

# Quand LabView parle SysML

CHRISTOPHE REVY<sup>[1]</sup>

*Pour former les enseignants de l'académie de Dijon à la plate-forme logicielle LabView, les formateurs ont conçu une application de contrôle-commande de store programmée à partir d'un diagramme comportemental SysML – langage qui fait maintenant partie intégrante des attendus des formations en sciences industrielles de l'ingénieur, prébac et postbac.*

À la suite des réformes du bac STI devenu le bac STI2D, du bac S-SI et des programmes de CPGE, l'académie de Dijon et la Région Bourgogne ont choisi, entre autres, l'environnement de développement graphique LabView comme plate-forme logicielle pour l'acquisition de données, la programmation du comportement et la partie gestion de projet. Le logiciel LabView, grâce à sa programmation graphique, permet d'acquérir et d'analyser des mesures (traitement du signal), de contrôler des instruments autonomes, de créer des systèmes de contrôle et de surveillance mais aussi d'automatiser des systèmes de tests et de validation.

Pour y former les professeurs des lycées enseignant dans les sections STI2D, S-SI, les STS et les CPGE de l'académie de Dijon, nous avons choisi de réaliser avec LabView la partie contrôle-commande d'un store extérieur automatisé qui équipe tous les lycées de l'académie, utilisant des signaux mesurant la luminosité et la vitesse du vent **1**. Notre objectif était de développer et mettre en pratique une application LabView reproduisant le comportement du système réel, illustrant les fonctions d'acquisition, de traitement de l'information, de modélisation et de commande de systèmes à partir d'une machine à états/transitions (diagramme de comportement SysML).

## Une formation en quatre étapes

Le plan de formation LabView, mis en place avec Robert Papanicola, professeur en CPGE au lycée Charlemagne à Paris, vise dans un premier temps un aspect technique de la plate-forme de développement. Une progression en quatre étapes nous a permis de réaliser différentes versions de l'application, de la plus simple (description en logique) à celle où le comportement est

**mots-clés**  
démarche  
pédagogique,  
travaux pratiques

modélisé par un diagramme comportemental SysML, et le système réel commandé par un appareil autonome. Voici ces quatre objectifs pédagogiques :

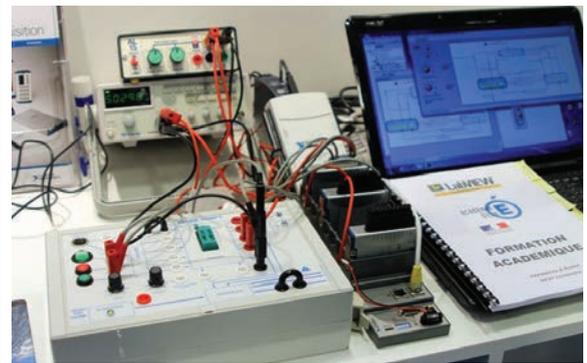
Acquisition et mesure  
Traitement et contrôle-commande  
Introduction au contrôle par diagramme état/transition de SysML  
Mise en œuvre d'un appareil autonome, le CompactRIO

Dans un second temps, les usages pédagogiques en cours ou en projet sont abordés (ils ne seront pas développés dans cet article).

## Le support de l'étude

Le choix s'est porté sur le store Somfy, système possédé par un très grand nombre d'établissements.

Un groupe de travail de sciences de l'ingénieur de l'académie de Dijon a créé, il y a quelques années, un module d'interface entre un microprocesseur PIC et des capteurs et actionneurs du store **2**. Cette platine nous



**1** La présentation de l'application lors du salon Éducatec 2012



**2** Le module d'interface Somfy

[1] Professeur certifié de sciences industrielles de l'ingénieur option « informatique et numérique » au lycée Catherine-et-Raymond-Janot de Sens (89). Courriel : christophe.revy@ac-dijon.fr

permet, en SI, de commander le store à l'aide d'une programmation en C ou graphiquement via FlowCode. Je l'ai donc utilisée comme interface avec la partie opérative.

