

	LYCÉE JULES VIETTE MONTBELIARD	1STI2D S ystème d' I nformation et N umérique
	<p style="text-align: center;">Simulation du fonctionnement de la borne de recharge : diagramme d'états-transitions</p>	
		<p style="text-align: center;">Thème : Confort</p>

Objectifs de formation.

07 : Imaginer une solution répondre à un besoin.

Compétence attendue:

CO7.sin3. Exprimer le principe de fonctionnement d'un système à partir des diagrammes SYSML pertinents.

Programme.

1. Projet technologique
- 1.3 Description et représentation

Centre d'intérêt.

CI1. Configuration et performances du traitement de l'information

Niveau taxonomique: 3

Problématique de la séquence.

A partir du cahier des charges, vous devez simuler le fonctionnement de la borne en programmant un diagramme SYSML : le diagramme états-transitions avec Statechart de LabVIEW.

Connaissances abordées :

Diagrammes SYSML



2- DONNÉES DISPONIBLES POUR REALISER LA TÂCHE

Cahier des charges de la borne
Logiciel Labview : module State Chart

3- SITUATION DE TRAVAIL

- Démarche retenue :

- Investigation
- Résolution de problème technique
- Projet
- Créativité

- Type d'activité :

- Analyse
- Réalisation
- Expérimentation
- Conception

- Durée : 3 heures.

A-Présentation

La borne de recharge standard permet à un utilisateur de raccorder son véhicule électrique pour le recharger en toute sécurité et rapidement. Elle dispose d'un ou deux socles de prises, d'organes de signalisation, de commande et de transmission de données pour son exploitation et sa maintenance.

Elle est utilisée seule ou en grappe de bornes. La borne standard est destinée à être installée dans un environnement à accès privé ou surveillé, en intérieur comme en extérieur.

La borne étudiée a les fonctionnalités suivantes :

Dialogue borne-utilisateur (interface basique), pour chaque prise de la borne,

- Voyants : "Disponible" - "Défaut" sur la face avant, "Charge" sur le côté ;
- Boutons poussoirs : "Charge" - "Stop" sur le côté

Gestion de recharge,

- Marche : Immédiate – dès raccordement des prises du câble et appui sur le bouton "Charge".
- Arrêt : automatique – batterie pleine ; manuel – par passage du badge RFID et appui sur le bouton stop

Identification d'utilisateur pour autorisation de recharge,

- Lecture de badge RFID.

Verrouillage,

- De l'ouverture du volet.
- De la prise branchée, en cours de rechargement.

Transmission de données,

- Entre borne et véhicule, via prise de recharge : protocole IEC 61851.
- Entre borne et système de gestion d'énergie : protocole Modbus.

Voyants face Avant

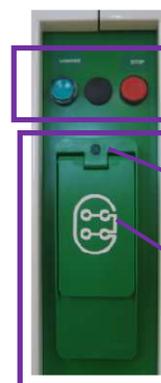


Non utilisé



Lecteur badge RFID

Côté droit de la borne



Boutons poussoirs

Prise de type 3

Voyant prise

Volet prise

Borne standard sur pied

B-Introduction au diagramme Etats-Transitions

L'objectif de ce TP est de simuler le fonctionnement de la borne grâce au diagramme SysML d'états-transitions.

Pour apprendre à utiliser cet outil, on va programmer un diagramme d'état-transition très simplifié.

Définition : un diagramme Etat-Transition rassemble et organise les états et les transitions d'un système. Dans un même système, on peut avoir plusieurs graphes d'états-transitions.

Etat initial et état final.

L'état initial est un pseudo état qui indique le début du graphe orienté ou l'état de départ. Il est forcément unique pour une région.

Etat initial



L'état final est un pseudo-état qui indique que le diagramme d'états-transitions est terminé.

Etat final



Transition.

Une **transition** définit la réponse du système à l'occurrence d'un événement. Elle possède un nom, des événements déclenchant, une garde et des actions.

La garde est une condition booléenne qui autorise ou non la prise en compte de l'évènement.

Une transition peut ne pas avoir d'évènement déclencheur, elle contient alors une condition de garde, c'est notamment le cas pour les transitions venant d'états initiaux ou d'états historiques qui ne peuvent rester actifs.



Etats.

Un **état** est représenté par un rectangle à coins arrondis, il a un nom (Etat N). Le rectangle peut être séparé par un trait horizontal, la partie haute est réservée au nom et la partie basse à des événements internes.

On va maintenant étudier une partie du diagramme de fonctionnement de la borne de recharge électrique concernant l'identification du badge.

Diagramme simplifié : partie identification de la borne.

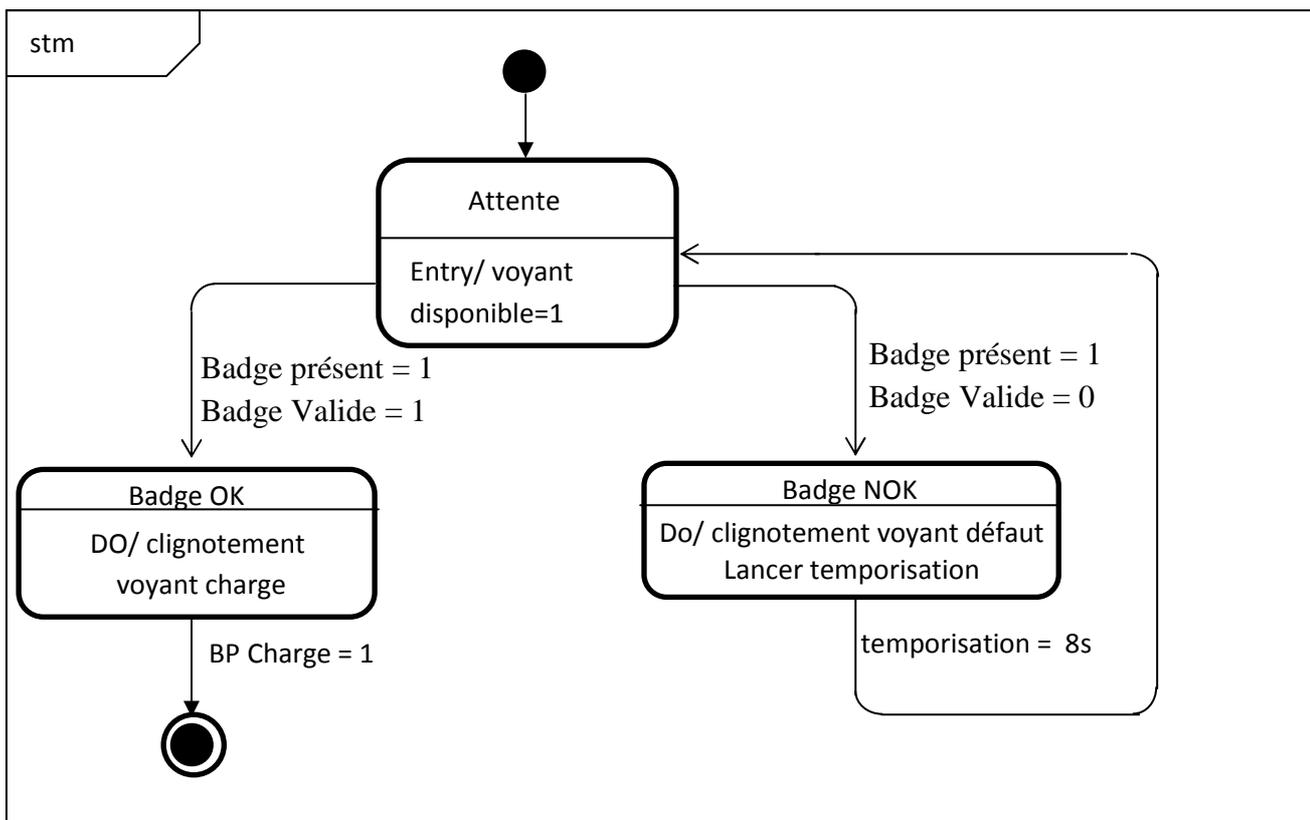
Après la mise sous tension, la borne doit être en attente d'un événement. Le premier état est donc un état d'attente. L'événement pour quitter cet état est la lecture d'un badge RFID. Si ce badge est autorisé, la borne passe à l'état badge OK sinon elle passe à l'état badge NOK.

