



Sciences et technologies

de l'Industrie et du développement durable

ET 24 : Modèle de comportement d'un système

Piloter un modèle Solidworks avec Labview softmotion.

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 piloter un modèle solidworks à partir d'un VI dans Labview.	
Pré-requis : Les bases de Labview - Le VI – Solidworks Motion	
Bases théoriques : Interfacer Labview softmotion avec solidworks motion	
Outil : Labview softmotion – Solidworks motion	
Support : Maquette numérique d'un robot, d'un vérin à vis.	
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	Introduction.....	3
2	Pilotage d'un moteur en vitesse.....	4
2.1	Création du projet et import des moteurs de solidworks motion.....	4
2.2	Création d'un axe dans labview.	4
2.3	Création du VI de commande dans Labview	6
2.3.1	Réalisation du diagramme (Programmation).....	6
2.3.2	Réalisation de la face avant (Interface utilisateur).....	8
2.4	Exécution du VI et simulation virtuelle du mouvement sous SW motion	9
2.5	Application, pilotage des deux moteurs du robot.....	10
2.5.1	Commande du 2 ^{ème} moteur	10
2.5.2	Simplification du diagramme et de la face avant.....	11
2.6	Ajout d'une commande d'arrêt des moteurs	12
3	Pilotage d'un moteur en position.....	13
3.1	Test du VI de commande du pousse seringue	13
3.2	Description du diagramme	14
3.2.1	1 ^{ère} observation	14
3.2.2	2 ^{ème} observation	14
4	Lecture de données (vitesses, position, couple...) : Fonction « READ »	15
4.1	Affichage de la vitesse et position du piston de la seringue, ainsi que du débit	15
4.2	Sauvegarde des données	16
5	Implantation d'un capteur	17
5.1	Création d'un capteur dans solidworks.....	17
5.1.1	Création de la cote	17
5.1.2	Récupération est utilisation de l'état du capteur dans Labview.....	18



Sciences et technologies de l'Industrie et du développement durable

1 Introduction.

Le logiciel de programmation Labview possède de nombreux modules. L'un de ces modules : « Softmotion », permet de communiquer avec solidworks, et ainsi piloter une maquette numérique.

Il est donc possible de prototyper et simuler une application de contrôle d'axes sur une maquette numérique avant d'acheter les éléments physiques.

La programmation graphique de haut niveau offre de nombreuses fonctionnalités intéressantes telles que :

- Contrôle et commande d'axes en vitesse, position, accélération.
- Suivi de trajectoires.
- Détection de collision.
- ...

Nous allons voir à travers quelques exemples certaines de ces fonctionnalités.



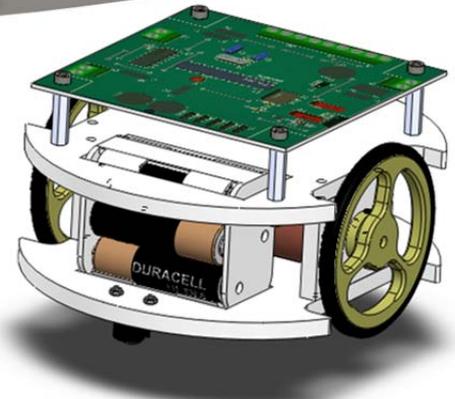
Sciences et technologies de l'Industrie et du développement durable

2 Pilotage d'un moteur en vitesse

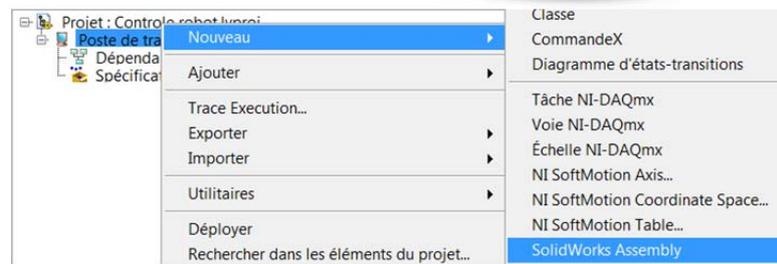
2.1 Création du projet et import des moteurs de solidworks motion.

Lancez le logiciel Labview, créez un nouveau projet intitulé « Controle robot.lvproj ».

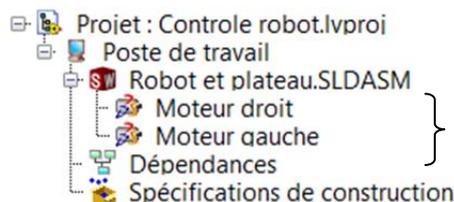
Lancez le logiciel solidworks, et ouvrez le fichier « Robot et plateau.SLDASM »



Dans le projet Labview, cliquez bouton droit sur *Poste de travail* puis *Nouveau / Solidworks Assembly*. L'assemblage précédemment ouvert dans solidworks est détecté, et Labview se propose d'importer les moteurs de cet assemblage. Validez par OK.



Remarquez que l'assemblage solidworks apparait maintenant dans le projet avec les deux moteurs.



2.2 Création d'un axe dans labview.

Vous allez maintenant créer un axe (axis) dans le projet Labview. Vous attribuerez cet axe au moteur gauche du robot dans solidworks motion, et ainsi le lien sera réalisé entre Labview et Solidworks.

Pour cela, dans le projet Labview, cliquez bouton droit sur *Poste de travail*, puis *Nouveau / NI SoftMotion Axis...*



Une fenêtre « Axis Manager » apparait. Cliquez sur *add New Axis* pour créer un axe.