Pour la chaîne d'information (partie commande), j'ai dans un premier temps utilisé une carte d'acquisition USB de National Instruments (NI USB-6212) ; cette solution nécessite un ordinateur supportant LabView. Le programme créé commandera donc l'automatisme du système réel en fonction du comportement voulu **3**. Les liaisons sont définies dans le diagramme de bloc interne du système **4**.

Dans un second temps, j'ai utilisé comme chaîne d'information le CompactRIO (voir « En ligne ») de National Instruments **5**, nouvelle dotation du conseil régional pour les établissements de l'académie. Ce système embarqué et reconfigurable permet des acquisitions, du contrôle et de la communication. Son architecture intègre des modules d'entrées/sorties (beaucoup de choix en fonction de nos applications), un châssis FPGA (*field-programmable gate array*) reconfigurable et un contrôleur. Il est programmable par la plate-forme LabView.

Le CompactRIO remplace la carte d'acquisition USB et l'ordinateur. La suite LabView permet l'acquisition et le traitement des données par programmation graphique, et son module (*toolkit*) Statechart de programmer un diagramme état/transition, diagramme SysML à enseigner en STI2D et en S-SI.

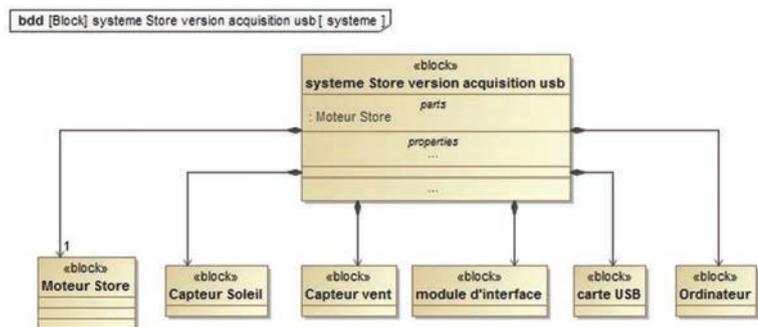
L'application qui a été voulue est bien une réplique de l'application du système réel fourni par Somfy.

### Acquisition et mesure

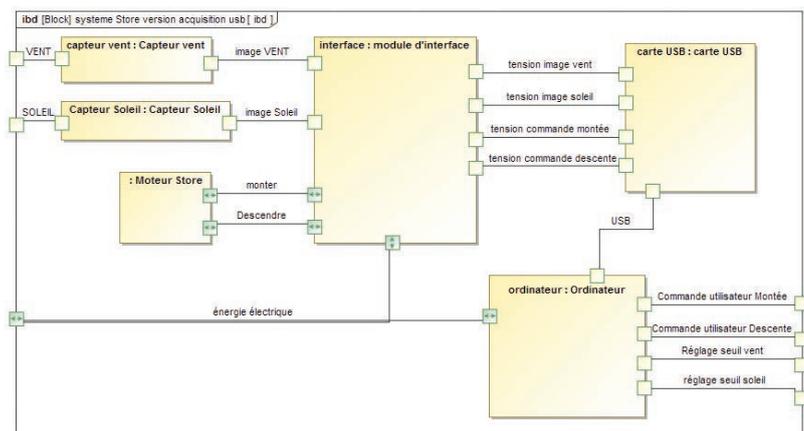
*Matériel utilisé* : un ordinateur avec LabView et une carte NI USB-6212

*Objectif* : acquérir les bases de l'acquisition d'un signal avec LabView, de la notion d'échantillonnage et de la connexion des E/S.