Pendant l'état d'attente, le voyant vert « Disponible » est allumé.

Pendant l'état badge OK, le voyant « charge » clignote

Pendant l'état badge NOK, le voyant « défaut » clignote.

Le diagramme (incomplet) est le suivant :

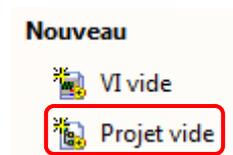
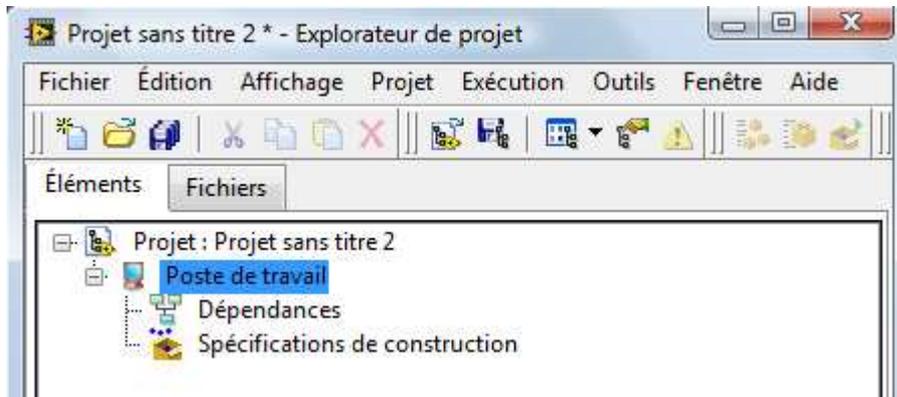


La durée réelle de la temporisation est de 30 secondes. La durée de 8 secondes a été choisie pour accélérer la simulation.

C- Programme d'un diagramme Etats-Transitions

C.1) Création d'un projet.

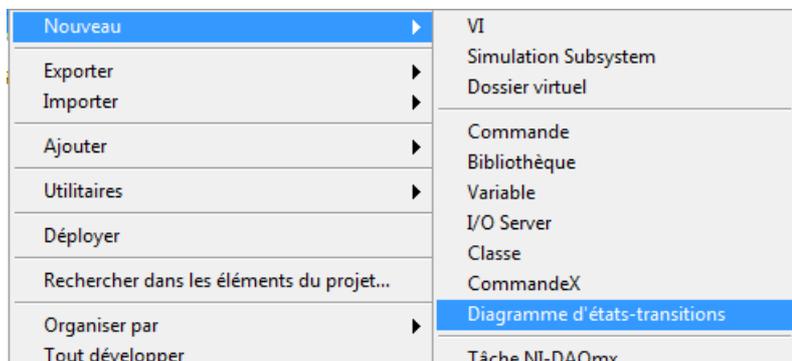
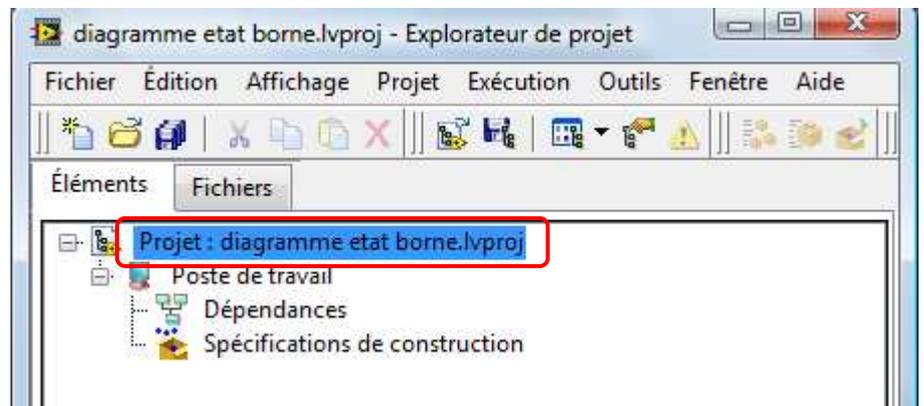
Lancez LabVIEW, puis créez un nouveau projet vide. Ce projet va contenir tous les objets nécessaires à votre programmation.



Sauvez votre projet, par exemple sous le nom

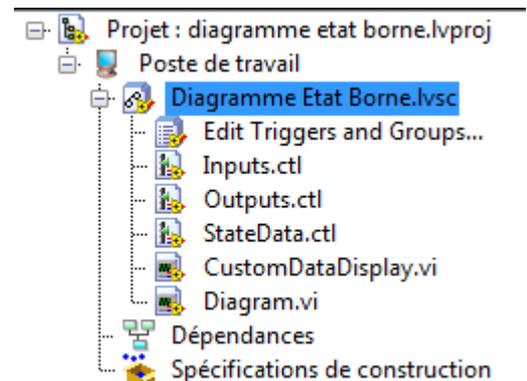
« diagrammeetat borne »

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



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

- une interface de gestion des événements (Triggers). Vous n'allez pas les gérer dans ce TP ;
- 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 TP ;
- un VI Diagram, qui servira à programmer graphiquement le diagramme d'états.



C.2) Création du diagramme d'état.

