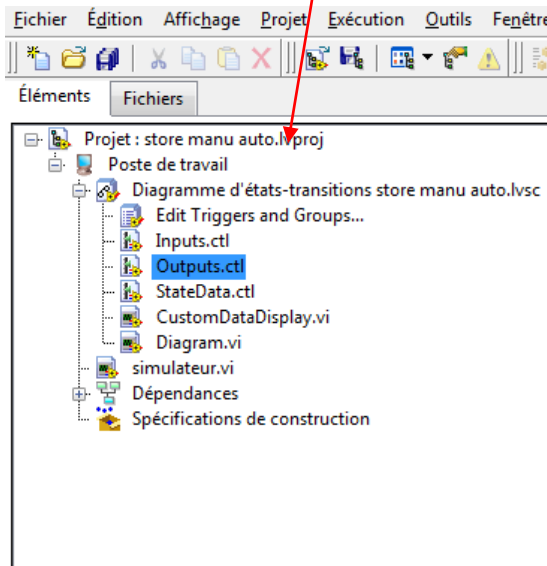
De ce fait l'action événementielle »

at(5s)/descente store :=vrai n'aura pas été réalisée.



5) ELEMENTS DE PROGRAMMATION EN STATECHART SOUS LABVIEW

5.1) Création du projet, déclaration des variables d'entrées et sorties : voir tutoriel niveau1

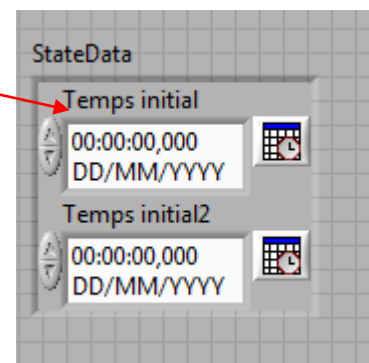
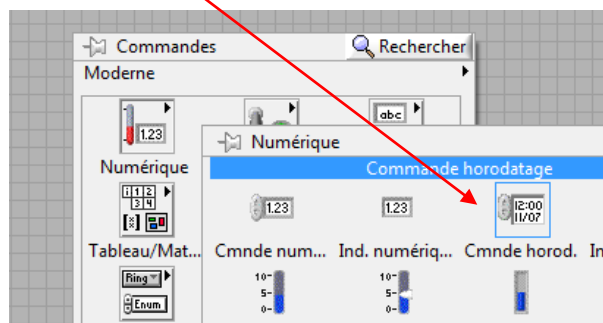


5.2) Création de variables internes : « states data »

Ce sont des variables utiles pour des initialisations ou des résultats de calculs,....

Ici on a besoin de deux variables type « horodatage » (exemple : temps écoulé entre deux dates) pour initialiser deux temporisations.

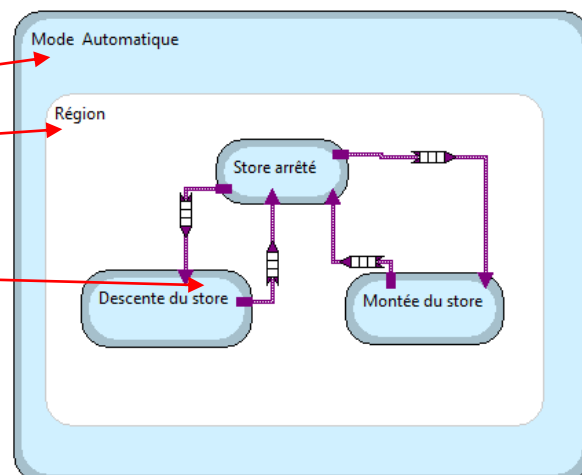
Clic sur palette « numérique » → Cmnde horod.