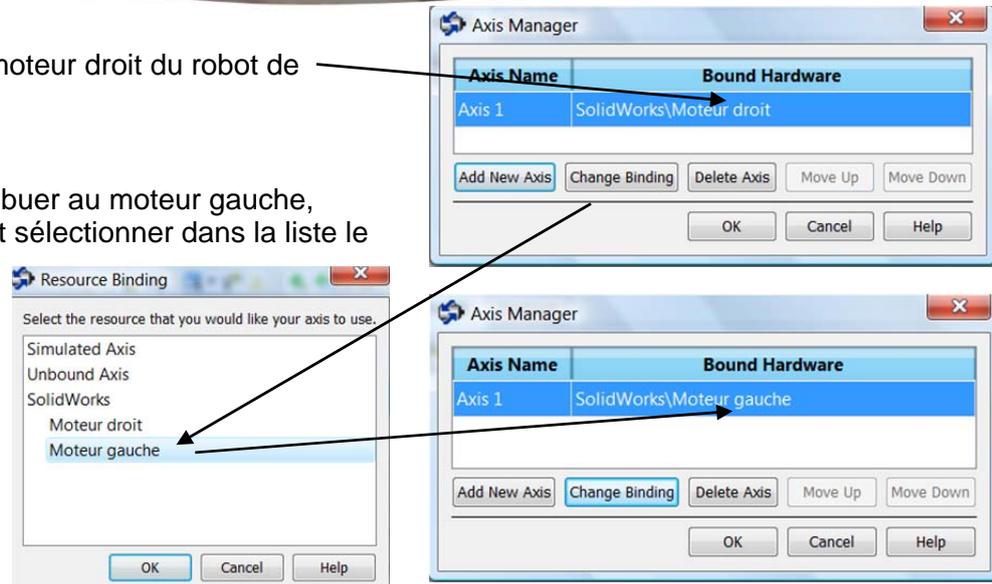




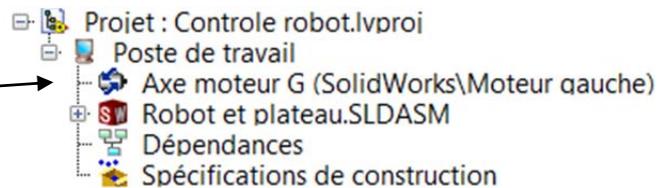
Sciences et technologies de l'Industrie et du développement durable

Par défaut cet axe est attribué au moteur droit du robot de solidworks motion.

Dans notre cas, nous voulons l'attribuer au moteur gauche, cliquez donc sur *Change Binding* et sélectionner dans la liste le moteur gauche du robot. Validez deux fois par OK.

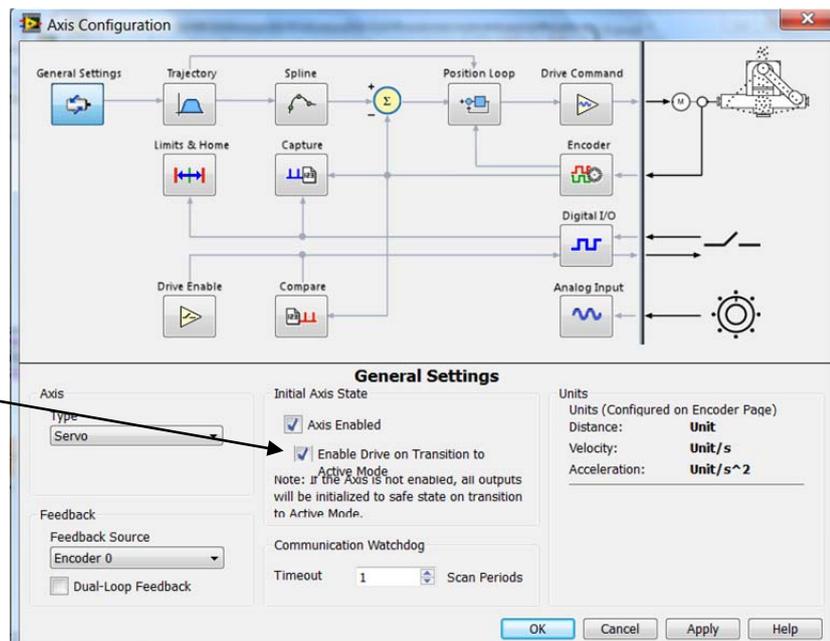


Un axe intitulé « Axis 1 » Apparaît dans le projet. Renommez le en « Axe moteur G »



Afin de simplifier notre exemple, vous allez dans les propriétés de l'axe cocher la case « Enable drive on transition to active mode ». Cela évite de rendre opérationnel (enable) l'axe par une fonction supplémentaire appelée Power dans le VI que vous allez créer par la suite. Cliquez bouton droit dans le projet Labview sur « Axe moteur G » puis *Propriétés*, et cocher cette case « Enable drive on transition to active mode ».

Au passage, dans les propriétés de l'axe, vous voyez qu'il est possible de modifier énormément de paramètres.





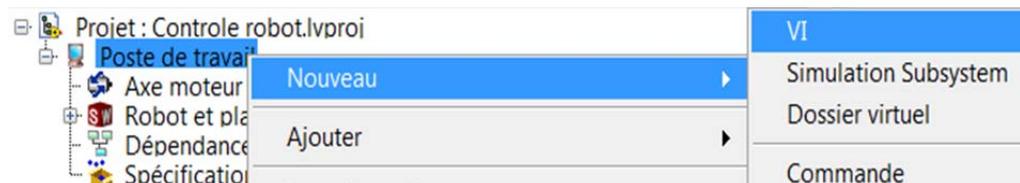
Sciences et technologies de l'Industrie et du développement durable

2.3 Création du VI de commande dans Labview

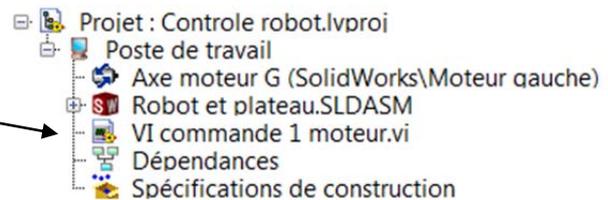
2.3.1 Réalisation du diagramme (Programmation)

Vous allez maintenant créer l'application permettant de piloter le moteur gauche du robot.