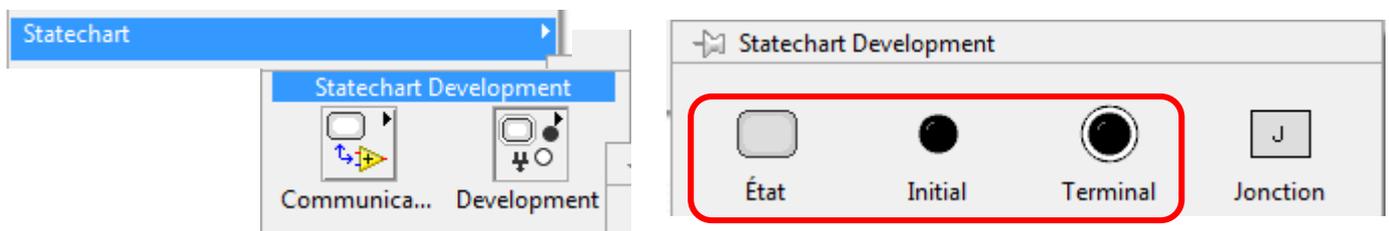
La démarche de création d'un diagramme d'état sous LabVIEW est exactement la même que celle que l'on pourrait utiliser pour les Grafset ou pour les Diagrammes d'état sous Simulink. Seule la démarche manipulative est différente. Il vous faudra donc :

- 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.

C.2.1) Définition des états.

Comme l'indique le diagramme d'état page 4, le diagramme d'état simplifié comporte trois états : *Attente* et *Badge OK* et *badge NOK*.

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

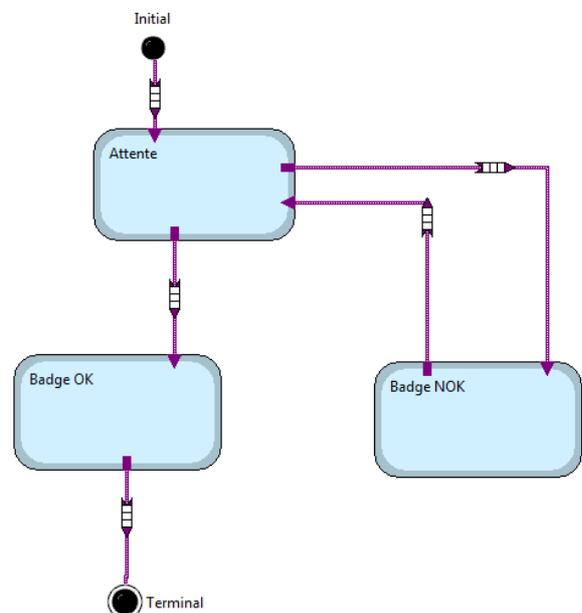


Renommez les états en cliquant sur les noms des états.

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.**



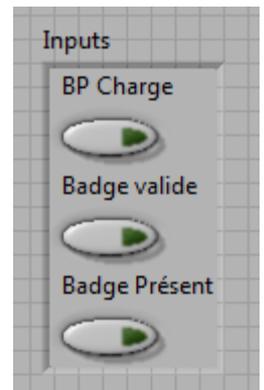
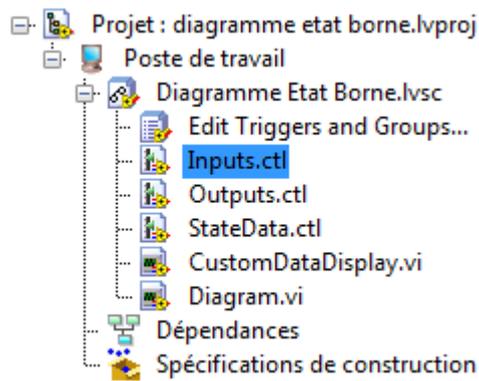
C.2.2) Définition des entrées.

Comme indiqué sur le diagramme d'état page 4, le diagramme d'état comporte trois entrées :

- Badge Présent
- Badge Valide
- BP Charge

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

Enregistrer le VI et le fermer.



C.2.3) Définition des sorties.

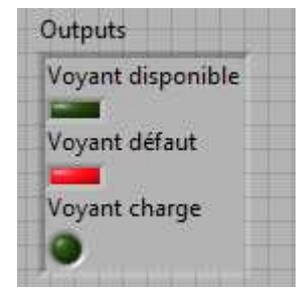
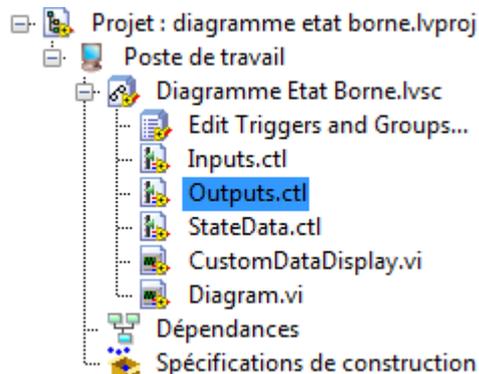
Le diagramme d'état comporte trois sorties, que l'on peut nommer

- *Voyant charge*
- *Voyant défaut*
- *Voyant disponible*

Ces sorties sont de type binaire car elles sont vraies ou fausses.

Pour les définir :

- double cliquer sur la VI outputs qui se trouve dans le projet ;
- placez trois sorties LED dans la fenêtre Outputs ; Vous pouvez changer la couleur du voyant défaut en cliquant dessus et propriétés
- effacez la sortie qui était placée dans cette fenêtre en guise d'exemple.



Vous pouvez fermer le VI.

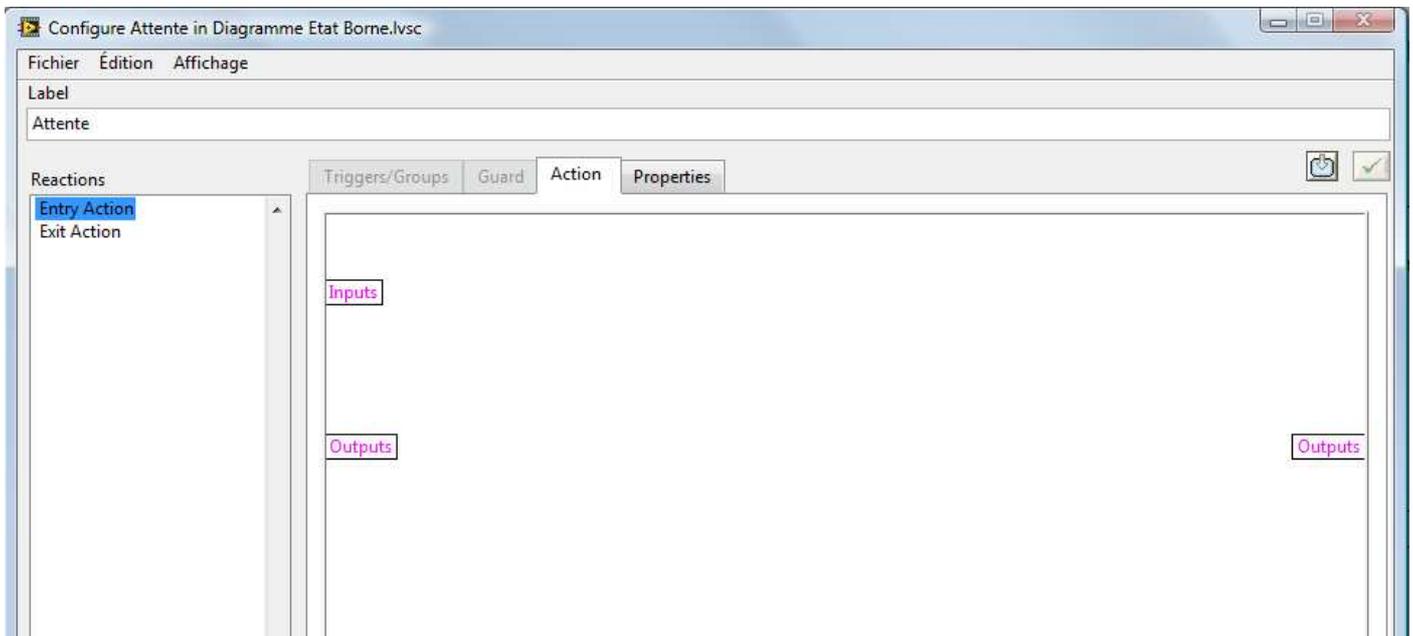
C.2.4) Codage des actions à effectuer dans les états.

