



Sciences et technologies

de l'Industrie et du développement durable

ET 24 : Modèle de comportement d'un système

Introduction aux notions de modélisation au comportement séquentiel avec Labview.

Sciences et Technologies de l'Industrie et du Développement Durable <i>Formation des enseignants</i>	
parcours : ET24	Modèle de comportement d'un système
Durée : 3 h.	
Objectif : Etre capable de créer un modèle de comportement séquentiel à l'aide de diagrammes d'état.	
Pré-requis : Aucun.	
Bases théoriques : Les bases de Labview.	
Outil : Labview	
Support :	
Modalités : Activité sous forme de TD	
Synthèse et validation : Être capable de recréer en autonomie les modèles proposés.	



Sciences et technologies

de l'Industrie et du développement durable

Sommaire

1	Présentation des diagrammes d'état et du sujet d'étude.....	3
1.1	<i>Les diagrammes d'état.....</i>	3
1.2	<i>Sujet d'étude.....</i>	3
2	Démarche de création de diagramme d'état sous Labview	4
2.1	<i>Création d'un projet.....</i>	4
2.2	<i>Création du diagramme d'état.....</i>	5
2.2.1	<i>Définition des états.....</i>	5
2.2.2	<i>Définition des entrées.....</i>	6
2.2.3	<i>Définition des sorties.....</i>	6
2.2.4	<i>Codage des actions à effectuer dans les états.....</i>	7
2.2.5	<i>Codage des transitions.....</i>	8
2.3	<i>Programmation du VI de contrôle.....</i>	10
2.3.1	<i>Création du VI.....</i>	10
2.3.2	<i>Mise en place d'une boucle.....</i>	11
2.3.3	<i>Câblage des entrées.....</i>	11
2.3.4	<i>Câblage des sorties.....</i>	12
2.4	<i>Visualisation de l'évolution du Diagramme d'état.....</i>	13
2.5	<i>Statechart synchrone / Statechart asynchrone.....</i>	15
3	Exercices	16
3.1	<i>Gestion de la position du store.....</i>	16
3.2	<i>Gestion de la montée et de la descente.....</i>	16
3.3	<i>Câblage du fonctionnement avec des cartes entrée/sortie.....</i>	17
4	Annexes.....	18
4.1	<i>Câblage des entrées de Statechart permettant de personnaliser la face avant du diagramme de contrôle.....</i>	18
4.2	<i>Câblage des sorties de Statechart permettant de personnaliser la face avant du diagramme de contrôle.....</i>	19



Sciences et technologies de l'Industrie et du développement durable

1 Présentation des diagrammes d'état et du sujet d'étude.

1.1 Les diagrammes d'état.

Les diagrammes d'état sont l'outil Sysml pour spécifier le fonctionnement séquentiel d'un système.

Il est usuel de dire que les Graficets sont un cas particulier des diagrammes d'état.

Il est possible d'écrire des diagrammes d'états qui n'ont pas vocation à être simulés, sous Magic Draw par exemple.

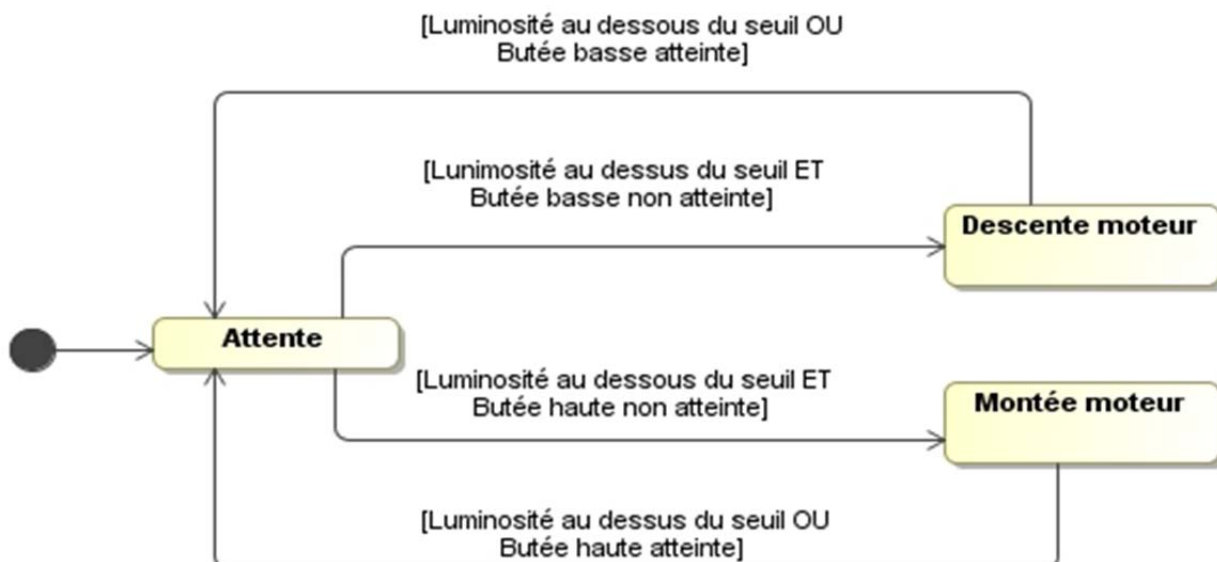
Sous Labview, les diagrammes d'état seront simulés, puis interfacés avec des cartes d'entrées/sorties.

Sous Labview, les diagrammes d'état sont nommés *Chart*, ou *State Chart*.

1.2 Sujet d'étude.

Le sujet d'étude choisi pour introduire aux diagrammes d'état sera un support que l'on peut aisément appréhender : un store automatisé, qui se remonte ou qui se rabaisse en fonction des demandes utilisateur et des conditions climatiques.

Le diagramme d'état du fonctionnement du store est décrit ci-dessous. Après une présentation de la démarche, vous allez construire ce diagramme sous Labview puis l'interfacé avec des entrées/sorties de la carte NI 6009.

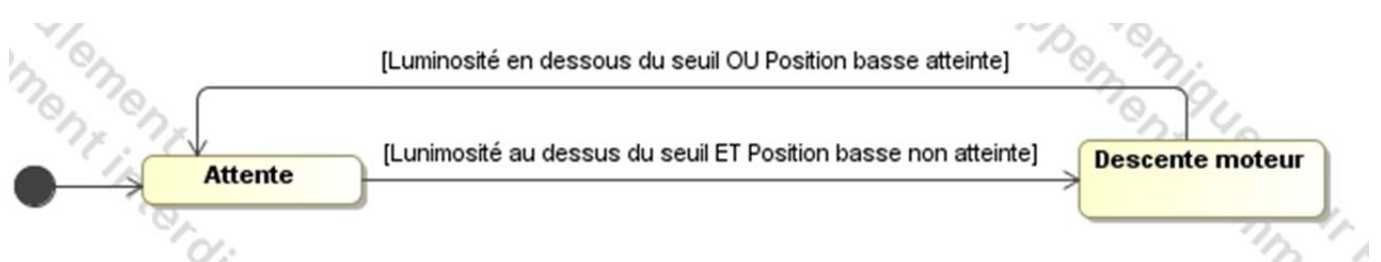




Sciences et technologies de l'Industrie et du développement durable

2 Démarche de création de diagramme d'état sous Labview

Pour introduire la démarche de modélisation sous Labview, vous allez coder une première partie du diagramme d'état :



La numérotation du chapitre « Démarche de création de diagramme d'état sous Labview » indique la démarche à suivre pour créer le Statechart. Vous remarquerez que cette démarche vous mènera jusqu'à la page... 12.