Pour cela cliquez bouton droit sur *Poste de travail* puis *Nouveau / VI*. Une face avant et un diagramme apparaissent.



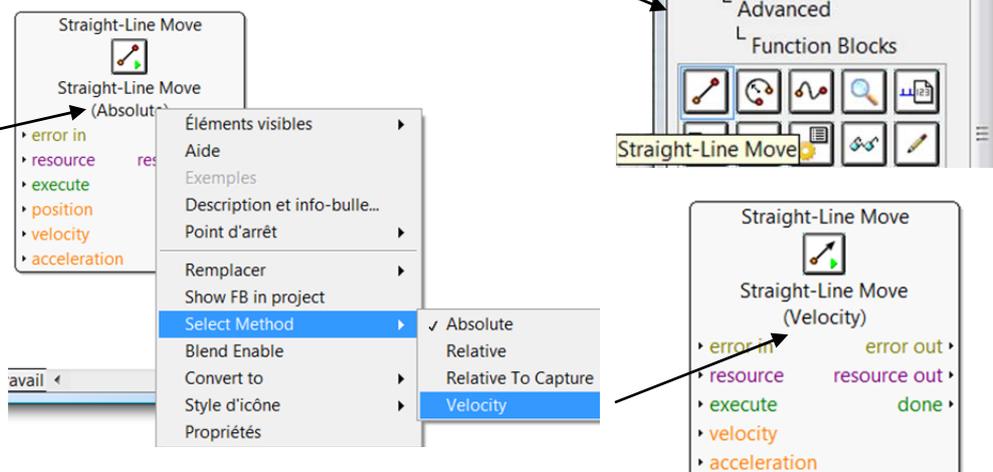
Enregistrez ce VI sous le nom « VI commande 1 moteur.vi ».



Réalisation de l'application de commande dans le VI

Vous allez dans un premier temps commander en vitesse et accélération le moteur gauche du robot. Insérez pour cela une commande « **Straight-Line Move** » dans le diagramme du VI, vous la trouverez dans la palette de fonctions à *Vision et Motion / SoftMotion / Advanced / Function Blocks / Straight-Line Move*.

Par défaut la fonction Straight-Line Move est configurée en Déplacement absolu « Absolute », or nous souhaitons piloter uniquement en vitesse et accélération le moteur gauche du robot et non en position. Cliquez bouton droit sur le bloc fonction Straight-Line Move puis *Select Method / Velocity*.



Ce bloc fonction Straight-Line Move comporte les entrées et sorties suivantes :

- **error in / error out** : entrée et sortie qui permettent de gérer les messages d'erreurs qui pourraient survenir durant l'exécution de l'application. Nous ne les utiliserons pas par la suite.
- **ressource** : Entrée sur laquelle sera connecté l'axe à commander, dans notre cas « Axe moteur G ».
- **ressource out** : Sortie permettant de connecter en série d'autres blocs fonction (Power, Stop Move,...) agissant sur le même axe que celui connecté sur ressource.

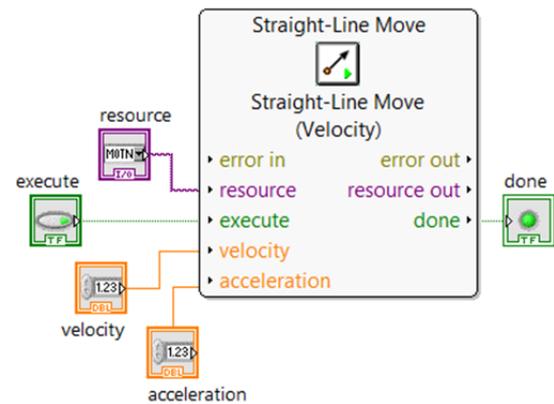
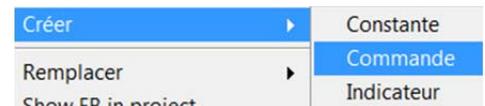


Sciences et technologies de l'Industrie et du développement durable

- **execute** : Entrée booléenne qui lorsqu'elle est à 1 exécute le bloc Straight-Line Move, c'est-à-dire accélère l'axe connecté sur ressource jusqu'à la vitesse désirée.
- **velocity** : Vitesse souhaitée en **degrés/s** de l'axe piloté connecté sur ressource.
- **acceleration** : Accélération en **degrés/s²** pour atteindre la vitesse souhaitée.
- **done** : Sortie booléenne qui est à 1 lorsque la vitesse souhaitée de l'axe est atteinte.

Connectez une par une ces entrées et sorties, pour cela :

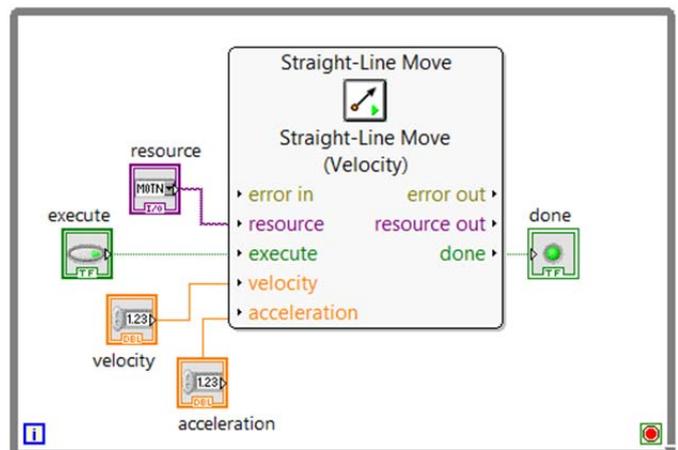
- Cliquez bouton droit sur l'entrée ressource (lorsque la bobine apparaît), puis *Créer / Commande*.
- Cliquez bouton droit sur l'entrée execute (lorsque la bobine apparaît), puis *Créer / Commande*.
De la même façon :
- Créez une commande pour la vitesse et l'accélération.
- Créez un indicateur sur la sortie done.