Vous allez maintenant coder le comportement des sorties. A savoir :

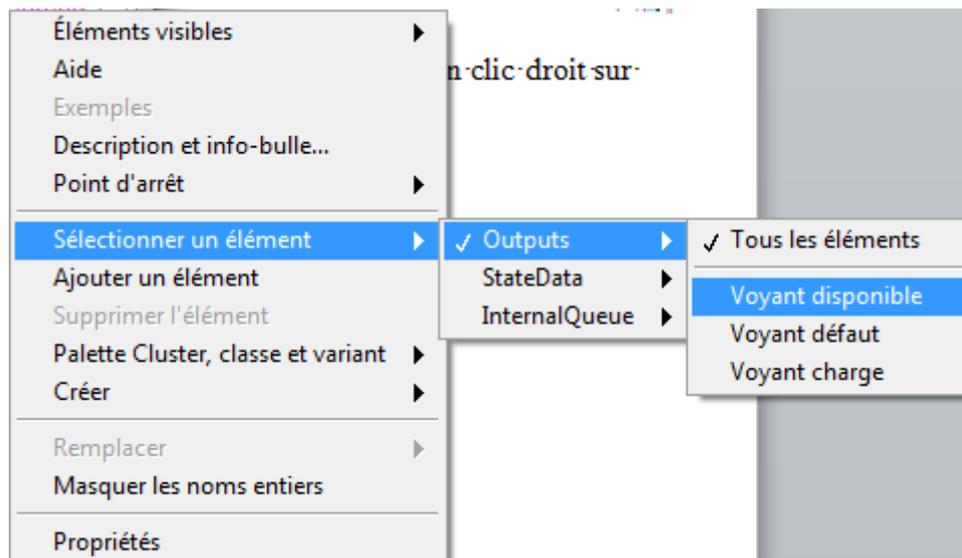
- dans l'état d'attente, la variable de sortie *Voyant disponible* doit être mise à 1 (True) ;
- dans l'état Badge OK, la variable *Voyant charge* doit être mise à 1 (True) ;
- dans l'état Badge NOK, la variable *Voyant défaut* doit être mise à 1 (True).

Traitement de l'état « attente » :

- retournez dans le VI *Diagram.VI* ;
- double cliquez sur le bord de l'étape *attente* : la fenêtre de codage du comportement des sorties s'ouvre ;



- Déclarez que vous voulez coder le comportement de la sortie *Voyant Disponible* en faisant un clic droit sur Output à droite, puis *Sélectionner un élément / Outputs / Descente*

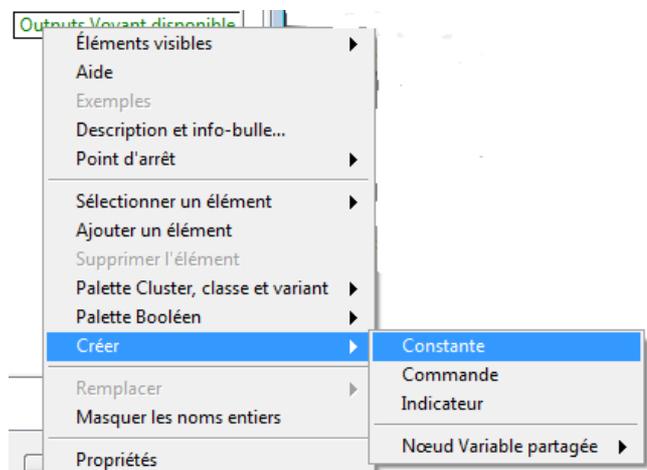
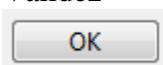


- Activez la variable en plaçant une constante True. Pour cela, faites un clic droit sur la sortie, puis sélectionnez Créer / Constante :

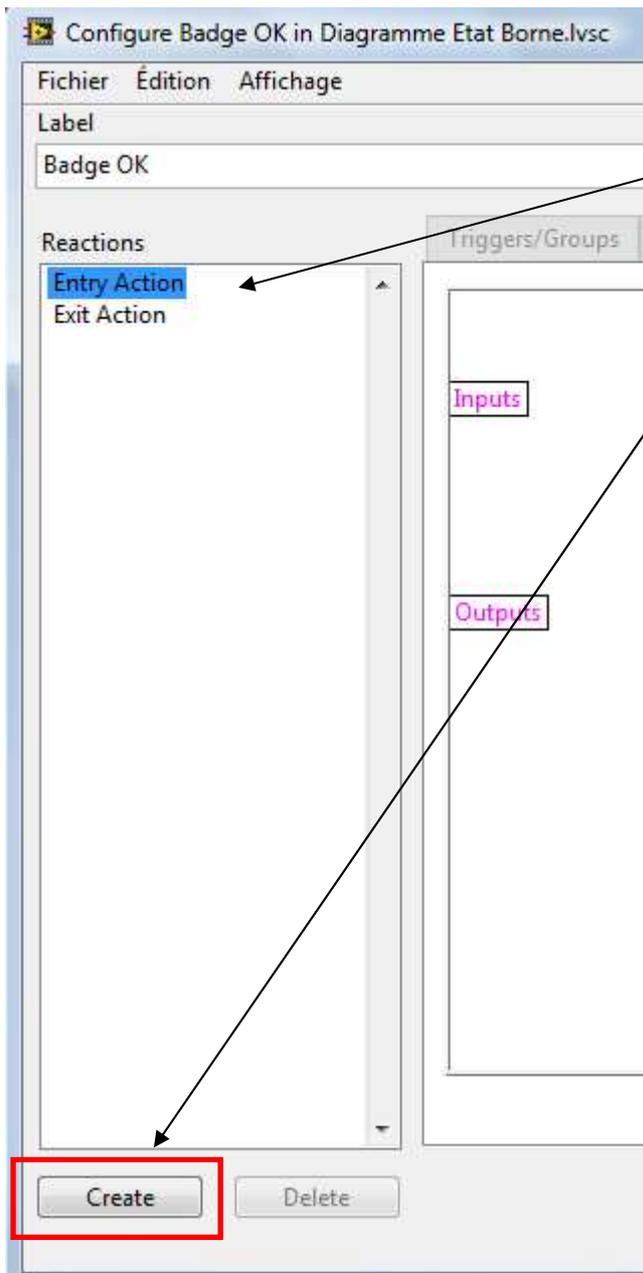
si besoin, cliquez sur la constante pour faire changer son état à True :