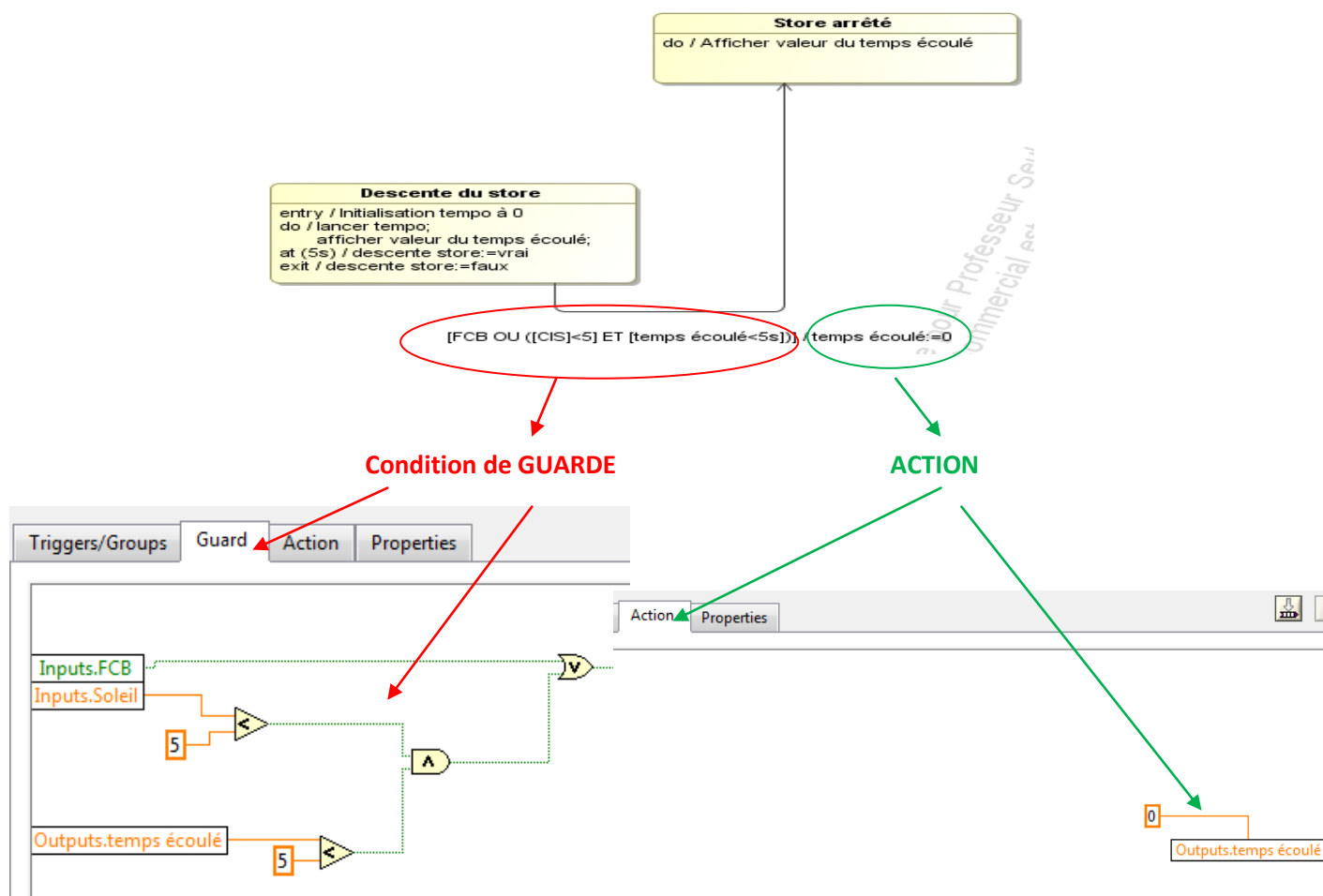


5.3) Création de l'état composite du diagramme états transitions

- Créer un grand état nommé : « Mode Automatique »
- Créer une région dans cet état :
(outil région de la palette statechart)
- Dans cette région créer le diagramme interne comme
vu dans le tutoriel niveau 1