Vous devriez obtenir un diagramme ressemblant à ça :

Maintenant vous allez insérer l'ensemble de votre programmation (tout le contenu de diagramme) dans une boucle While. Pour cela sélectionnez dans la palette de fonction à *Programmation / Structures* la boucle While, et entourer le contenu du diagramme.

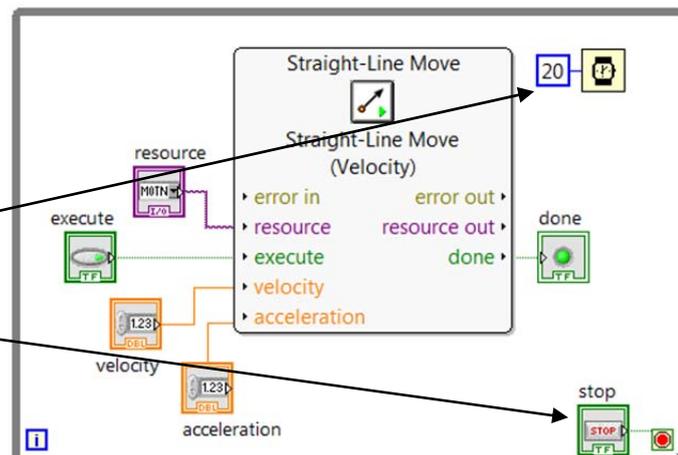
Vous devez obtenir ceci :



Vous allez maintenant cadencer cette boucle pour quelle soit parcourue toutes les 20 ms. Insérez n'importe où dans la boucle While précédemment créée la fonction

« attendre »  présente dans la palette à *Programmation / Informations temporelles*. Créer une constante en entrée (bouton droit puis *créer / constante*) et tapez 20 dans le cadre.

Créez une commande « Stop » afin de sortir de la boucle While. Pour cela clic droit sur le rond rouge (condition de boucle) puis *Créer / Commande*.





Sciences et technologies de l'Industrie et du développement durable

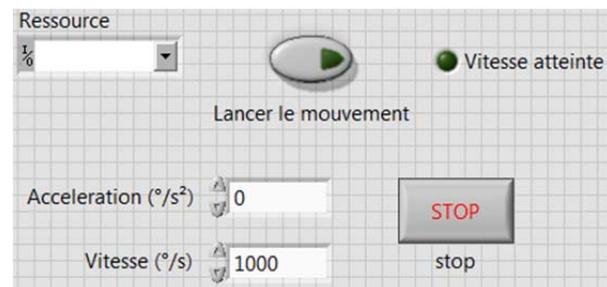
2.3.2 Réalisation de la face avant (Interface utilisateur)

Vous allez maintenant réorganiser la face avant afin qu'elle soit plus lisible.

Afficher la fenêtre windows dans laquelle se trouve la face avant du VI, les différentes commandes et indicateurs insérés dans le diagramme apparaissent mais sont mal positionnés.

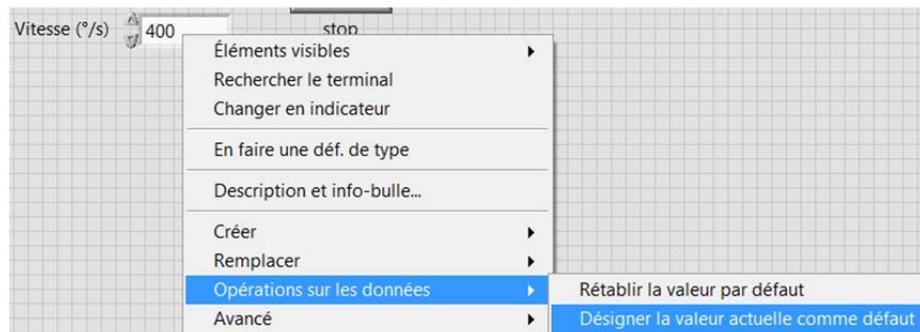


Réorganisez les afin d'y voir plus clair et renommer les différentes étiquettes de la sorte :

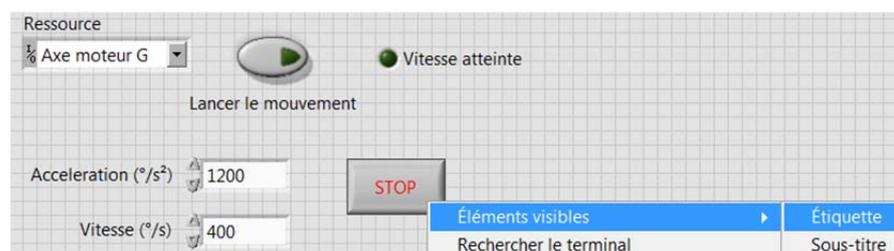


Lorsque vous exécuterez le VI, l'accélération sera de 0 °/s², et la vitesse maxi de 1000 °/s (jamais atteinte du fait de l'accélération nulle). De même la ressource, c'est-à-dire l'axe commandé en vitesse par le VI, n'est pas présélectionné dans le menu déroulant. Vous allez établir des paramètres réalistes qui seront utilisés par défaut dès l'exécution du VI. Pour cela :

- Tapez 400 (°/s) dans la fenêtre de saisie de la vitesse, cliquez bouton droit sur cette fenêtre puis *Opération sur les données / Désigner la valeur actuelle par défaut*.
- De même, saisissez une accélération de 1200 (°/s²), puis clic droit sur la fenêtre de saisie de l'accélération et *Opération sur les données / Désigner la valeur actuelle par défaut*.
- De la même façon, sélectionner dans le menu déroulant la ressource « Axe moteur G », puis clic droit sur cette fenêtre et *Opération sur les données / Désigner la valeur actuelle par défaut*.