- Validez



- Programmer l'état « badge OK ».
 Dans cet état, il faut programmer l'action DO.
 Double cliquer sur le bord de l'état.



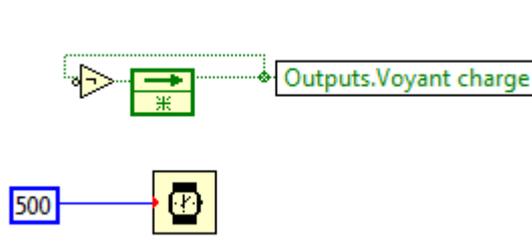
DO n'existe pas, il faut le créer

Après l'appui sur Create, vous obtenez :



Saisir DO

Et programmer l'action associée à DO :



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.*

A vous de coder l'action associé à l'état : « badge NOK »

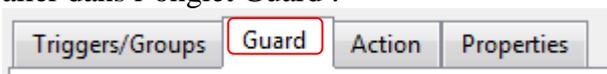
Le comportement des sorties est prêt. Il faut maintenant programmer le comportement des transitions.

C.2.5) Codage des transitions

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

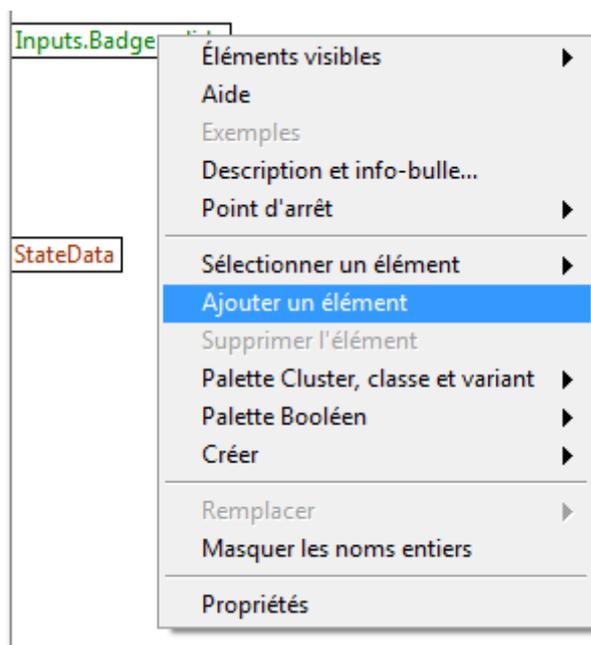
Pour coder la transition Attente => Badge OK :

- 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 C.2.4 Codage des actions à effectuer dans les états, sélectionner l'entrée Badge Valide : `Inputs.Badge valide`

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



Vous devez obtenir ce diagramme :



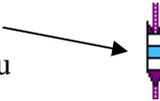
Il faut maintenant câbler la condition Bade Valide et Badge Présent avec les fonctions booléennes.



Valider avec OK.

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.



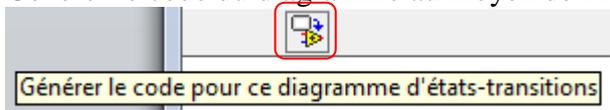
A vous de coder les autres transitions.

Pour la temporisation (transition de badgeNOK vers attente) utiliser



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 :



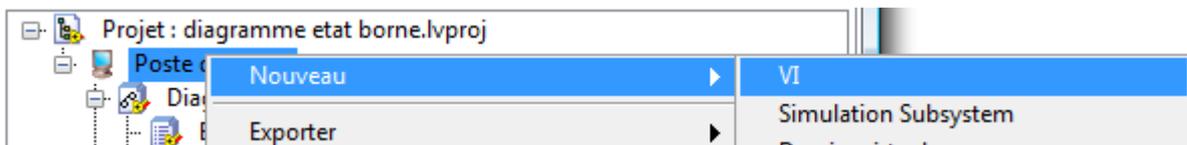
- Vous remarquez que l'icône passe à l'état grisé.
- Fermer le diagramme.



C.3) Programmation du VVI de contrôle.

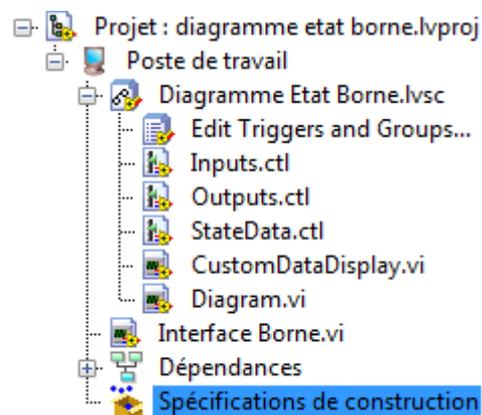
C.3.1) Création du VI

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



Enregistrez-le avec un nom approprié. **Interface Borne**

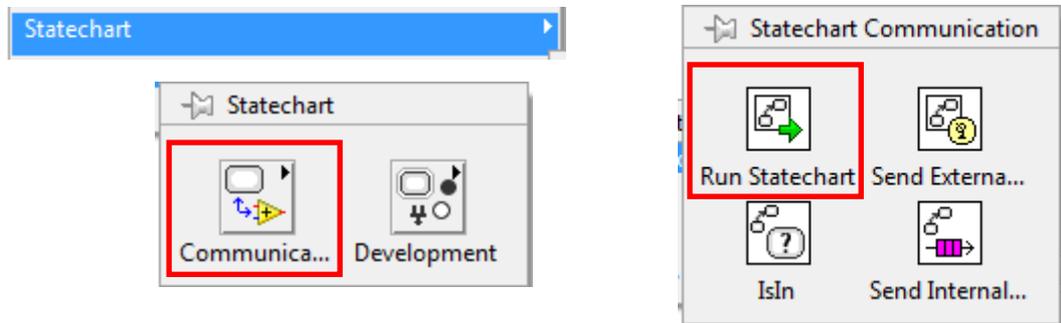
Remarquez son emplacement dans le projet :