Restez motivés ! Cette démarche est toujours la même, et s'effectue rapidement avec un peu d'habitude grâce au principe de programmation graphique de Labview.

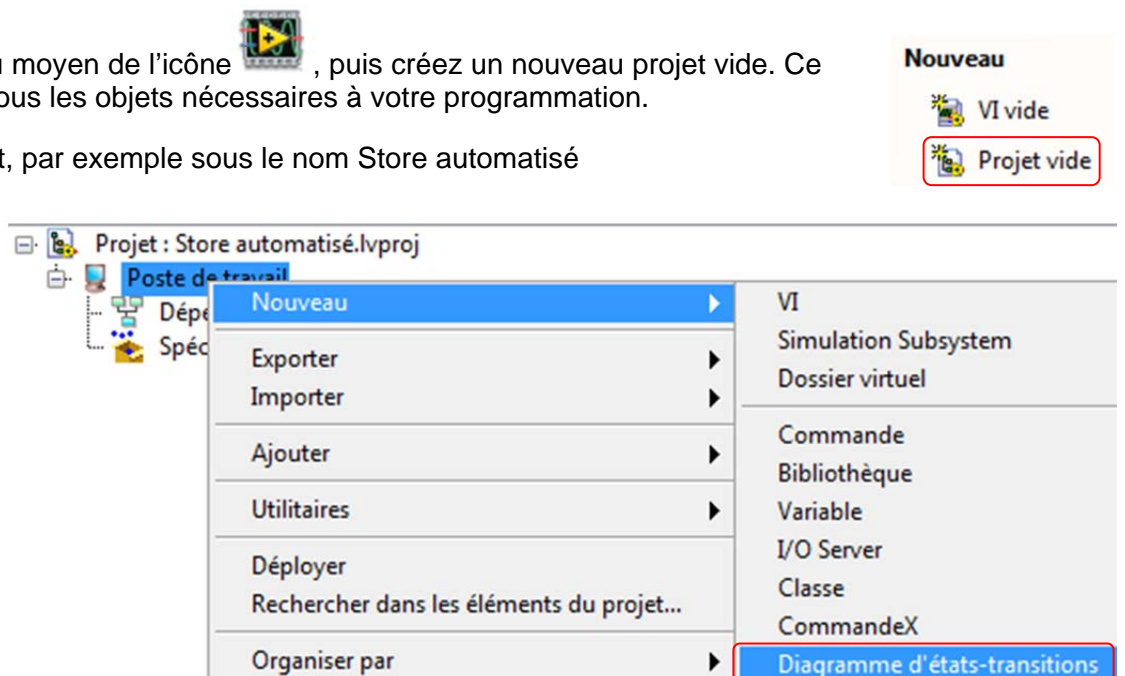
2.1 Création d'un projet.

Lancez Labview au moyen de l'icône , puis créez un nouveau projet vide. Ce projet va contenir tous les objets nécessaires à votre programmation.

Sauvez votre projet, par exemple sous le nom Store automatisé

Ajoutez un nouveau diagramme d'état transition à votre projet par un clic droit sur *Poste de travail*, puis *Diagramme d'états-transitions*.

Si Labview ne vous propose pas cette option, c'est sûrement que le module Statechart n'est pas correctement installé sur votre ordinateur.



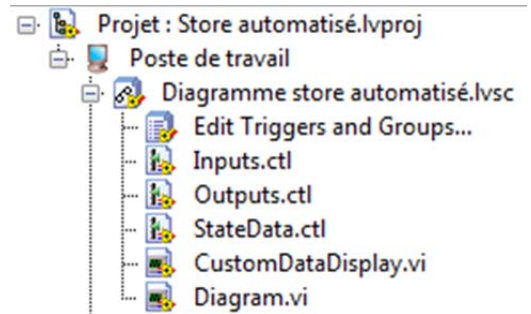


Sciences et technologies

de l'Industrie et du développement durable

Sauvegardez le diagramme sous un nom approprié. Remarquez que Labview place :

- une interface de gestion des évènements (Triggers). Vous n'allez pas les gérer dans ce TD ;
- un VI d'entrées (Input), qui servira à déclarer les entrées du diagramme d'états ;
- un VI de sorties (Output), qui servira à déclarer les sorties du diagramme d'états ;
- un VI de variables internes (Statedata) ;
- un VI CustomDataDisplay, non utilisé dans ce TD ;
- un VI Diagram, qui servira à programmer graphiquement le diagramme d'états.



2.2 Création du diagramme d'état.

La démarche de création d'un diagramme d'état sous Labview est exactement la même que celle que l'on pourrait utiliser pour les Grafcet ou pour les Diagrammes d'état sous Simulink. Seule la démarche manipulative est différente. Il vous faudra donc :

- définir les états du système
- définir les sorties du système
- définir les actions à effectuer dans les états
- définir les entrées du système
- définir les transitions entre les états.

2.2.1 Définition des états.

Comme l'indique le diagramme d'état page 4, le diagramme d'état simplifié comporte deux états : *Attente* et *Descente store*.

Pour les programmer, double cliquer sur le VI Diagram.vi. Allez dans la palette *Statechart / Statechart Development*. Placez deux états et une entrée dans le diagramme.

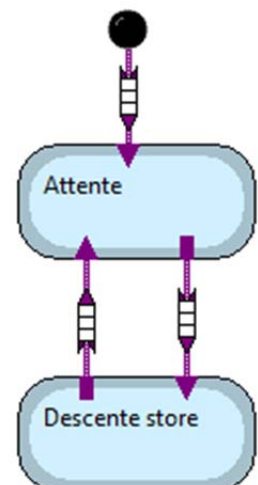


Renommez les états en cliquant sur les noms des états.

Câblez les transitions entre les états comme indiqué ci-contre. Pour câbler des transitions, il faut :

- placer la souris au-dessus de l'état de départ jusqu'à obtenir le bobine de câblage ;
- cliquer sur l'état d'arrivée.

Les états sont définis, vous pouvez fermer le VI. En enregistrant les modifications naturellement.