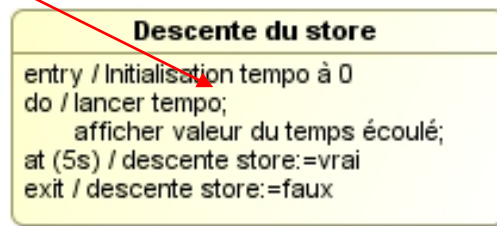


5.4) Exemple d'implémentation d'une transition type celle représentée ci dessous

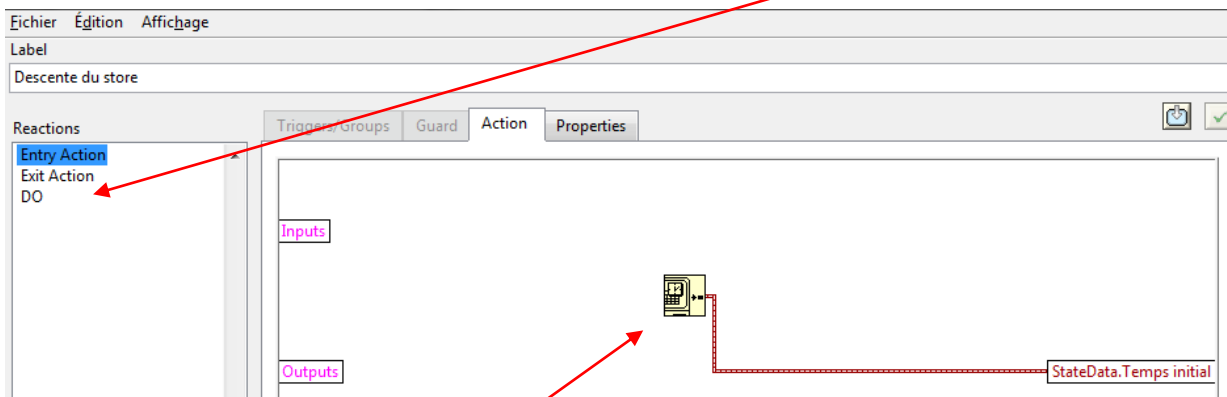




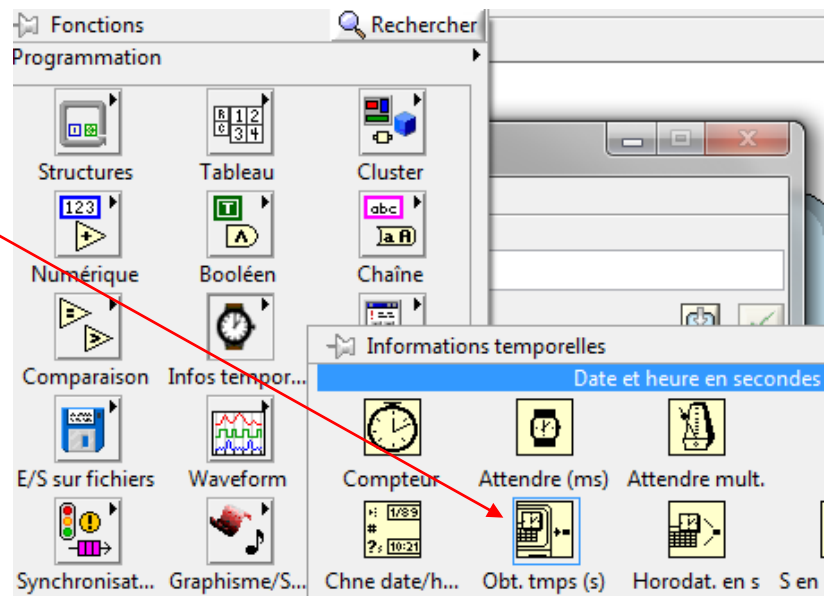
5.5) Implémentation de la temporisation figurant dans l'état : « Descente Store »



5.5.1) L'initialisation de la tempo se réalise sur l'entrée de l'état : « Entry Action »



Cette fonction donne le nombre de secondes écoulées depuis le 1^{er} janvier 1904 à 0 heure.