Dans ce premier temps, l'application, support de la formation, doit permettre de découvrir l'utilisation de LabView dans le cadre de l'acquisition de données uniquement. Ces données sont portées par un signal analogique pour le phénomène physique soleil (tension analogique variant avec la luminosité) et par un signal discrétisé pour le phénomène physique vent (créneaux dont la fréquence dépend de la vitesse du vent). Pour l'acquisition des données, nous avons fait appel



**3** La composition de la première solution matérielle (diagramme de définition de bloc)



**4** Le diagramme de bloc interne du système montrant les connexions



**5** Le CompactRIO avec ses modules d'entrées/sorties

à la carte multifonction USB-6212 : nous utilisons les entrées analogiques afin de mettre en évidence les phénomènes de fréquence d'échantillonnage.

Côté logiciel, il s'agit de réaliser un VI (*virtual instrumentation*) : programme sur la plate-forme de développement d'acquisition utilisant les données fournies par la carte USB-6212, avec une interface homme-machine (IHM) simple intégrant des graphes déroulants.

On trouve en **6** l'acquisition des deux grandeurs, l'une, dont la valeur analogique est directement utilisable, issue du capteur soleil, l'autre, issue du capteur vent, dont il faut calculer le temps entre deux fronts, afin d'en déduire la fréquence puis la vitesse du vent. On note sur cette figure que la programmation avec LabView est totalement graphique : on place sur le schéma des blocs représentant les fonctions ; les liens portent quant à eux les flux d'information.

La figure **7** montre la face avant (l'IHM) de ce premier programme, deux écrans matérialisés par le logiciel LabView pour la partie acquisition, correspondant aux icônes des graphes déroulants de la figure **6**.

**Contrôle-commande**

*Matériel utilisé* : un ordinateur avec LabView et une carte NI USB-6212

*Objectif* : réaliser le traitement des signaux avec LabView, appréhender l'élaboration d'un contrôle et de sa commande.

Dans ce deuxième temps, l'IHM de LabView doit permettre de régler les seuils (vent et soleil), de traiter les données et de commander la descente et la montée du store, pour rendre le processus automatique. Remplacer la partie commande existante impose d'acquérir les informations des capteurs de vent et de soleil, et de les traiter afin de pouvoir élaborer en conséquence les ordres à envoyer au moteur du store.

À cette étape correspondent quatre versions différentes du logiciel d'application.

Dans la première, le traitement des données est réalisé en logique combinatoire, et les commandes du moteur à l'aide de l'assistant DAQmx. Cela permet de découvrir des fonctions de calcul et de traitement logique.

Dans la deuxième, la commande du moteur est réalisée par une tâche d'écriture, qui permet d'explorer les fonctions liées aux tâches d'acquisition.

Dans la troisième, le traitement est scindé en deux parties par une structure « séquence » (structure dessinée sous l'aspect « film » dans LabView), avec une phase acquisition et une phase traitement qui permet de prendre une décision pour la commande du moteur du store.

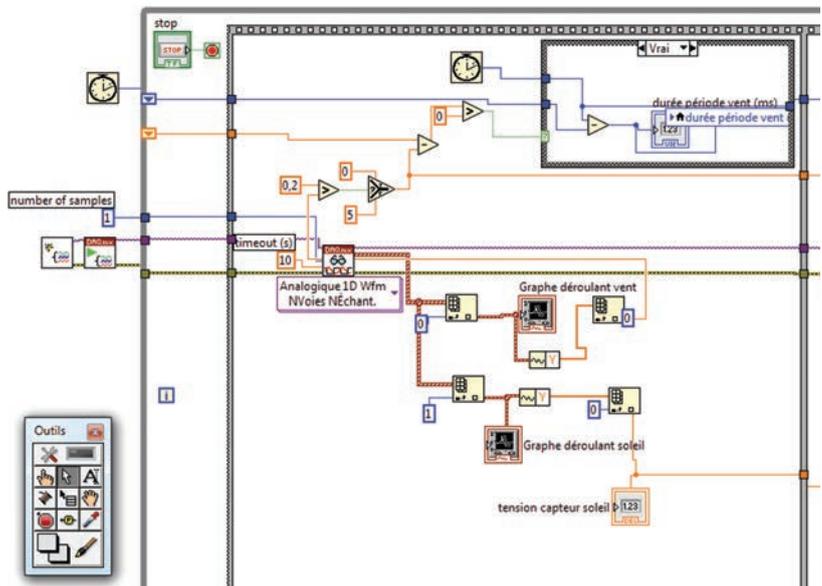
Enfin, la version finale **8** améliore le comportement avec la mise en place de rétroactions afin de préserver le système en cas de changements météorologiques trop rapides.