Sciences et technologies de l'Industrie et du développement durable

2.2.2 Définition des entrées.

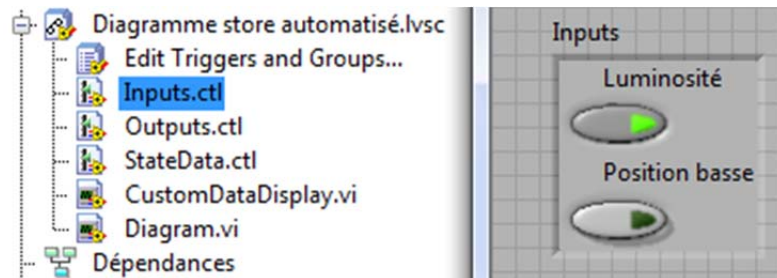
Comme l'indique le diagramme d'état page 4, le diagramme d'état comporte deux entrées :

- Luminosité
- Position basse

Pour des raisons pratiques :

- Au-dessus du seuil tolérable, la variable *Luminosité* sera vraie, en dessous du seuil tolérable, la variable *Luminosité* sera fausse.
- Si la position basse est atteinte, la variable *Position basse* sera vraie (et inversement).

Ces deux variables sont donc de type binaire. Pour les définir, double cliquer sur le VI Inputs, et placer deux boutons correctement nommés dans la zone Inputs :



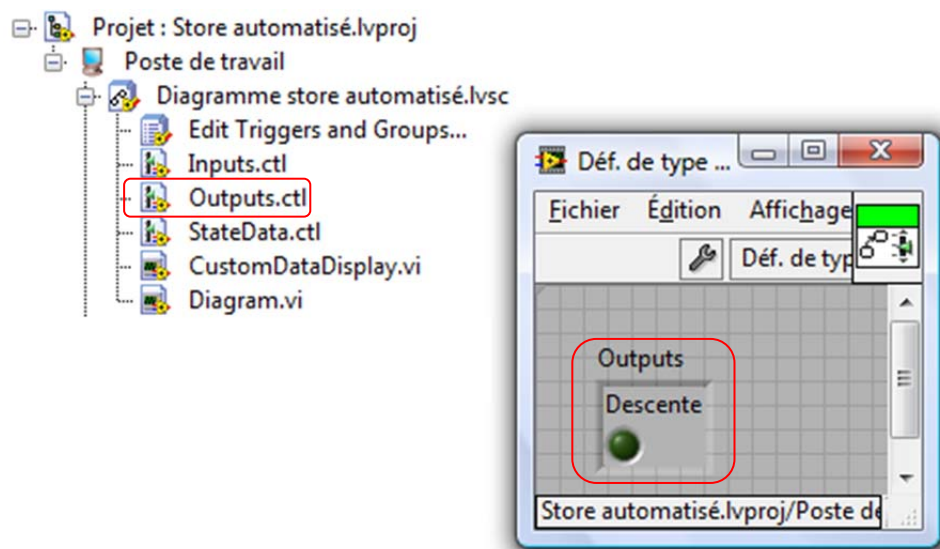
Vous pouvez fermer le VI, après l'avoir enregistré bien sûr.

2.2.3 Définition des sorties.

Le diagramme d'état ne comporte qu'une seule sortie, que l'on peut nommer *Descente*. Cette sortie est de type binaire car elle est vraie ou fausse.

Pour la définir :

- double cliquer sur la VI outputs qui se trouve dans le projet ;
- placez une sortie LED dans la fenêtre Outputs ;
- effacez la sortie qui était placée dans cette fenêtre en guise d'exemple.





Sciences et technologies de l'Industrie et du développement durable

Vous pouvez fermer le VI.

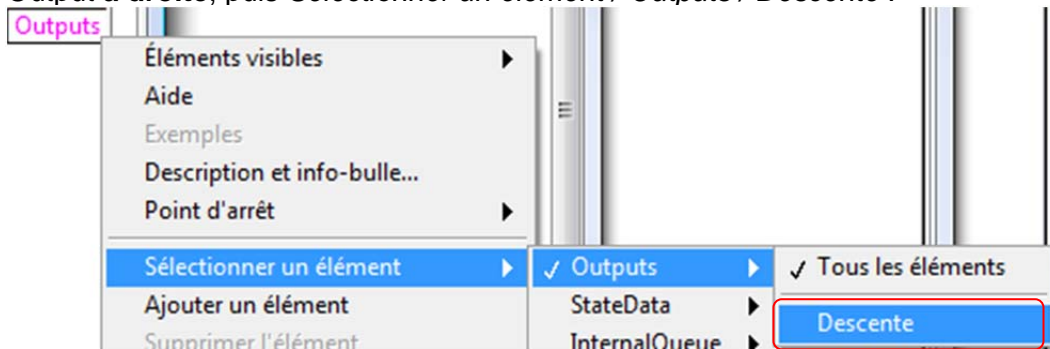
2.2.4 Codage des actions à effectuer dans les états.

Vous allez maintenant coder le comportement de la sortie. A savoir :

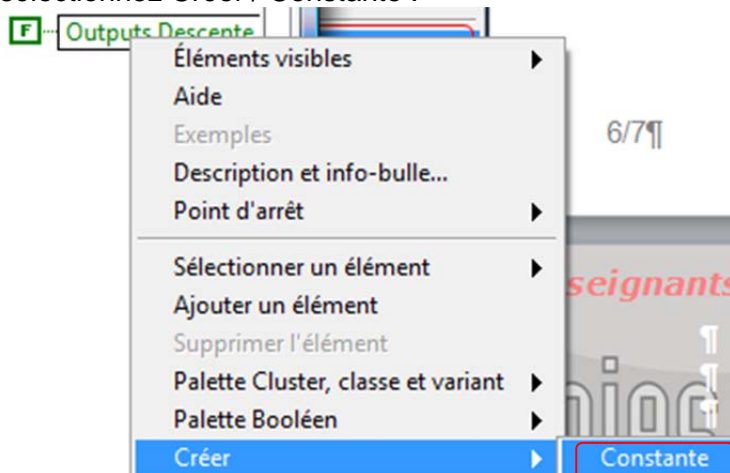
- dans l'étape de descente du store, la variable de sortie Descente doit être mise à 1 (True) ;
- dans l'étape d'attente, la variable de monté du store doit être mise à 0 (False).