Certaines étiquettes peuvent être superflues comme par exemple celle du bouton stop. Pour ne plus les afficher : cliquez bouton droit sur le bouton stop, puis *Éléments visibles / étiquette*, afin de la décocher.





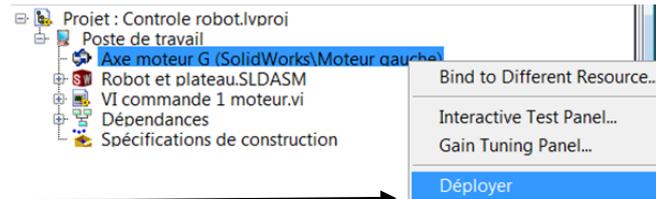
Sciences et technologies de l'Industrie et du développement durable

2.4 Exécution du VI et simulation virtuelle du mouvement sous SW motion

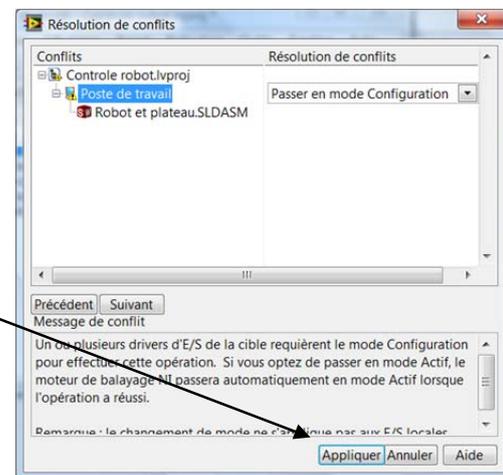
A présent, tout est fin prêt pour piloter, à l'aide votre application (VI), la maquette virtuelle sous solidworks motion (enfin... 😊).

Dernières étapes avant de voir s'animer la maquette :

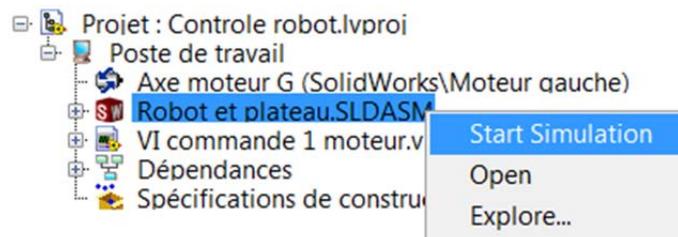
- Vous devez déployer l'axe Labview dans solidworks motion, cliquez pour cela bouton droit sur « Axe moteur G » dans la fenêtre du projet Labview, puis *Déployer*.



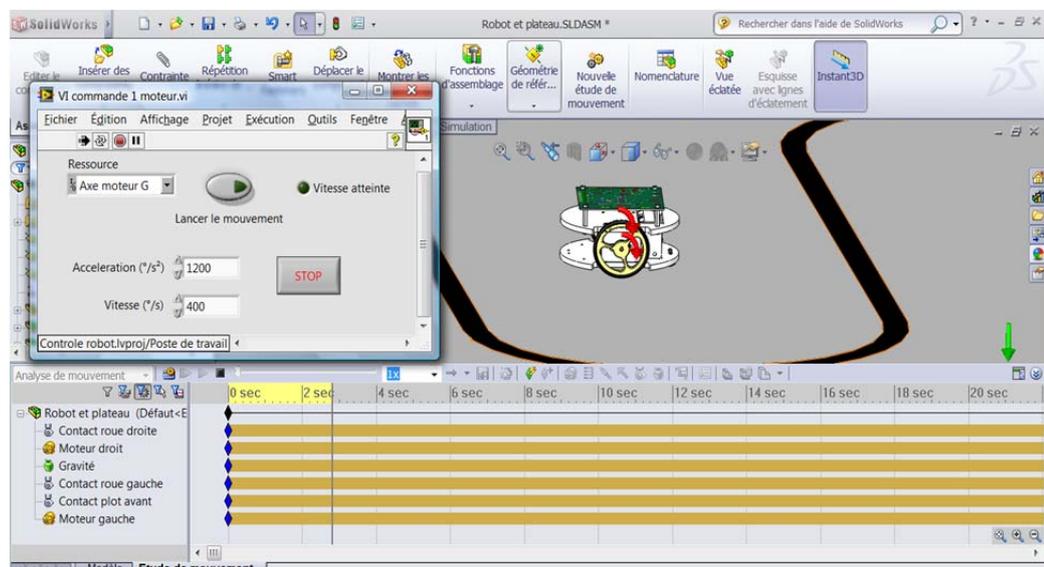
Une fenêtre apparaît et propose de « Passer en mode configuration puis en mode actif » afin de déployer l'Axe moteur G. Cliquez sur appliquer.



- Lancez la simulation (le moteur de calcul) dans solidworks motion en cliquant bouton droit sur « Robot et plateau.SLDASM » dans la fenêtre de projet Labview, puis *Start simulation*. Si vous revenez dans la fenêtre Solidworks, vous vous apercevrez que le moteur de calcul à démarrer et que le temps s'écoule.



- Affichez la face avant de votre VI puis exécutez le en cliquant sur . Essayez d'avoir à l'écran votre face avant et la maquette solidworks en manipulant les fenêtres windows. Pour l'instant rien ne se passe, la maquette ne s'anime pas, c'est normal...