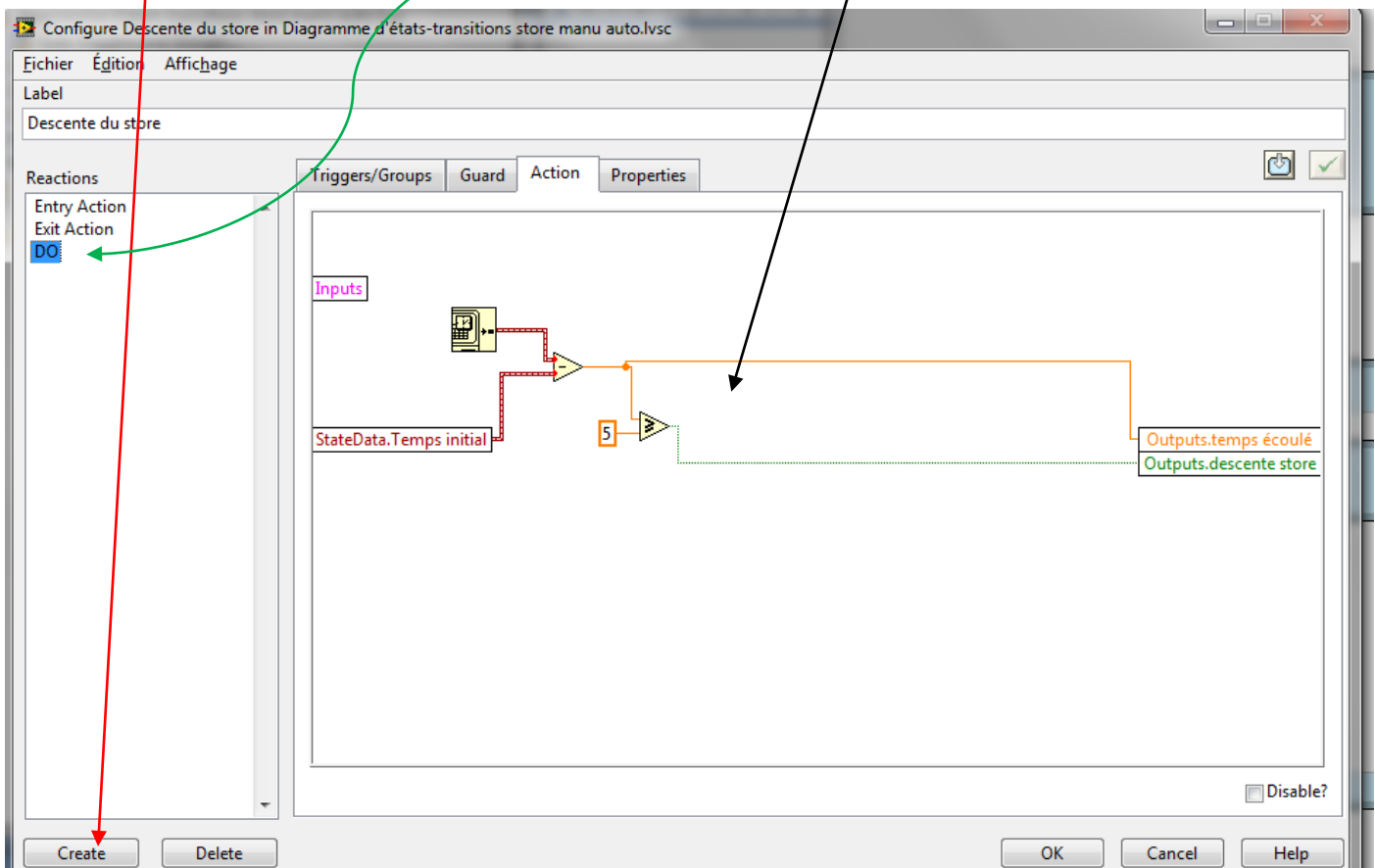


5.5.2) Lancement de l'activé « lancer la tempo » qui suit le mot clé « do/ »