Traitement de l'état « **descente** » :

- retournez dans le VI *Diagram.VI* ;
- double cliquez sur le bord de l'étape *Descente store* : la fenêtre de codage du comportement des sorties s'ouvre ;
- déclarez que vous voulez coder le comportement de la sortie Descente en faisant un clic droit sur Output à droite, puis *Sélectionner un élément / Outputs / Descente* :



- activez la variable en plaçant une constante True. Pour cela, faites un clic droit sur la sortie, puis sélectionnez *Créer / Constante* :



- si besoin, cliquez sur la constante pour faire changer son état à True :



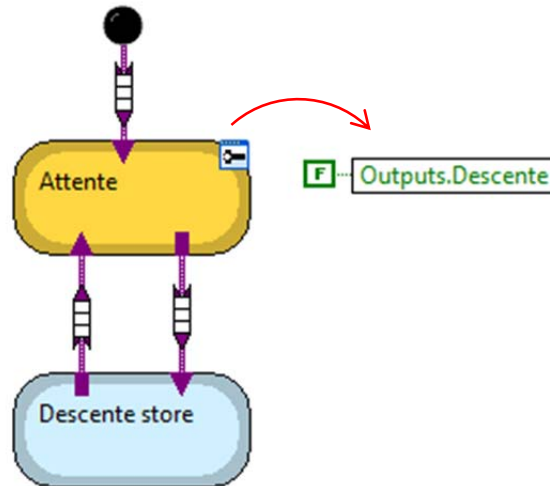
- Validez.

Cette méthode de programmation peut paraître curieuse au premier abord. Mais remarquez que c'est exactement la même démarche que celle de programmation d'un VI classique. *Il n'y a donc rien de nouveau à apprendre.*



Sciences et technologies de l'Industrie et du développement durable

De même, placez la sortie Descente à False dans l'état « **Attente** » :



Le comportement des sorties est prêt. Il faut maintenant programmer le comportement des transitions.

2.2.5 Codage des transitions.

Le codage des transitions respecte le principe de programmation de Labview.

La transition Attente => Descente store est vraie si :

- *Luminosité* est vraie ET
- *Position basse* est fausse.

La transition Descente store => Attente est vrai si :

- *Luminosité* est fausse OU
- *Position basse* est vraie.

Pour coder la transition Attente => Descente store :

- retourner dans *Diagram.vi* ;
- double cliquer sur la transition correspondante ;
- aller dans l'onglet Guard :



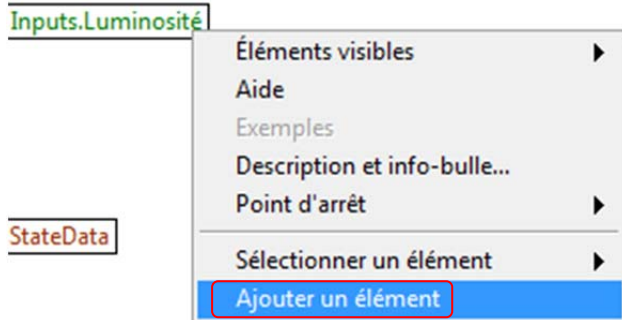
- en utilisant le même principe que celui vu dans le chapitre 2.2.4 *Codage des actions à effectuer dans les états.*, sélectionner l'entrée Luminosité :

`Inputs.Luminosité`

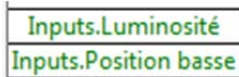


Sciences et technologies de l'Industrie et du développement durable

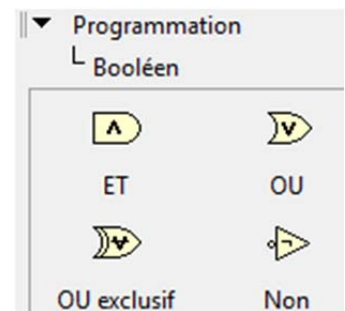
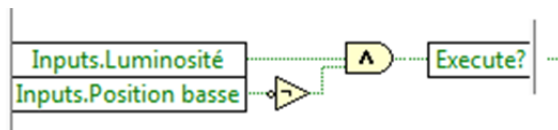
- comme il y a deux variables, en ajouter une par un clic droit, puis Ajouter un élément



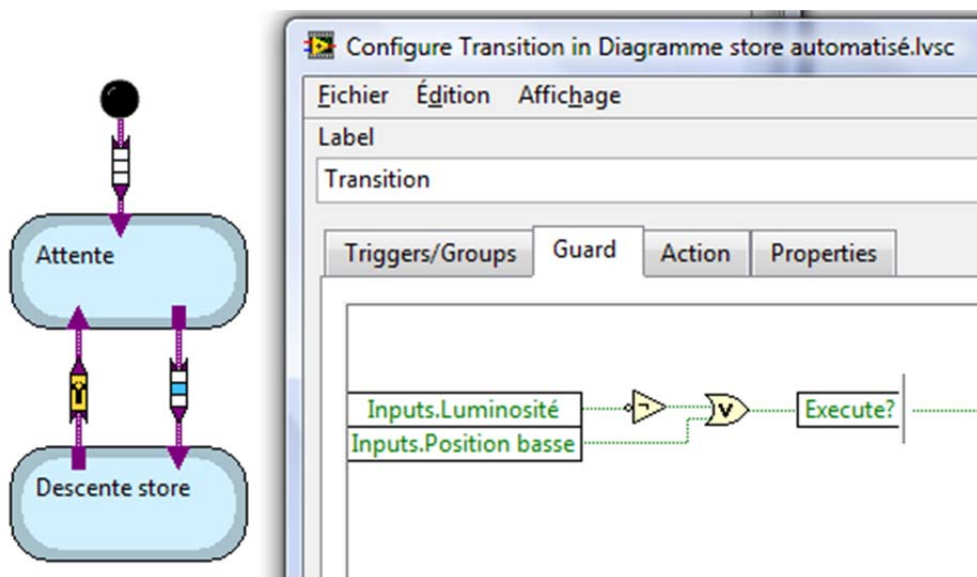
- par le même principe que celui du chapitre 2.2.4 Codage des actions à effectuer dans les états., sélectionner l'entrée Position basse :



- en utilisant la palette de programmation booléens, câblez la condition permettant d'activer la transition :



De la même manière, codez la condition permettant d'activer la transition Descente store => Attente :



Remarques :

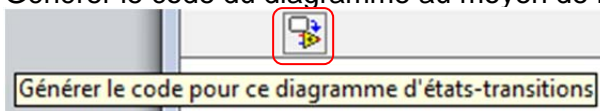
- Lorsqu'une *Guard* est programmée, le carré du milieu de la transition devient bleu.
- Il est possible de définir une action à effectuer lors du passage d'un état à un autre au moyen de l'onglet *Action*.
- L'onglet *Trigger* permet de définir des évolutions en fonctions d'évènements, non abordés ici.




Sciences et technologies de l'Industrie et du développement durable

Félicitations, vous venez de coder votre premier diagramme d'état sous Labview. Avant de l'interfacer avec des entrées/sorties réelles ou virtuelles, il ne vous reste plus qu'à

- Ouvrir le VI Diagram, au cas où vous l'avez refermé ;
- Générer le code du diagramme au moyen de l'icône :



- Vous remarquez que l'icône passe à l'état grisé .
- Fermer le diagramme.