● **Résolution logique**

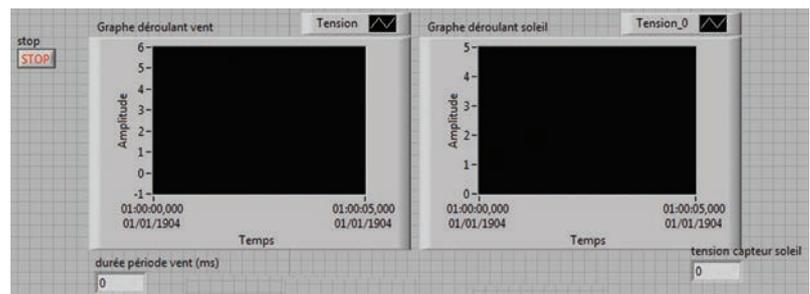
Soit

PV : présence vent, booléen vrai (niveau logique 1) si la vitesse du vent acquise est supérieure au seuil réglé

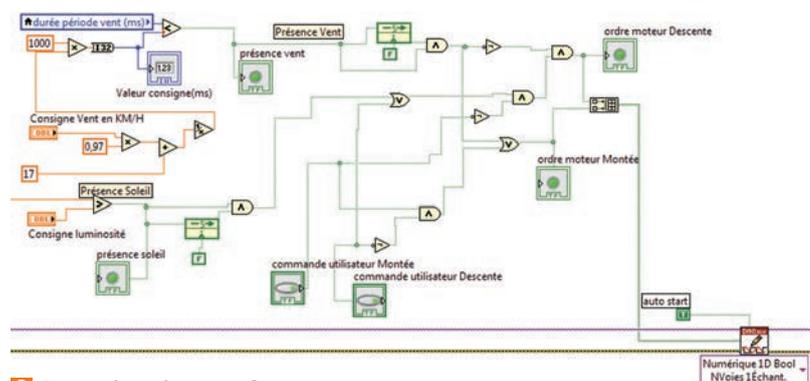
PS : présence soleil, booléen vrai si l'intensité lumineuse du soleil est supérieure au seuil réglé



**6** Le programme sous forme de diagramme de la partie acquisition



**7** La face avant de la partie acquisition



**8** La partie traitement du programme



**9** La partie traitement de l'IHM

CUM : commande utilisateur de montée du store  
 CUD : commande utilisateur de descente du store  
 CdeMOTM : commande moteur permettant une montée du store  
 CdeMOTD : commande moteur permettant une descente du store

Le store remonte sa toile s'il y a du vent ou une commande de montée de la part de l'utilisateur, et la descend s'il y a du soleil ou une commande de descente, mais pas de vent, destructeur de la toile.

Nous obtenons les équations logiques suivantes :

$$CdeMOTM = PV + (CUM \cdot CUD)$$

$$CdeMOTD = \overline{PV} \cdot \overline{CM} \cdot (PS + CUD)$$

L'IHM permet d'une part à l'utilisateur de commander le store par boutons (CUM et CUD), et d'autre part d'offrir le mode « réglage » réalisé par des boutons rotatifs [9].

La partie commande « pure » est réalisée en logique à partir des équations précédentes comportant les boutons et les résultats des comparaisons des seuils et des valeurs réelles issues des capteurs.