Sciences et technologies de l'Industrie et du développement durable

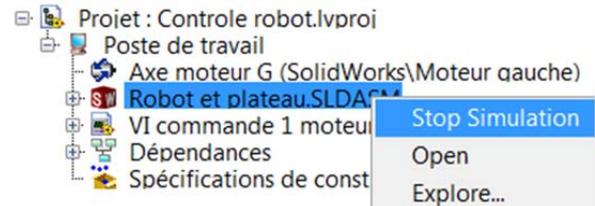
- Enfin appuyez sur le bouton « Lancer le mouvement » de votre face avant, et OBSERVEZ LE ROBOT SOUS SOLIDWORKS ACCÉLÉRER ET SE METTRE A TOURNER EN ROND.

Quel beau spectacle !



Lorsque vous souhaitez interrompre le calcul de l'animation, faites en cliquant bouton droit sur « Robot et plateau.SLDASM » puis *Stop Simulation*.

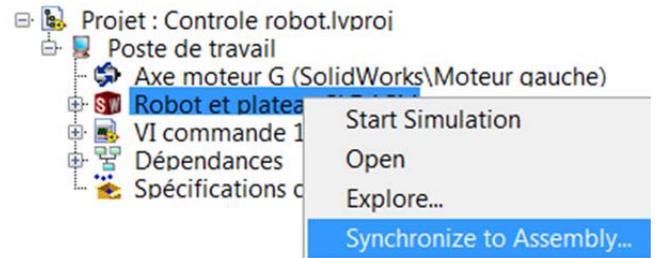
Si vous arrêtez la simulation directement dans motion, il vous faudra quitter votre projet Labview et le ré-ouvrir.



Vous pouvez ensuite revisionner l'animation en cliquant sur lecture  dans Solidworks motion.

Bon à savoir :

Si entre deux simulations vous modifiez votre modèle solidworks (rapport de transmission dans une liaison, position d'un moteur, ...), il vous faut mettre à jour le projet Labview en cliquant bouton droit sur « Robot et plateau.SLDASM » dans la fenêtre de projet puis *Synchronize to Assembly...*



De temps à autres des bugs informatiques nécessitent de fermer le projet Labview et de le ré-ouvrir afin de pouvoir communiquer avec solidworks motion et l'exécuter.

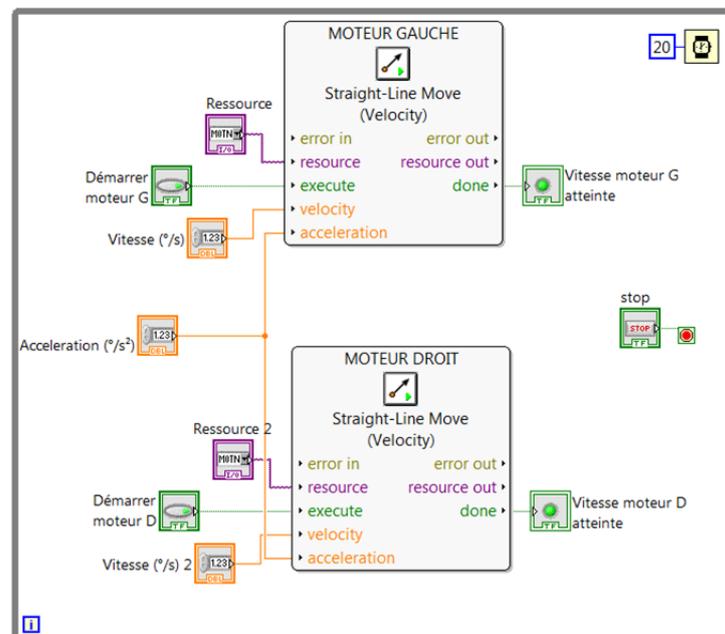
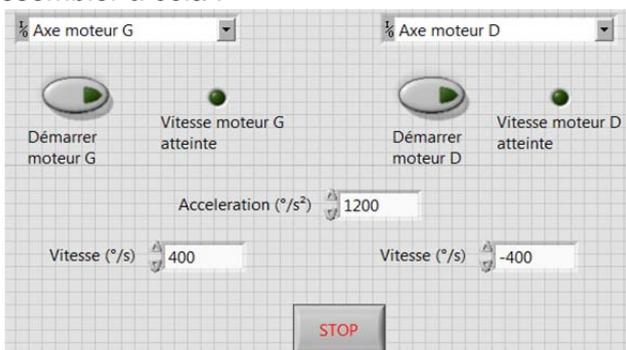
2.5 Application, pilotage des deux moteurs du robot

2.5.1 Commande du 2^{ème} moteur

En réitérant les opérations depuis le §2.2 :

- Commandez le 2^{ème} moteur (moteur droit) du robot en imposant par défaut une accélération de 1200 °/s², et une vitesse de -400 °/s.
- Testez votre programme en simulant le mouvement dans solidworks motion sur une durée de 3s.