« Do/ lancer la tempo »

Il faut créer une réaction que l'on appellera « do »

○ Puis tracer le diagramme ci-dessous



Explications concernant le fonctionnement de cette partie de diagramme

- La soustraction fait la différence entre le temps donné par l'horodateur de cette fenêtre et le temps initialisé en entrée d'état.
- Le résultat donne bien le nombre de secondes écoulées depuis l'entrée dans cet état.
- Dès que le résultat de la soustraction est supérieur ou égal à 5s, la sortie « descente store » est mise à 1.

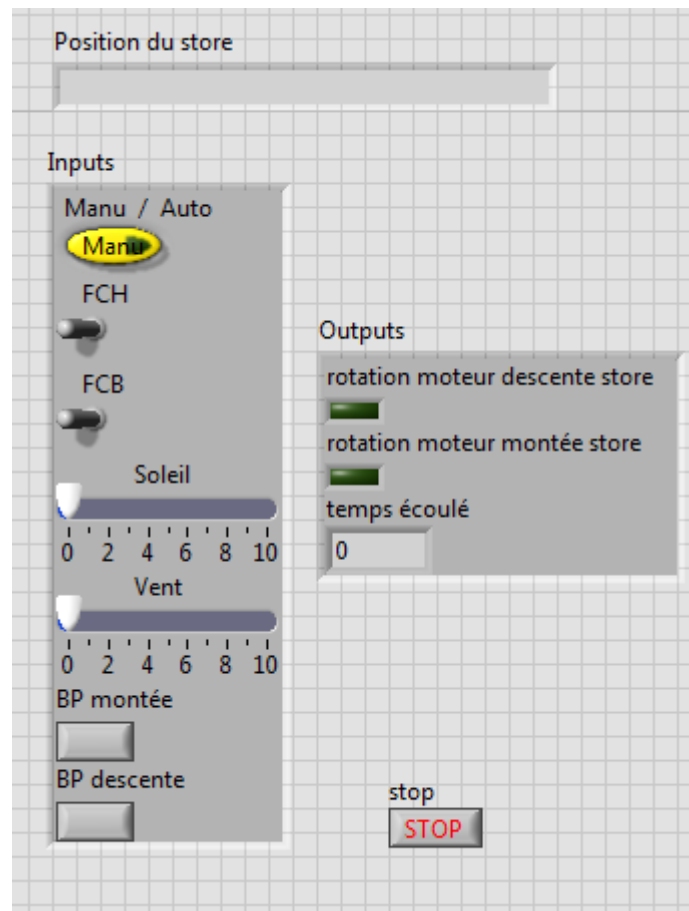


6) REALISATION DU VI APPELANT DE SIMULATION

- ✚ Reprendre la même procédure décrite dans le tutoriel niveau 1
- ✚ Sur le Poste de travail, avec un clic droit choisir Nouveau → VI. Enregistrer ce vi sous « »

Réalisation de la face avant (même procédure décrite dans tutoriel niveau 1):

Vous pouvez établir une face avant suivant l'exemple ci-dessous



- Mise en place des entrées
(ici j'ai mis un seul bouton « Manu/Auto » : état 0 = Manu (jaune) = complément de « auto » dans le diagramme états transitions)
état 1 = Auto (vert)

Les 2 boutons poussoirs sont configurés en mode : « **commutation jusqu'au relâchement** »