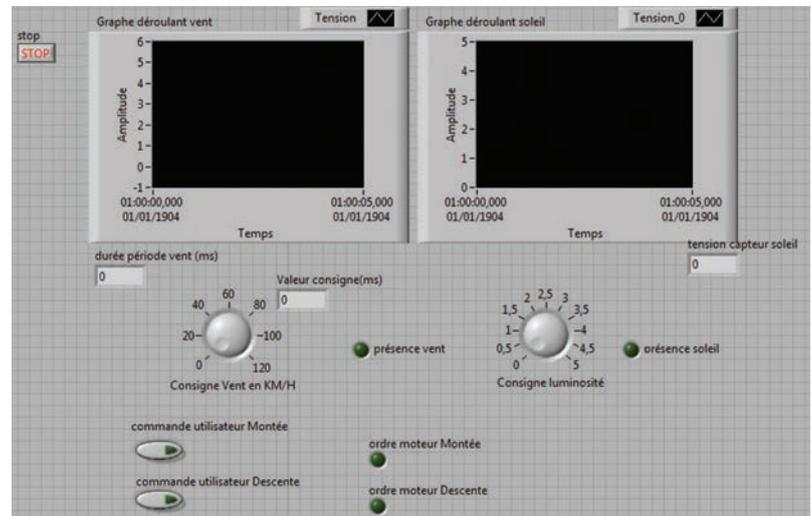
L'IHM de l'application complète reprend les affichages de l'acquisition, ainsi que les commandes utilisateur [10].

Le programme complet présente donc ici des structures en programmation (itérations « tant que », structure conditionnelle, structure séquentielle, etc.) [11].

### Introduction au diagramme état/transition de SysML

*Matériel utilisé :* un ordinateur avec LabView et une carte USB-6212

*Objectif :* acquérir les bases d'un contrôle réalisé sous forme d'un diagramme état/transition.