Le diagramme et la face avant du VI devraient ressembler à cela :





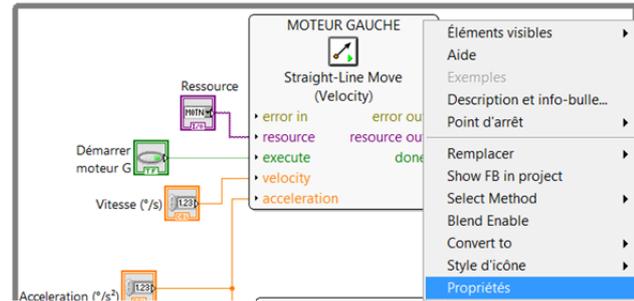
Sciences et technologies de l'Industrie et du développement durable

2.5.2 Simplification du diagramme et de la face avant

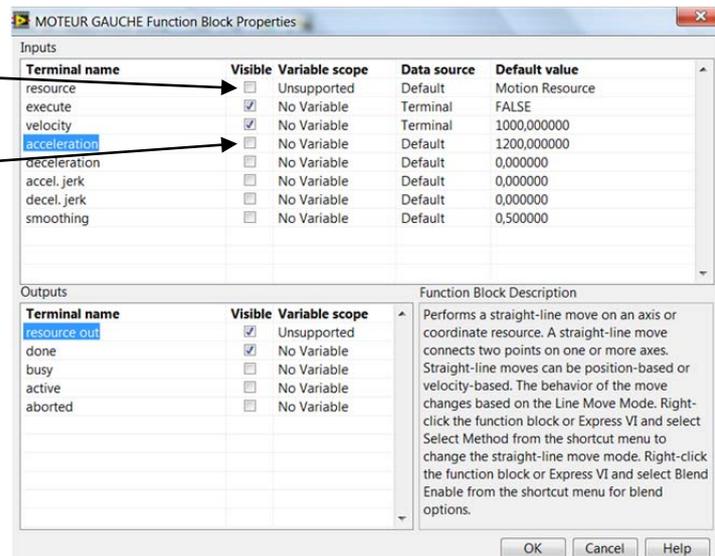
L'accélération est liée à la technologie des moteurs, à la masse du robot... Si l'on ne souhaite pas la faire varier, il est possible de la définir par défaut dans les fonctions « STRAIGHT-LINE MOVE » au sein du diagramme.

De même, la ressource (« Axe moteur G », ou « Axe moteur D ») en entrée de ces fonctions « STRAIGHT-LINE MOVE » n'est pas amenée à changer dans notre exemple, vous allez la définir par défaut afin qu'elle n'apparaisse plus dans la face avant.

Pour cela, cliquez bouton droit sur la fonction « STRAIGHT-LINE MOVE » du moteur Gauche dans le diagramme et



« Propriétés », puis dans la fenêtre qui apparaît:

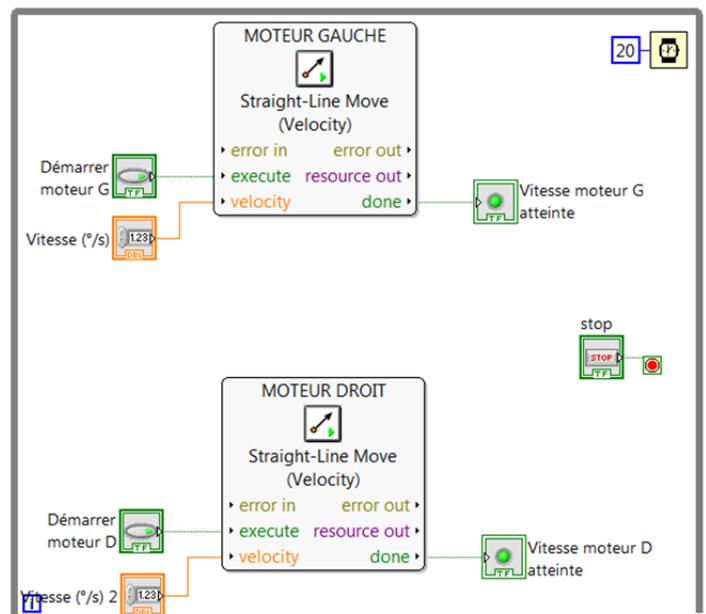
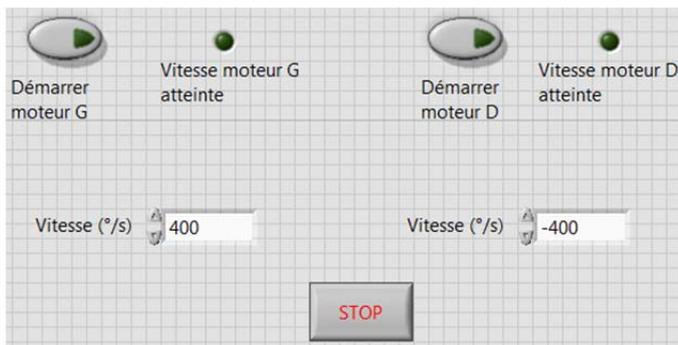


- Décochez « resource » et dans la colonne « Default value » choisissez « Axe moteur G ».
- Décochez « acceleration » et dans la colonne « Default value » saisissez 1200.

Validez par « OK »

Faites de même pour la fonction « STRAIGHT-LINE MOVE » du moteur Droit, puis effacer toutes les connections superflues du diagramme (ctrl + B), ainsi que les commandes qui sont désormais inutiles (« Accélération (°/s²) », « Ressource », et « Ressource 2 »)

Le diagramme et la face avant sont ainsi simplifiés et doivent ressembler à cela :





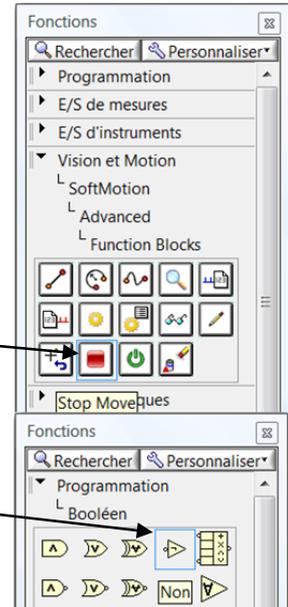
Sciences et technologies de l'Industrie et du développement durable