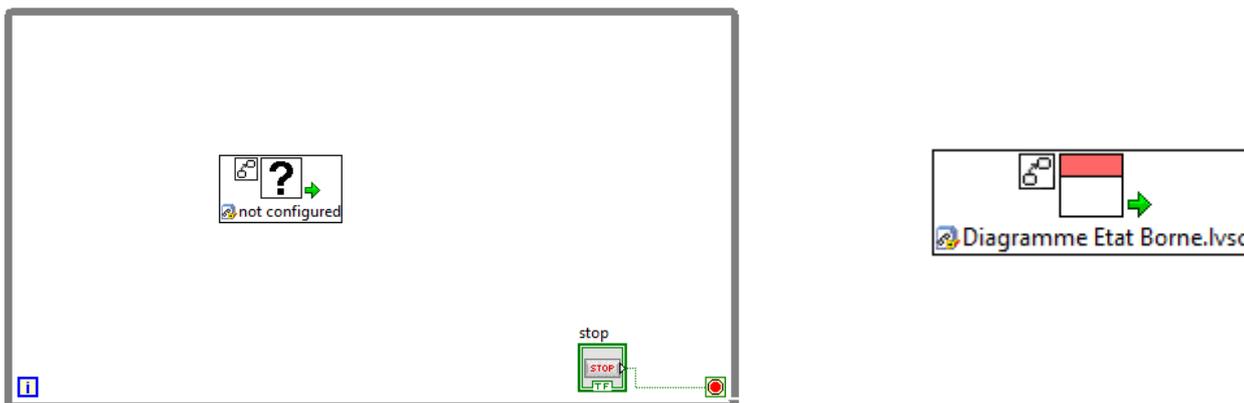
C.3.2) Mise en place d'une boucle

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 *RunStatechart* 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.



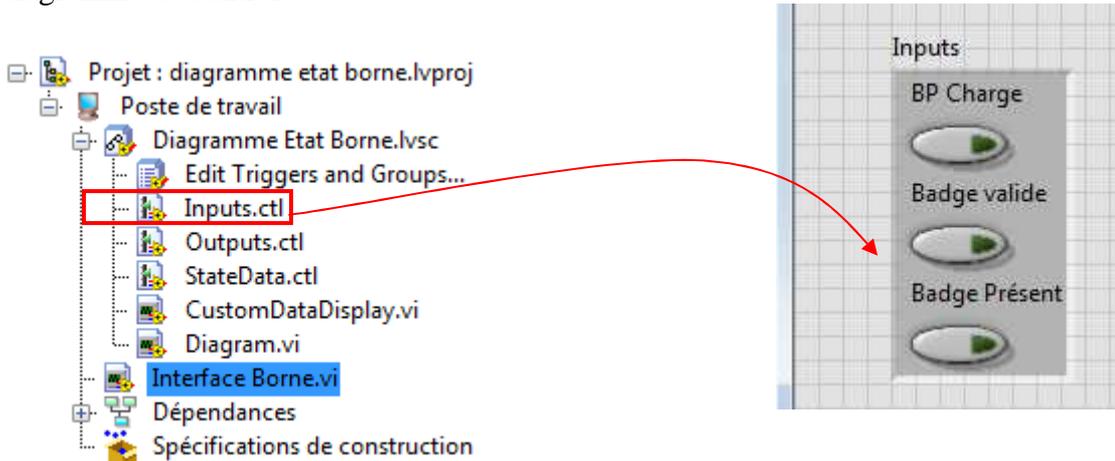
C.3.3) Câblage des entrées.

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

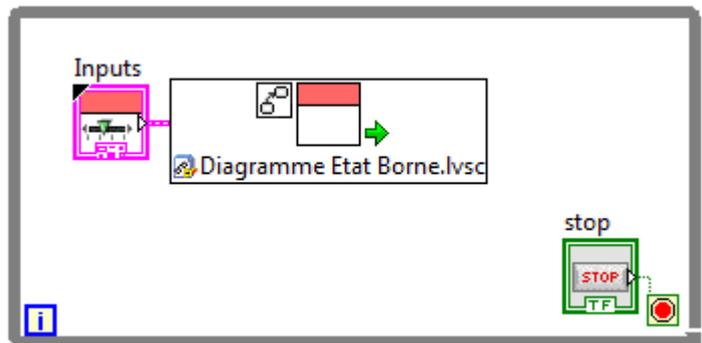
Un cluster est une ligne dans laquelle sont rangées toutes les informations à passer. Un cluster est comparable à une trame de réseau.

Il existe de nombreuses méthodes pour câbler des entrées au diagramme d'état dans le VI de contrôle.

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

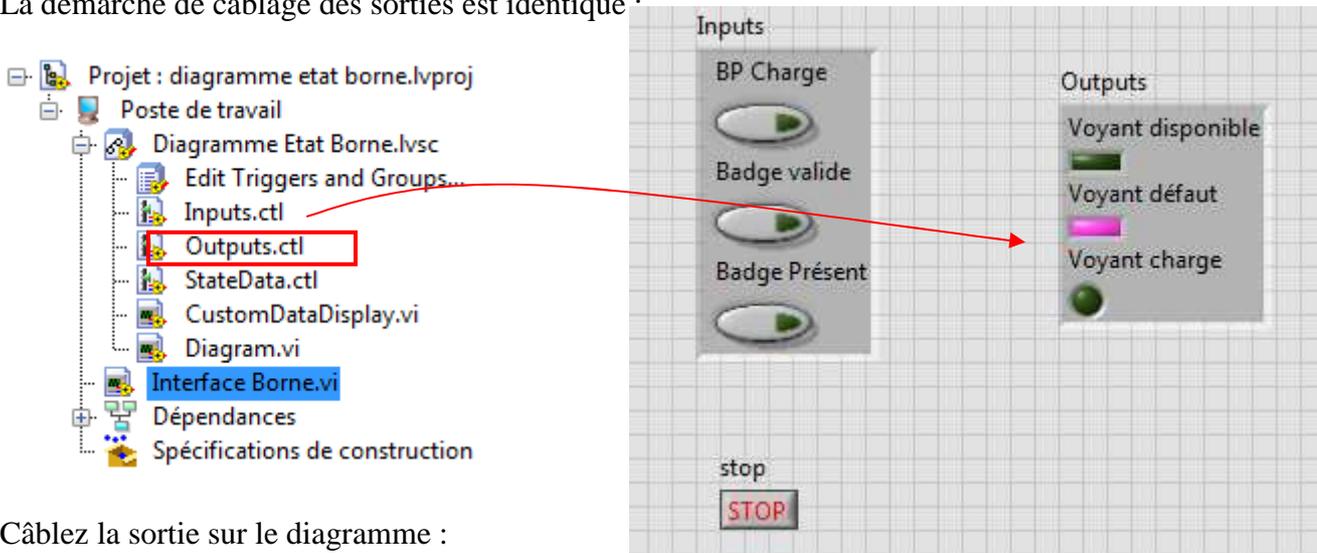


Le cluster des entrées s'est aussi inséré sur le diagramme ; il faut seulement câbler ces entrées au diagramme d'état :

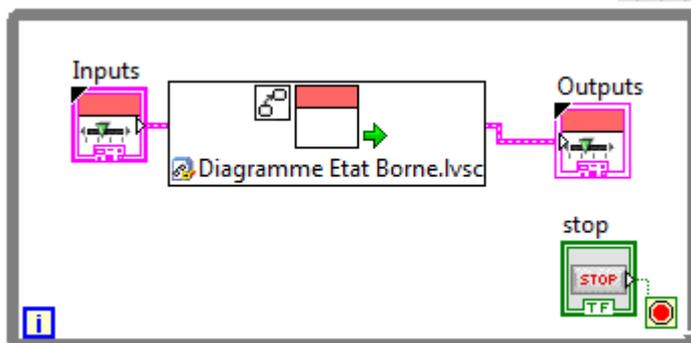


C.3.4) Câblage des sorties.