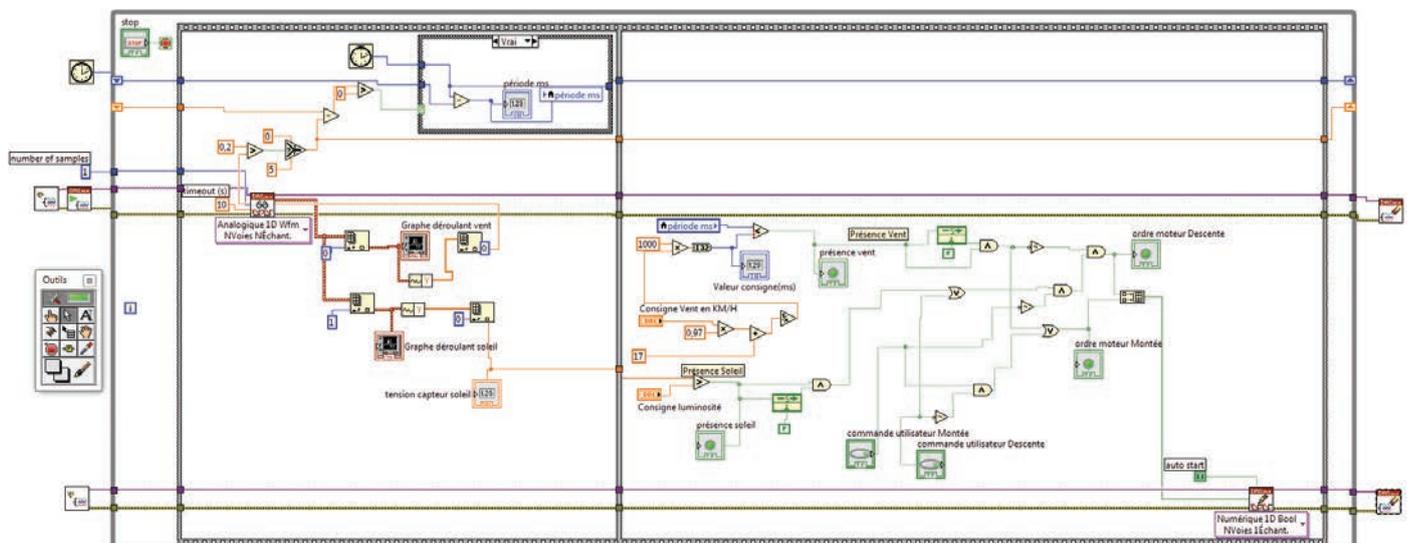
10 L'IHM de l'application complète

La troisième étape consiste à mettre en œuvre le traitement par diagramme état/transition à l'aide du *toolkit* Statechart (version 1.3), qui permet de programmer les machines à états avec une représentation proche de la norme SysML.

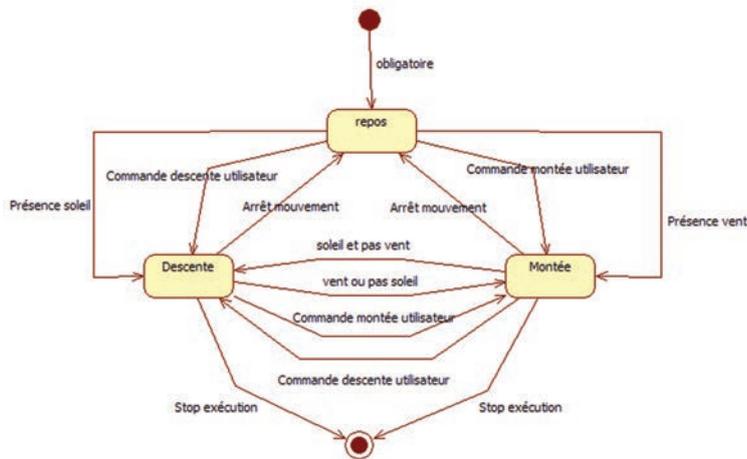
Deux versions logicielles se succèdent à cette étape. La première effectue un traitement par événements des boutons proposés à l'utilisateur, ce qui permet de découvrir le principe du traitement des événements. Dans la seconde, le traitement des informations des capteurs de vent et de soleil est inclus dans les états et transitions [12].

À l'aide de LabView et de Statechart, on crée le modèle comportemental sous la forme d'un diagramme état/transition [13].

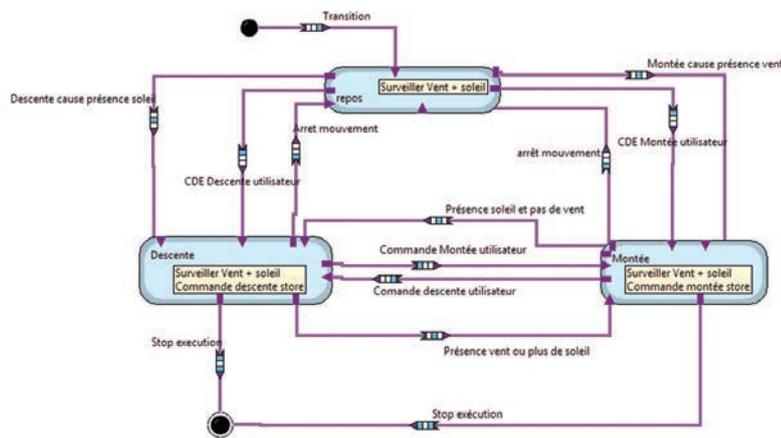
Nous pouvons observer les similitudes entre le diagramme permettant une description à l'aide de SysML et celui réalisé avec Statechart, qui permet



11 Le programme de l'application complète



12 Le modèle comportemental partiel du store (diagramme état/transition)



13 Le diagramme état/transition du projet LabView avec Statechart

la programmation du comportement. En effet, LabView permet :