2.3 Programmation du VI de contrôle.

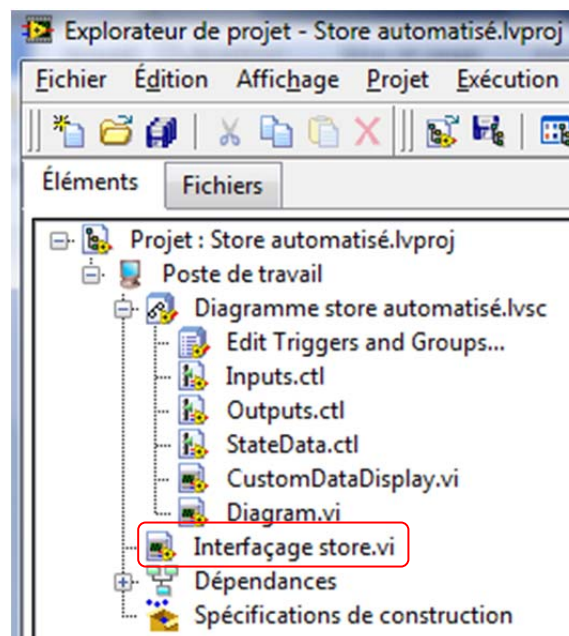
2.3.1 Création du VI.

Dans le projet, créez un nouveau VI par un clic droit sur *Poste de travail*, puis *Nouveau / VI*.



Enregistrez-le avec un nom approprié.

Remarquez son emplacement dans le projet :





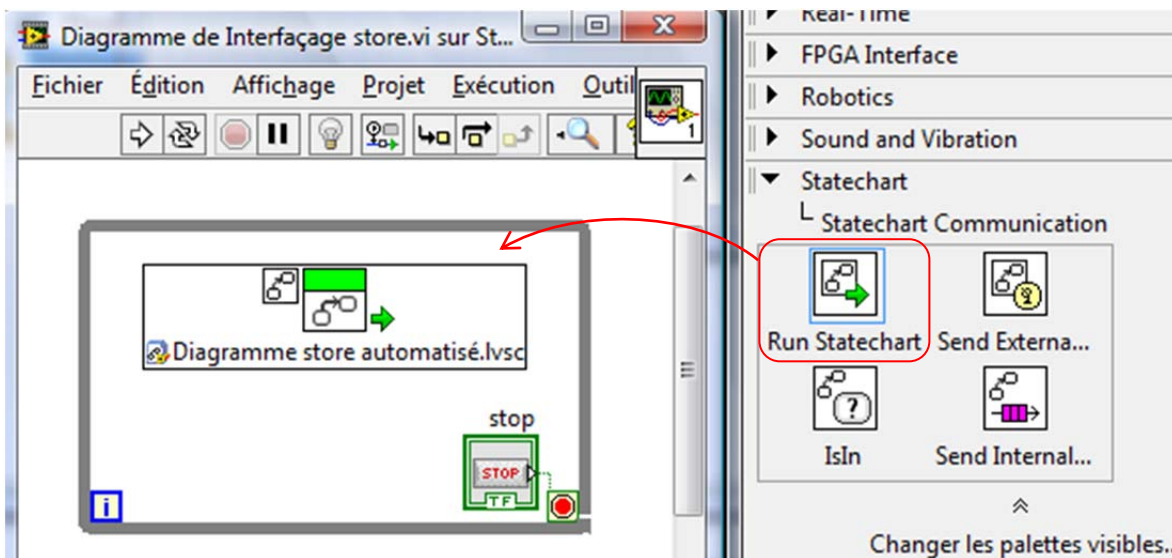
Sciences et technologies de l'Industrie et du développement durable

2.3.2 Mise en place d'une boucle.

Ouvrez le diagramme de ce VI. Placez-y une boucle WHILE. (Au passage, vous pouvez en profiter pour câbler dès maintenant la condition de fin).

Au moyen de la palette *Statechart / Statechart Communication*, placez un *Run Statechart* dans la boucle *While*.

Un message vous indique que le statechart n'est pas encore configuré : double cliquez dessus afin de rechercher le fichier .lvsc correspondant.



2.3.3 Câblage des entrées.

Labview fait communiquer les entrées/sorties du *Statechart* et celles du *VI de contrôle* au moyen de *Clusters*.

Un cluster est une ligne dans laquelle sont rangées toutes les informations à passer. Un cluster est comparable à une trame de réseau.

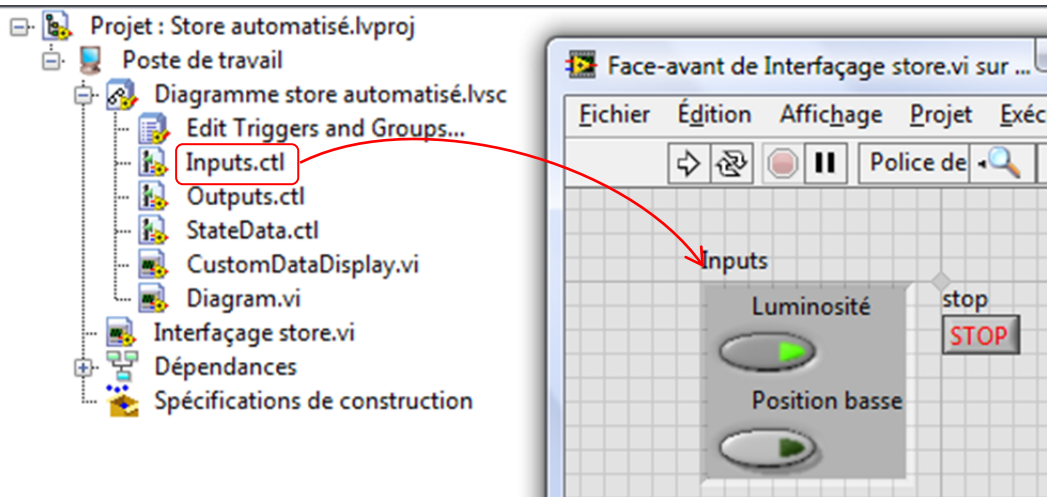
Il existe de nombreuses méthodes pour câbler des entrées au diagramme d'état dans le VI de contrôle. Nous en avons sélectionné deux :

- Une méthode 1, dite « facile » : rapide, mais qui ne permet pas de disposer les entrées exactement comme on le souhaite sur la face avant. C'est la méthode qui est présentée dans ce chapitre.
- Une méthode 2, dite « moins facile » : moins rapide (euphémisme), mais qui permet de placer les entrées comme on le souhaite sur la face avant. Cette méthode est présentée dans le chapitre 4.1 *Câblage des entrées de Statechart permettant de personnaliser la face avant du diagramme de contrôle*.