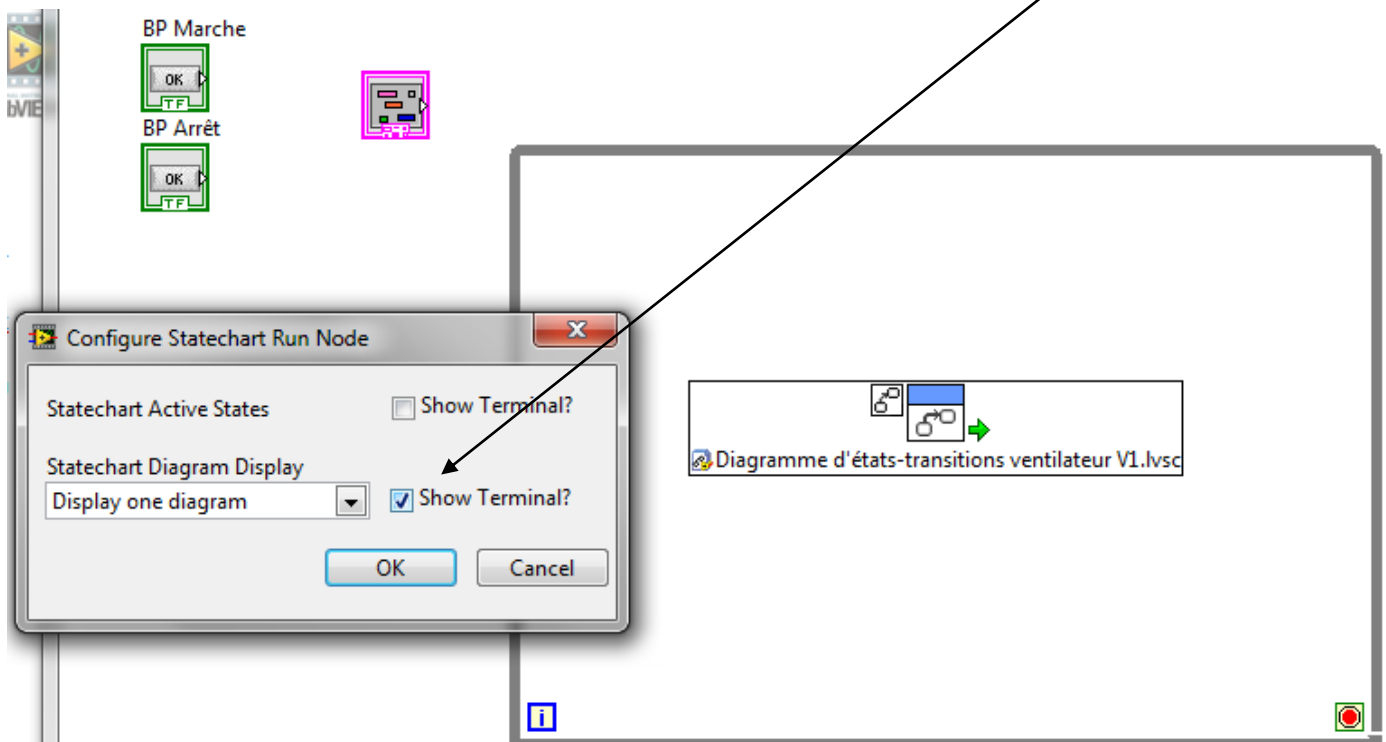
- Mise en place des sorties (2 booléens et un indicateur numérique).
- Le bouton « stop » est mis automatiquement lors de la création de la boucle while au niveau du diagramme.



Réalisation du diagramme

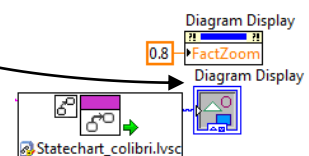
Voir tutoriel niveau 1

- Sur le diagramme du VI « **Simulateur Store...** » : Créer une boucle « **While** » dans Programmation → Structures.
- Dans cette boucle **While**, ajouter la fonction **"Run statechart"** dans **StateChart** → **StateChart Communication**.
- Faite un clic droit sur cette fonction et choisir Link to StateChart et avec l'icône ouvrir, indiquer le chemin où est stocké le "Diagramme d'états-transitions de cette application " puis valider par OK.
- Faite un clic droit sur cette fonction et choisir « **Configure...** » puis cocher « **Show terminal?** »