La démarche de câblage des sorties est identique :



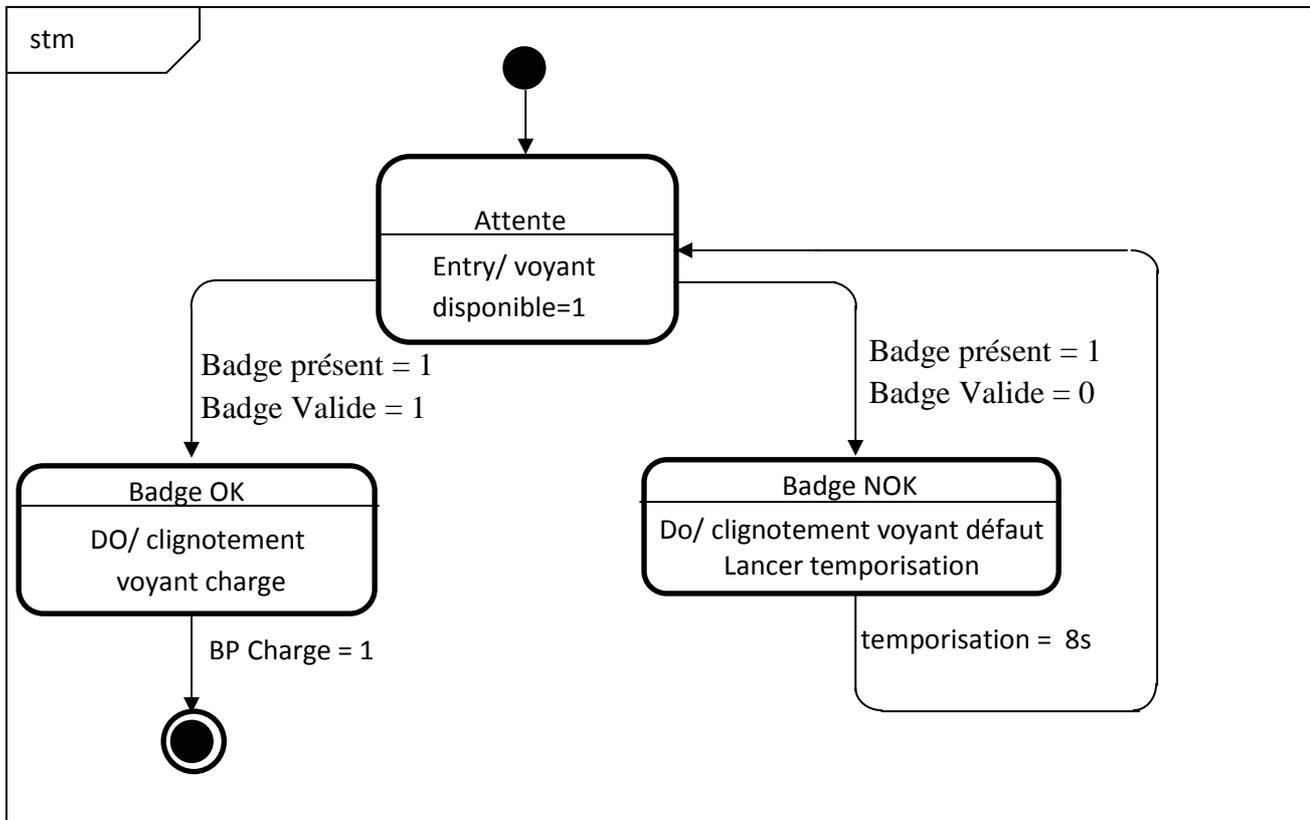
Câblez la sortie sur le diagramme :



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

Lorsque les sorties ont mises à 1 dans un état elles restent à ce niveau logique même si l'état n'est plus actif.

Procéder de la même manière pour terminer le diagramme ci-dessous.



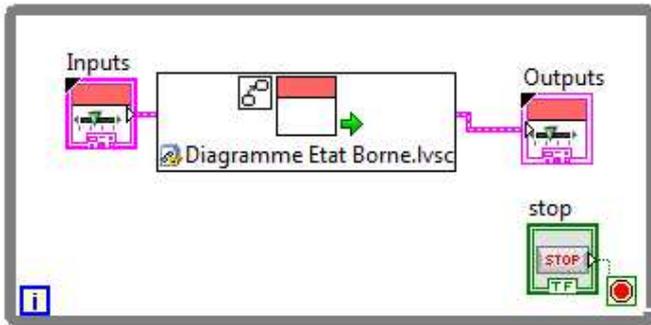
Lorsque les sorties ont mises à 1 dans un état elles restent à ce niveau logique même si l'état n'est plus actif.

Modifier l'état Badge OK pour éteindre le voyant disponible.

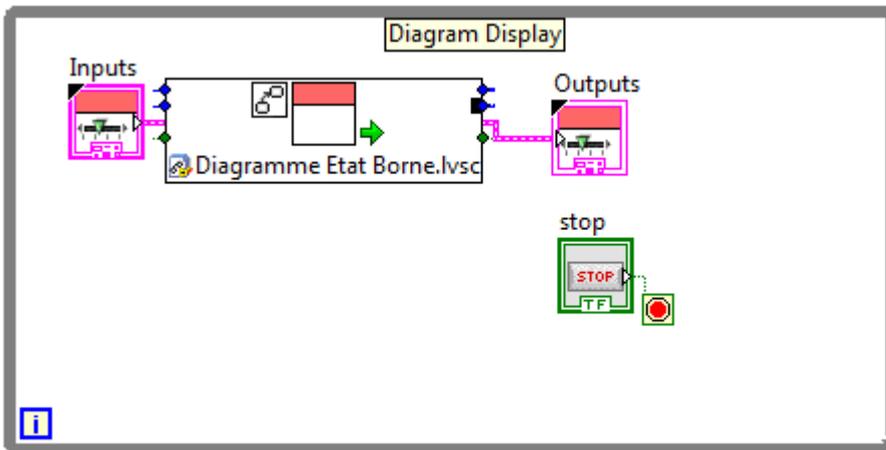
C.4) Visualisation de l'évolution du Diagramme d'état.