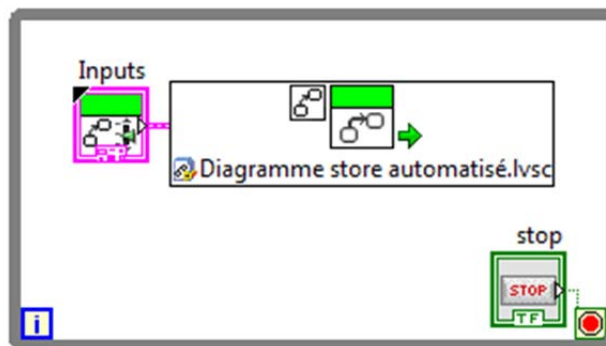


Sciences et technologies de l'Industrie et du développement durable

Pour câbler facilement les entrées, effectuez un glisser/déposer de Inputs.ctl vers la face avant du diagramme de contrôle :

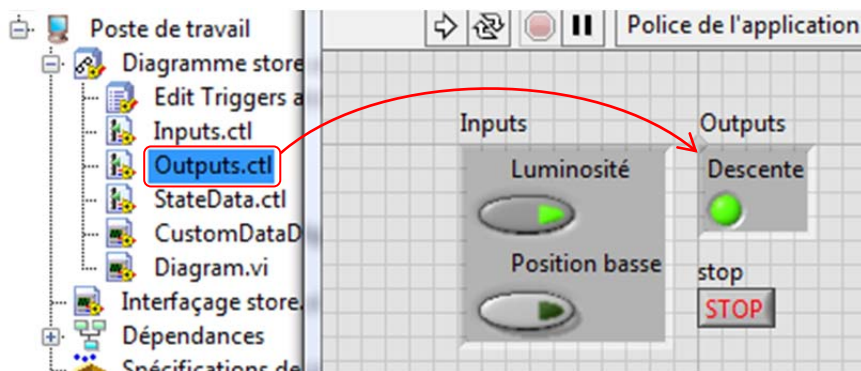


Le cluster des entrées s'est aussi inséré sur le diagramme ; il faut seulement câbler ces entrées au diagramme d'état :



2.3.4 Câblage des sorties.

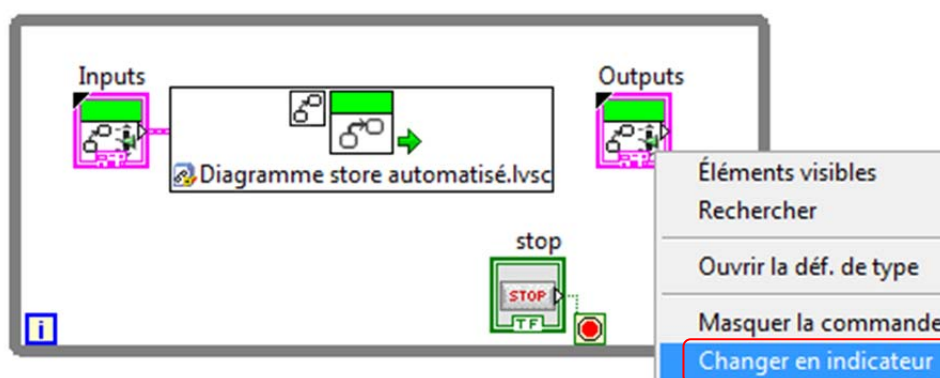
La démarche de câblage des sorties est identique :



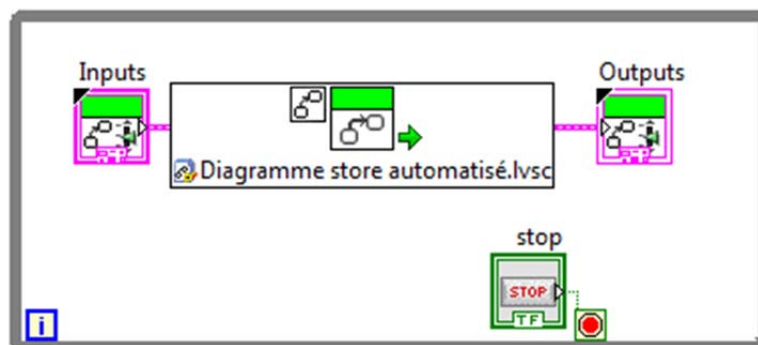


Sciences et technologies de l'Industrie et du développement durable

Sur le diagramme, il est possible que Labview insère le cluster des sorties comme des *Commandes*. Si tel est le cas, faites un clic droit sur le cluster des sorties pour le *Changer en indicateur* :



Puis câblez la sortie au diagramme d'état :



De même que pour les entrées, vous pouvez remarquer que les sorties sont confinées dans un cluster sur la face avant. Il existe une méthode pour placer autrement les sorties. Cette méthode est indiquée au chapitre 4.2 Câblage des sorties de Statechart permettant de personnaliser la face avant du diagramme de contrôle.

Testez le fonctionnement du diagramme d'état, normalement, cela fonctionne !

2.4 Visualisation de l'évolution du Diagramme d'état.

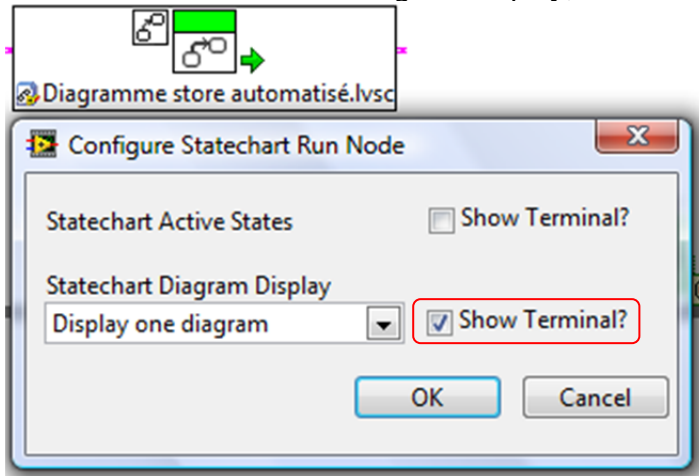
Il peut être intéressant de visualiser l'évolution du diagramme d'état sur la face avant du VI. Pour cela :

- Double cliquez sur le diagramme d'état transitions;

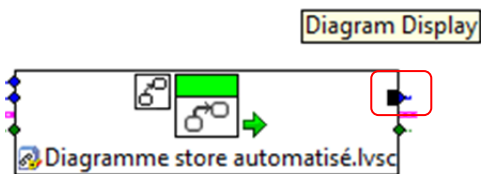


Sciences et technologies de l'Industrie et du développement durable

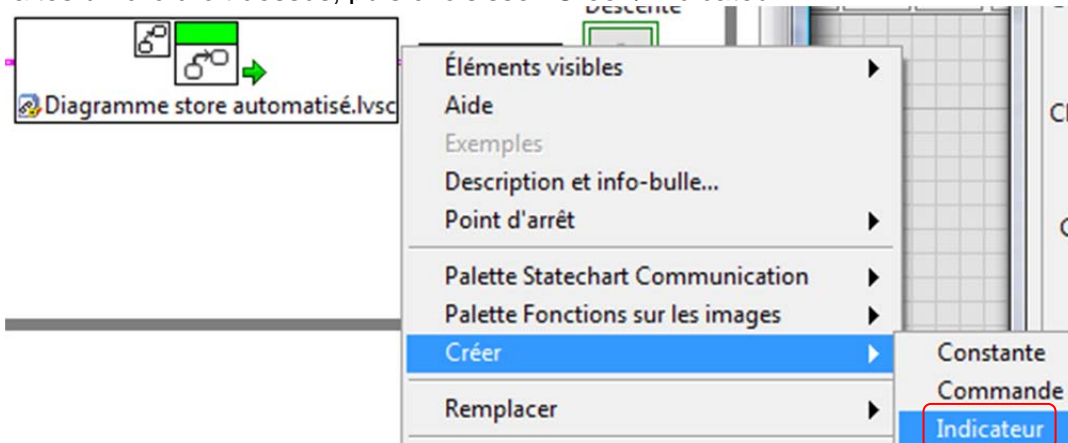
- dans la section *Statechart Diagram Display*, cocher *Show Terminal* :