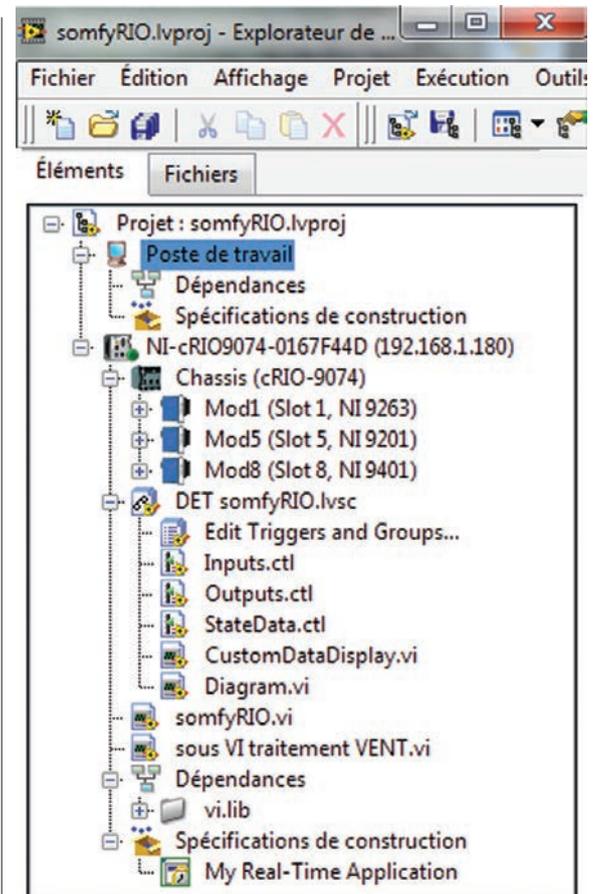
- dans les transitions, la prise en compte des événements (limités aux événements Windows avec la carte USB) avec leur condition de garde et une action possible ;
- dans les états, la programmation d'actions en *Entry*, *Do* et *Exit* ;
- l'encapsulation des états ;
- la simulation des diagrammes.

**Mise en œuvre du CompactRIO**

*Matériel utilisé* : un ordinateur avec LabView et un CompactRIO NI cRIO-9074, deux modules d'E/S TOR NI 9401 et un module d'entrées analogiques NI 9201

*Objectif* : réaliser le contrôle-commande sur un matériel autonome à l'aide d'un modèle état/transition.

La description du comportement étant fournie sous forme d'un diagramme état/transition SysML, on implante le programme dans le contrôleur CompactRIO NI cRIO-9074 équipé de deux modules d'E/S TOR NI 9401 et d'un module d'entrées analogiques NI 9201. Cela nécessite une modification du diagramme état/transition et du VI (non pas dans le concept, mais dans les états et



14 Le projet LabView

transitions, car le nommage des E/S du CompactRIO est différent de celui de la carte USB utilisée jusqu'ici) pour un fonctionnement en autonomie dans la cible.

Les avantages de travailler sur cette cible sont la prise en compte des événements sur toutes les entrées de façon asynchrone et l'aspect « temps réel » de son exécution.

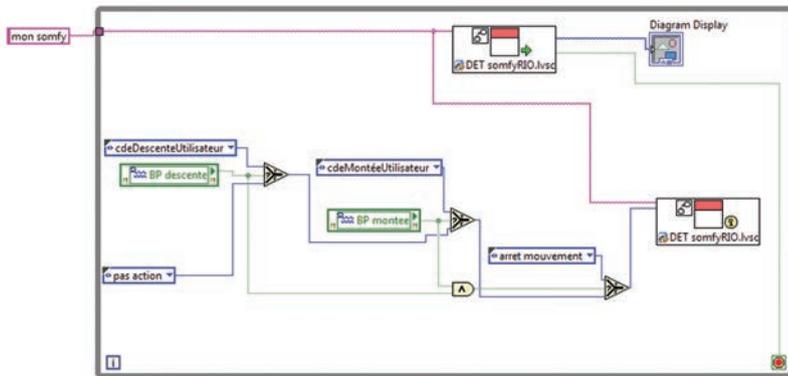
Il est nécessaire de créer un projet LabView 14 avec lequel nous allons déployer notre programme 15 et le comportement voulu : diagramme état/transition 13 dans la cible par connexion Ethernet.