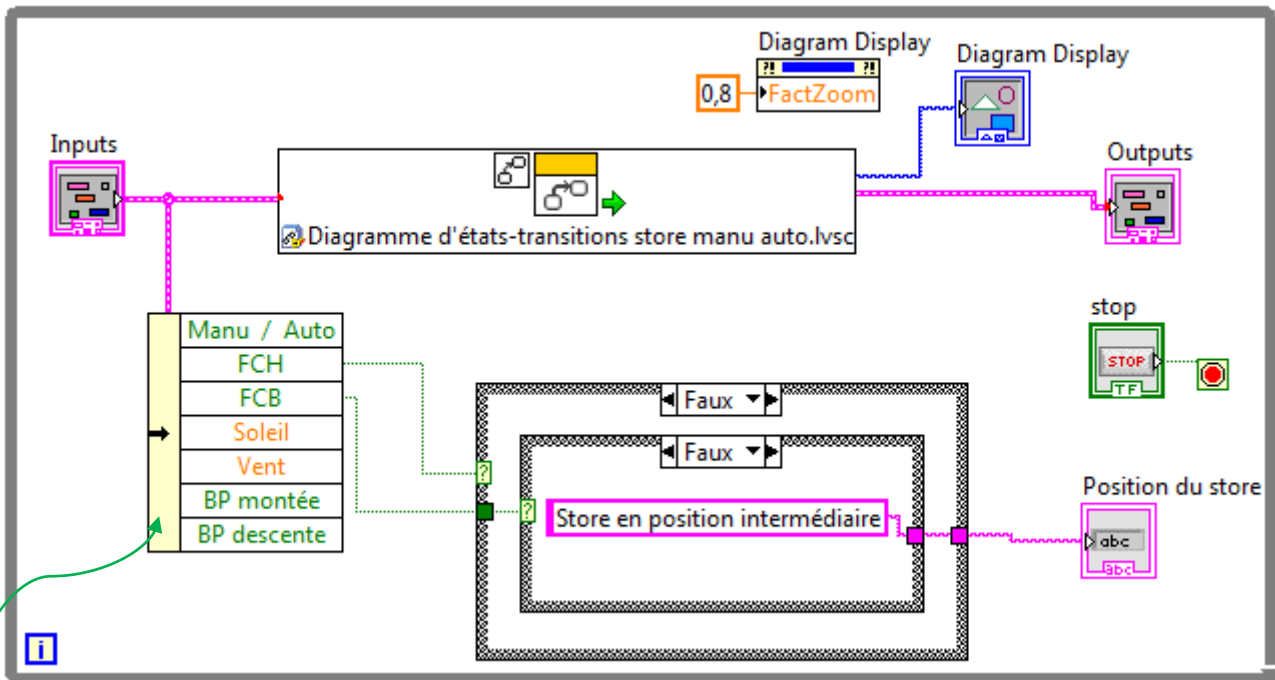
- Sur la sortie « **Display Terminal** », Créer → **Un indicateur**.
- Sur cet indicateur faire **Créer** → **Nœud de propriété** → **Facteur de zoom**.

Faite un clic droit sur ce nœud et **changer en écriture** puis créer lui une **constante égale à 0.8** en faisant clic droit **Créer** → **Constante**.