- une sortie *Diagram Display* devient disponible :



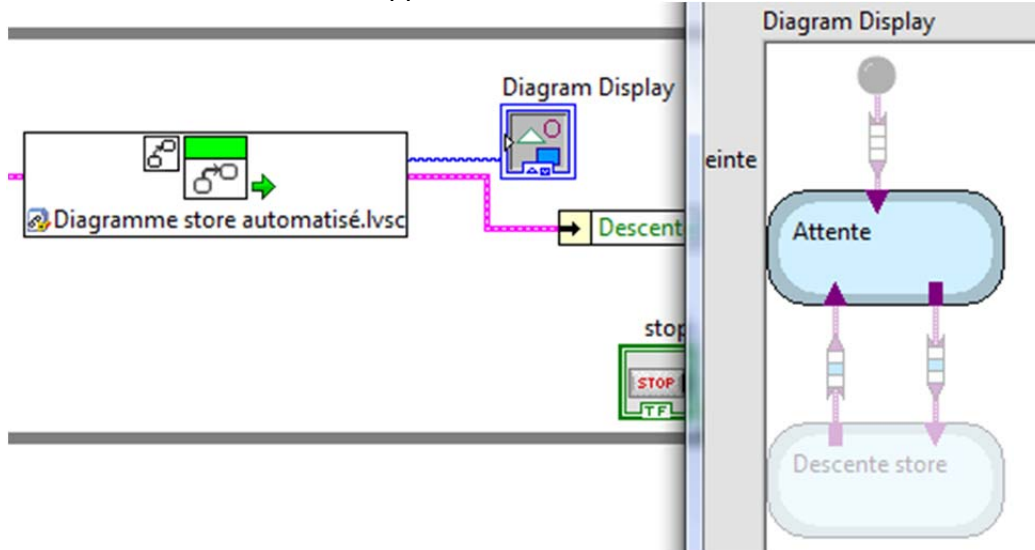
- faites un clic droit dessus, puis choisissez *Créer / Indicateur* :





Sciences et technologies de l'Industrie et du développement durable

- une fenêtre de visualisation apparaît sur la face avant :



2.5 Statechart synchrone / Statechart asynchrone.

Dans l'aide Labview apparaît très vite la notion de statechart synchrone et de statechart asynchrone. Ces notions ne seront pas développées, car nous utiliserons principalement des statechart synchrones.

Les statechart asynchrones sont destinés à gérer les évènements (Triggers), que nous n'aborderons pas dans cette formation.

Pour résumer :

- un statechart synchrone s'exécute dans un temps continu ;
- un statechart asynchrone s'exécute dans un temps discret, à intervalles régulier.

Le statechart asynchrone est donc destiné à des applications avancées type Temps réel (Real-time)... qui ne seront pas immédiatement abordées en STI2D !



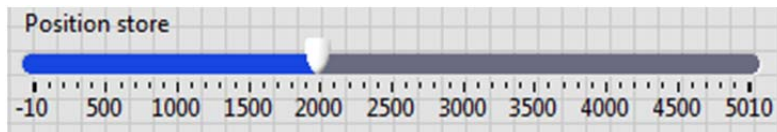
Sciences et technologies

de l'Industrie et du développement durable

3 Exercices

3.1 Gestion de la position du store.

Pour l'instant, rien ne simule vraiment la position du store. Remplacez l'entrée TOR « position basse » du diagramme d'état par une glissière allant de la position -10 à la position 5010 :



- en dessous de 0, le store est considéré comme se trouvant en *Position haute*.
- en dessous de 5000, le store est considéré comme se trouvant en *Position basse*.

Câblez et testez le fonctionnement.

De même, introduisez une glissière *Intensité lumineuse* allant de 0 à 1.

- en dessus de 0,5 la luminosité est considérée comme supérieur au seuil ;
- en dessous de 0,5, la luminosité est considérée comme inférieure au seuil.

3.2 Gestion de la montée et de la descente.

Modifiez le diagramme d'état de manière à ce qu'il réponde à la spécification de la page 3.

Câblez et testez le fonctionnement.

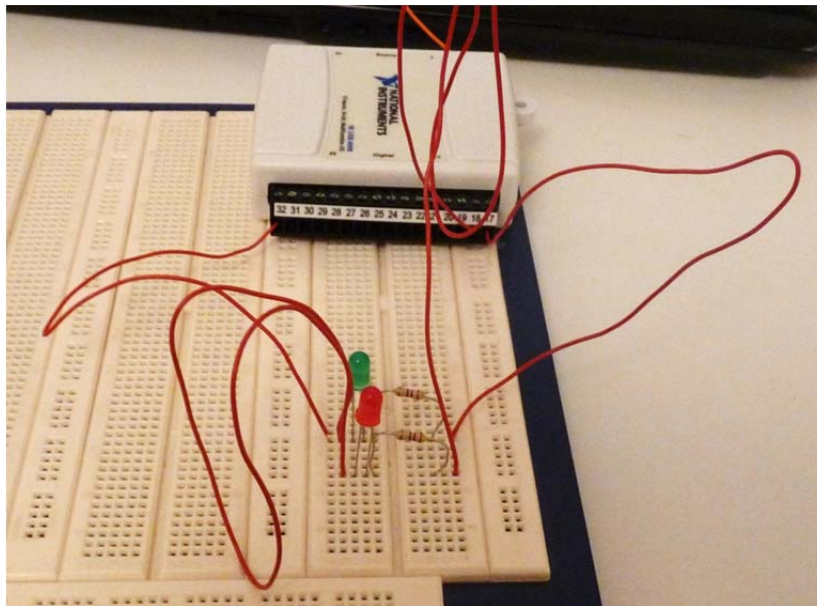


Sciences et technologies de l'Industrie et du développement durable

3.3 Câblage du fonctionnement avec des cartes entrée/sortie.

Il vous est demandé de simuler le fonctionnement sur une platine de câblage à l'aide de :

- une Led verte pour la montée ;
- une Led rouge pour la descente ;
- un potentiomètre pour simuler l'intensité lumineuse.