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

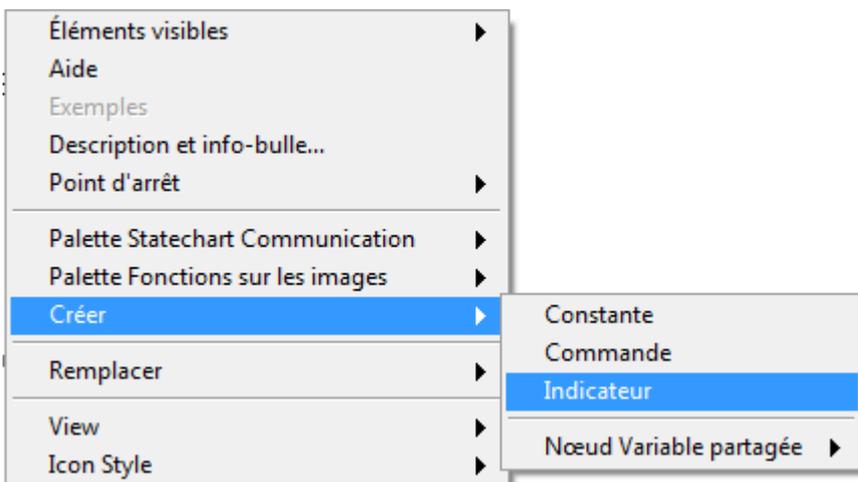
- Double cliquez sur le diagramme d'état transitions; dans la section *StatechartDiagram Display*, cocher *Show Terminal* :



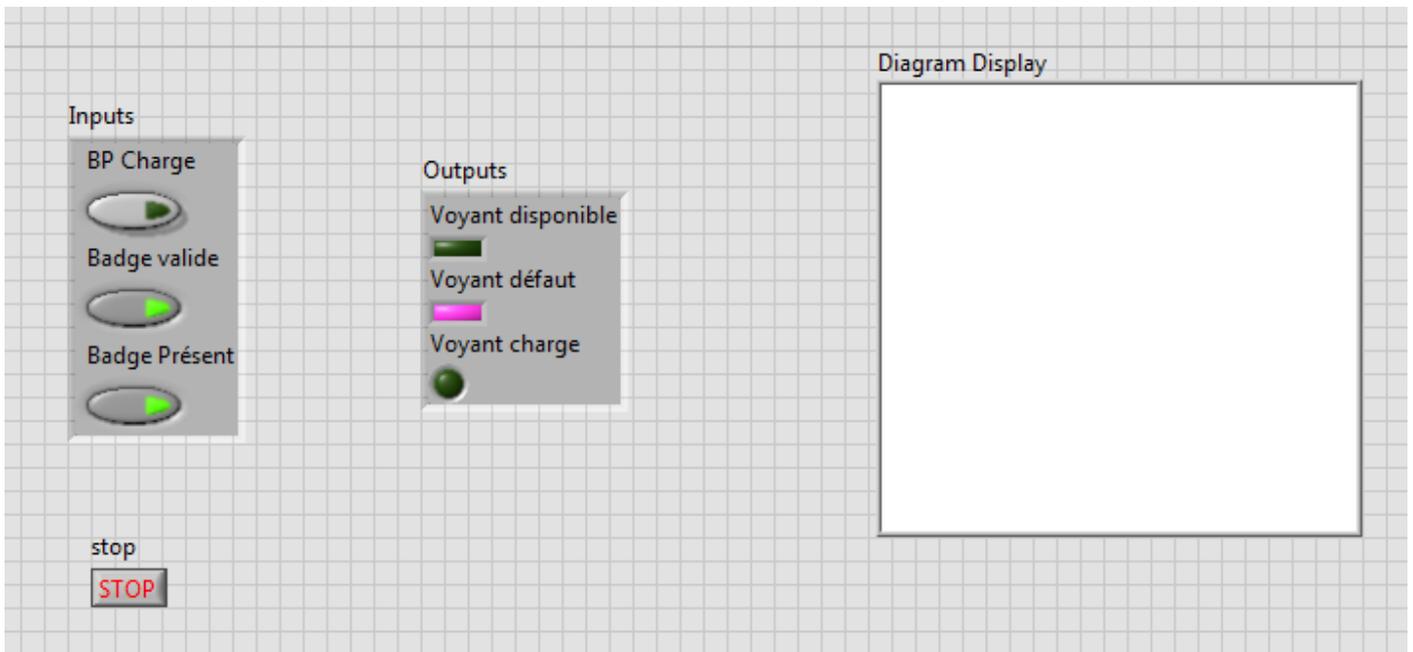
- une sortie Diagram Display devient disponible :



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



- une fenêtre de visualisation apparaît sur la face avant.



Exécutez le programme : le diagramme d'états-transitions apparaît.

Faire valider par le professeur.

D- Programme du diagramme Etats-Transitions de la borne

On va maintenant compléter le diagramme précédent. Le fonctionnement de la borne est toujours simplifié.

A l'état d'attente, le verrou de la trappe de la prise doit être fermé (Verrou trappe = 1) et le voyant disponible est allumé (Voyant disponible = 1).

Après l'authentification réussie :

le voyant charge (Voyant charge = 1) doit clignoter, le voyant disponible s'éteint (Voyant disponible = 0).

si après 8 secondes, le BP charge n'est pas appuyé, revenir à l'étape d'attente,

si l'utilisateur appuie sur le BP charge en moins d'une minute, la trappe de la prise doit se déverrouiller (Verrou trappe = 0),

ensuite, l'utilisateur doit connecter la prise (Prise = 1)

la prise doit être verrouillée (Verrou Prise = 1)

La charge démarre (contacteur = 1) et le voyant au-dessus du socle est allumé (voyant socle = 1)

La personne vient reprendre son véhicule :

Après l'authentification réussie : (badge valide = 1, badge présent = 1)

Attendre l'appui sur le BP Stop (BP Stop = 1). Si après 8 secondes, le BP stop n'est pas appuyé revenir à l'étape précédente.

Si l'utilisateur appuie sur le BP Stop, il faut déverrouiller la prise (Verrou Prise = 0), éteindre le voyant au-dessus du socle (Voyant Socle = 0) et ouvrir le contacteur (contacteur = 0)

L'utilisateur doit débrancher la prise (Prise = 0)

Ensuite, la trappe se verrouille (Verrou trappe = 1) et le voyant disponible est allumé (voyant disponible = 1)

Travail demandé :

- a) Recenser les entrées et les sorties pour votre diagramme.
- b) Créer le diagramme dans LabVIEW.
- c) Tester le fonctionnement et faire valider par le professeur.