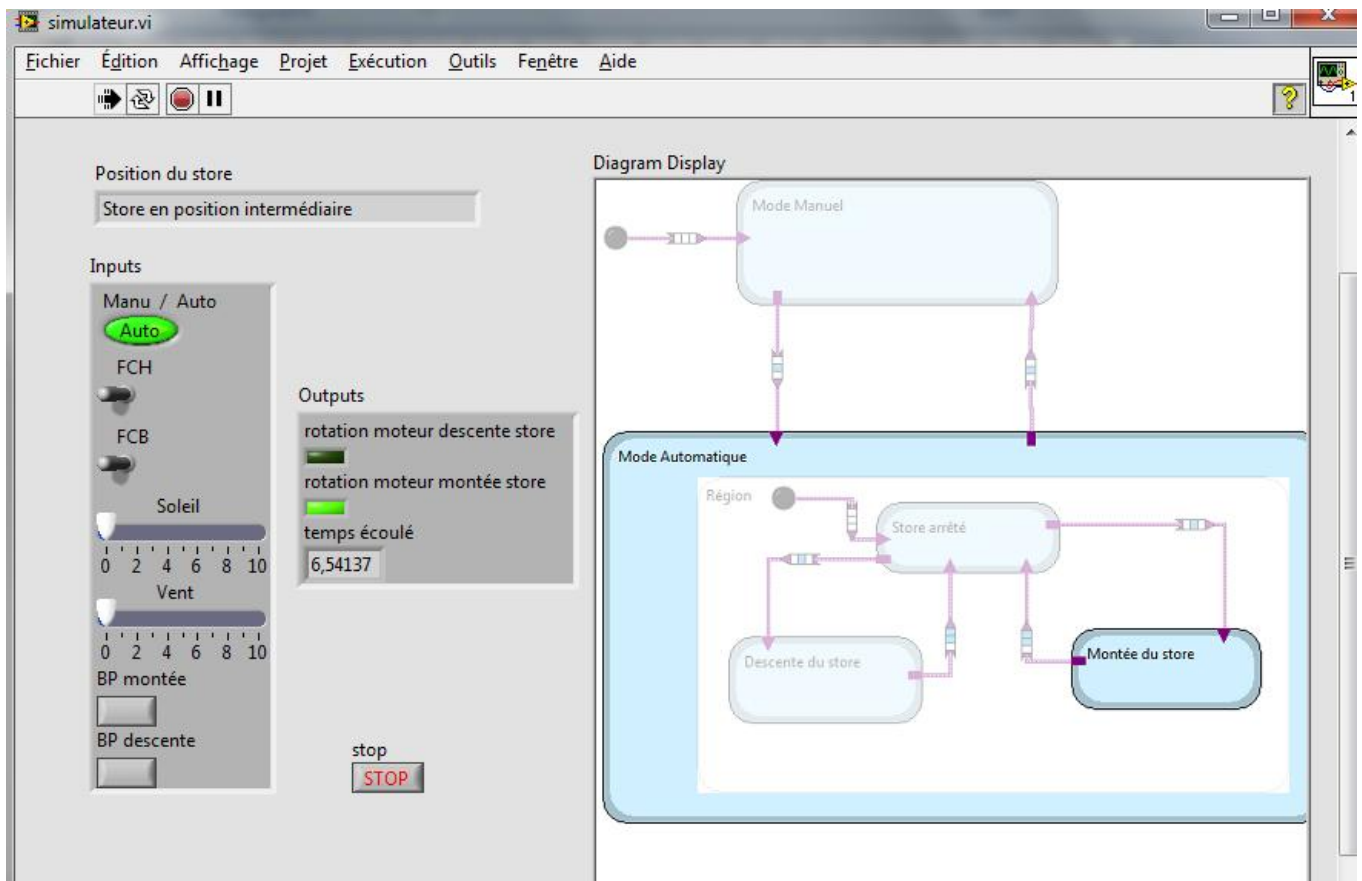
Terminer le diagramme pour obtenir ceci (enregistrer tout depuis la fenêtre projet)



Attention : j'ai ajouté l'affichage de la position du store sous forme de texte d'où les deux structures conditions imbriquées (avec des transferts de chaînes de caractères), mais pas indispensables pour notre étude. On aurait pu mettre ces infos directement dans le diagramme en activités sur états...

Ceci est une « fonction de désassemblage de cluster par nom » afin de d'isoler chaque variable.

Faire apparaître la fenêtre de la face avant, mettre en RUN et observer le bon fonctionnement



Exemple du Store en mode Automatique dans l'état « Montée du store »