Pour cela, il est nécessaire de :

- câbler les entrées et les sorties comme indiqué dans les annexes ;
- utiliser les fonctions d'acquisition NI Daq.



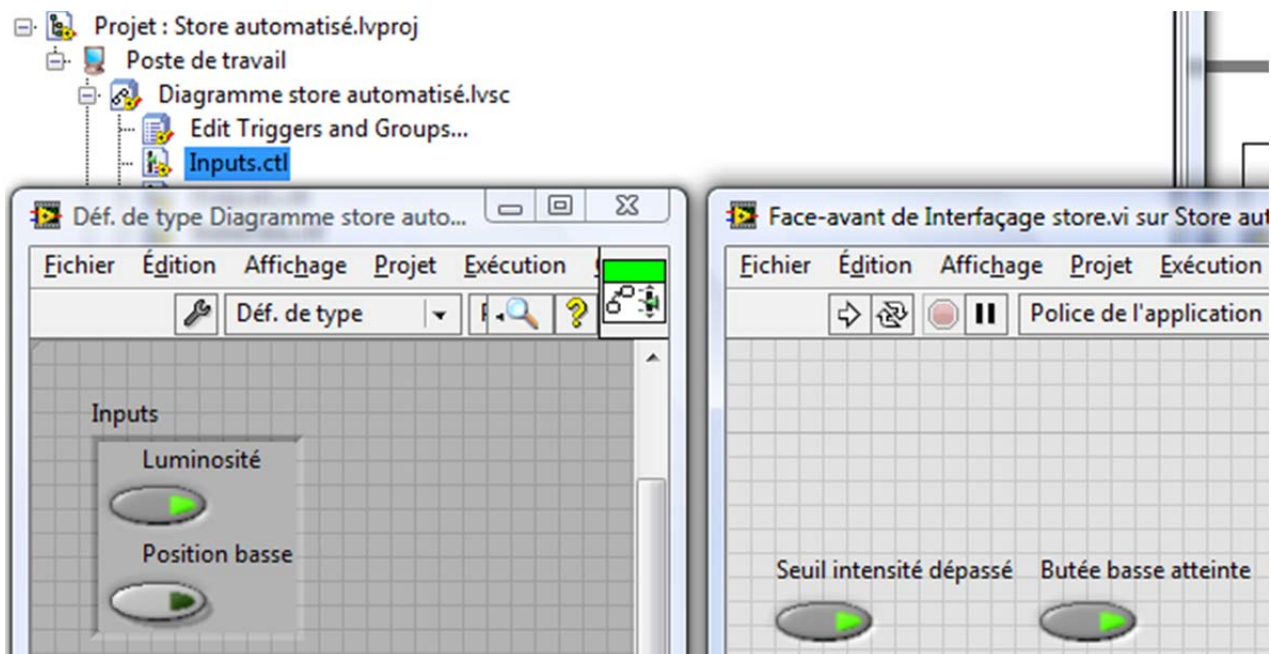
Sciences et technologies de l'Industrie et du développement durable

4 Annexes.

4.1 Câblage des entrées de Statechart permettant de personnaliser la face avant du diagramme de contrôle.

Le problème est le suivant : la méthode présentée chapitre 2.2.4 est efficace. Mais sur la face avant, les entrées sont cloisonnées dans le cluster *Inputs*.

Comment présenter les entrées de manière personnalisées, tant du point de vue de leur emplacement que de leur nom, comme le montre le diagramme ci-dessous ?

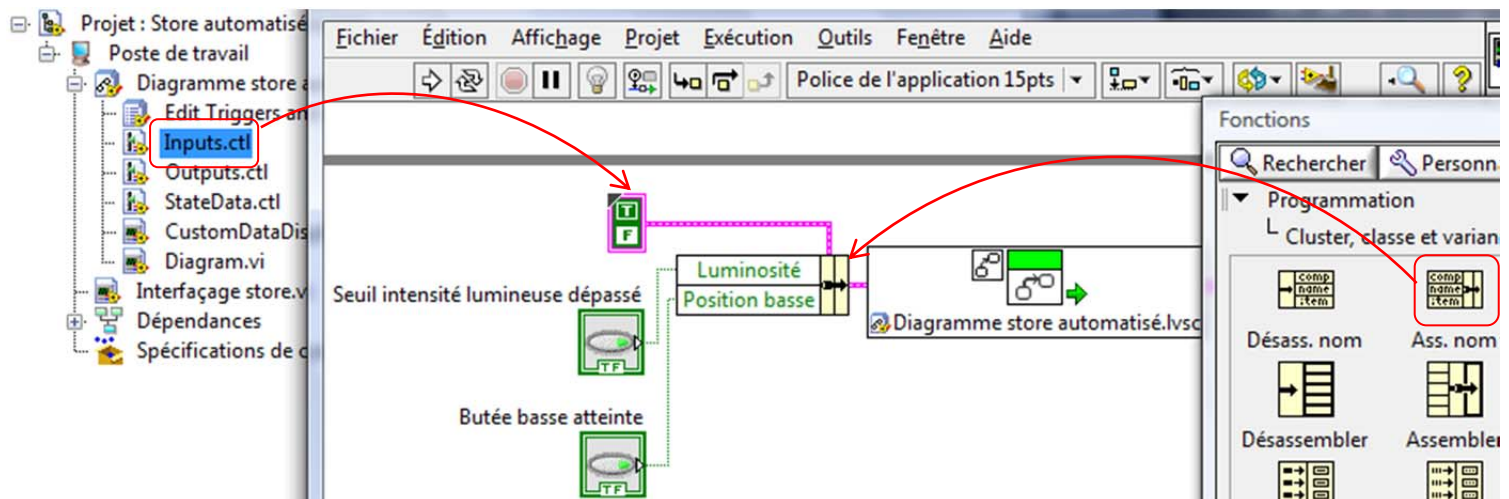




Sciences et technologies de l'Industrie et du développement durable

La méthode est la suivante :

- placer le cluster *Inputs* sur le diagramme ;
- placer une fonction d'assemblage de cluster par nom, présente dans la palette Programmation / Cluster / Assemblage par nom ;
- câbler le cluster *Inputs* sur l'entrée haute de la fonction placée ; vous remarquerez que les noms des entrées à câbler apparaissent ;
- placer les entrées sur la face avant du diagramme de contrôle comme indiqué ci-dessus. Ces entrées doivent être de même type que celles du cluster *Output*, mais elles peuvent avoir une autre présentation et un autre nom ;
- sur le diagramme, câbler ces entrées à la fonction *Assembler*.
- câbler le cluster ainsi créé au diagramme d'état.



4.2 Câblage des sorties de Statechart permettant de personnaliser la face avant du diagramme de contrôle.

Le problème est identique pour le cluster de sortie. La résolution est plus simple : il faut uniquement brancher la fonction Désassembler par nom à la sortie du diagramme d'état.