Au final, notre application fonctionne en autonomie dans le CompactRIO. L'IHM obtenue permet de prendre en compte les informations utilisateur (comme précédemment) et de montrer quel état est actif dans une zone appelée *diagram display* 16.

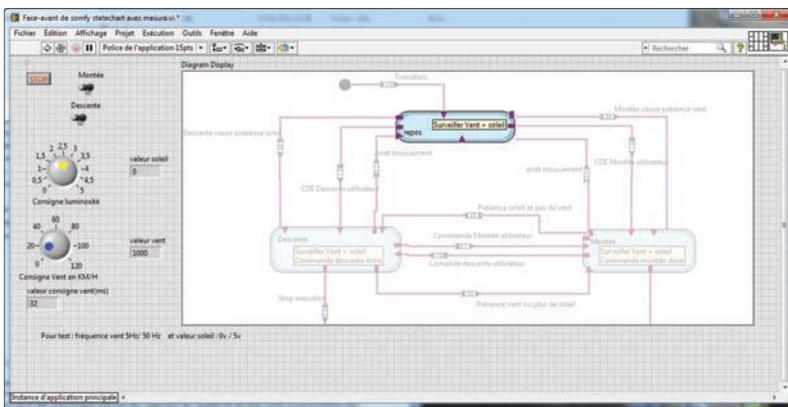
**Une solution évolutive**

National Instruments nous a permis, à nous formateurs académiques, de suivre des stages de développement LabView. Nous avons pu ainsi étendre nos connaissances du logiciel et réaliser les synthèses de nos propres travaux d'autoformation.

Pour les professeurs de l'académie de Dijon, qui, au terme de trois jours de formation, deviennent « référents LabView », cette application sert de synthèse de connaissances sur LabView. Elle permet aussi de proposer à nos collègues une solution logicielle et



15 Le programme « SomfyRIO.VI »



16 La face avant du programme

**En ligne**

Après création d'un compte, l'application complète est téléchargeable sur le site de National Instruments.

Présentation :

<http://sine.ni.com/cs/app/doc/p/id/cs-15747>

L'application complète en téléchargement :

<https://decibel.ni.com/content/docs/DOC-25902>

CompactRIO :

[www.ni.com/compactrio/whatis/f/](http://www.ni.com/compactrio/whatis/f/)

Retrouvez tous les liens sur <http://eduscol.education.fr/sti/revue-technologie>

matérielle à des problématiques autant techniques que pédagogiques, par exemple lors de projets de fin d'année aussi bien en terminale S-SI et STI2D qu'en BTS. Avec le *toolkit* LabView Statechart, associé à un affichage graphique de l'état actif du diagramme, nos élèves parviendront mieux à comprendre le comportement du système, essentiel dans notre pédagogie. Au cours de la formation, un temps de réflexion d'ailleurs été consacré aux usages de la plate-forme dans notre pédagogie tant en cours qu'en phase de projet.

Enfin, cette solution a pour avantage principal de permettre des évolutions de comportement. L'application peut évoluer dans la communication et la domotique en recevant ses ordres d'un superviseur, par exemple en communication TCP/IP. ■

[PUBLICITÉ]

# technologie s'affiche sur Éduscol...



*Vous y trouverez :*

- Le sommaire détaillé de chaque nouveau numéro
- Des liens vers de nombreux articles
- Un lien vers les archives de la revue

*Vous pourrez y télécharger :*

- Des articles d'archives de la revue
- L'éditorial et le Technomag de chaque numéro

... mettez-le dans vos favoris !



<http://eduscol.education.fr/sti/revue-technologie>