2.6 Ajout d'une commande d'arrêt des moteurs

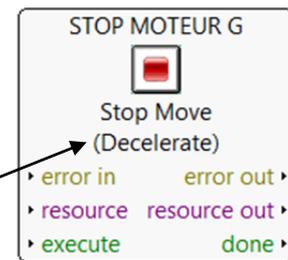
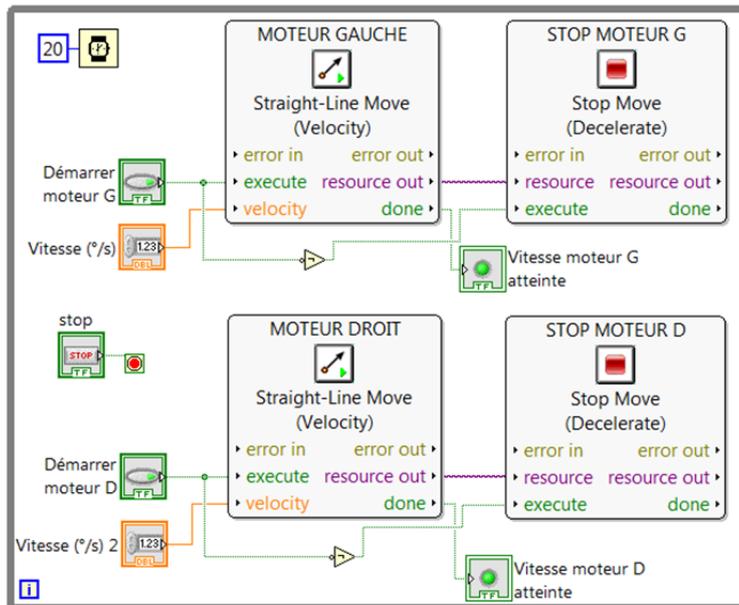
Vous allez maintenant insérer dans le diagramme de pilotage du robot, 2 commandes « Stop Move » qui permettront de stopper les moteurs, et de les relancer au cours de la simulation.

Pour cela insérer dans le diagramme :

- Deux commandes « Stop Move » présentes dans la palettes des fonctions à *Vision et Motion / SoftMotion / Advanced / Function Blocks / Stop Move*. Renommez les en « STOP MOTEUR G » et « STOP MOTEUR D ».
- Deux fonctions booléennes « Non » présentes dans la palettes des fonctions à *Programmation / Booléen / Non*.

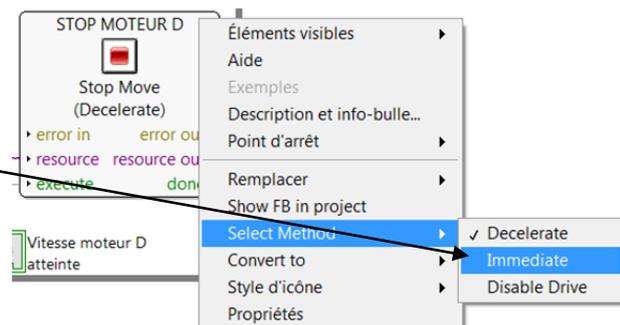


Connectez ces fonctions dans le diagramme de la façon suivante :



Vous remarquerez que les deux fonctions « Stop Move » sont configurées par défaut en « Decelerate ». C'est à dire que le moteur ne s'arrêtera pas immédiatement dès que la fonction « Stop Move » sera exécutée, il sera décélééré jusqu'à atteindre une vitesse nulle. Vous allez changer cette méthode d'arrêt par défaut en imposant un arrêt immédiat.

Pour cela cliquez bouton droit sur les fonctions « Stop Move » puis *Select method / Immediate*.



Testez maintenant votre application en simulant le déplacement du robot, vous pouvez à votre gré stopper puis redémarrer les moteurs, et ce en modifiant les vitesses si vous le souhaitez.

Génial... 😊



Sciences et technologies de l'Industrie et du développement durable

3 Pilotage d'un moteur en position

Vous allez à présent commander un moteur en position dans Solidworks Motion, à partir de Labview. L'exemple choisi est un pousse seringue, pour lequel vous saisissez sur la face avant du VI le volume de médicament à injecter, et la maquette solidworks s'exécutera.

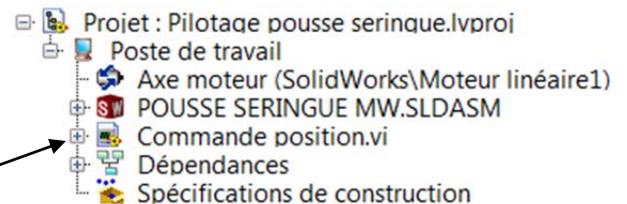
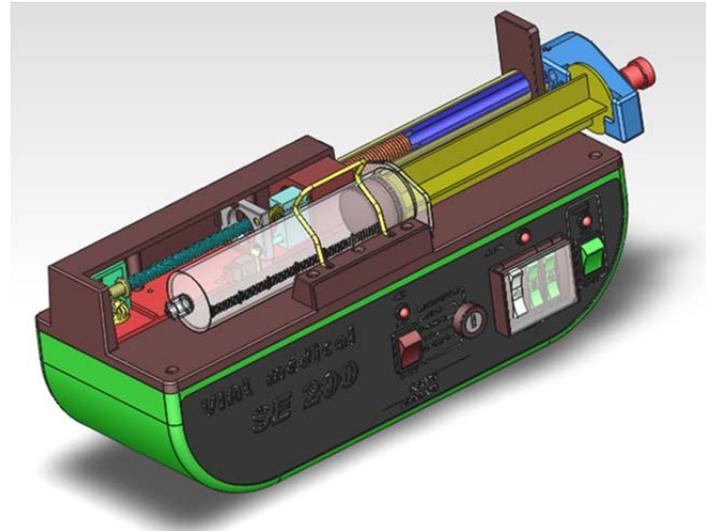
Afin de gagner du temps, le modèle solidworks Motion, et le projet Labview sont donnés.

3.1 Test du VI de commande du pousse seringue

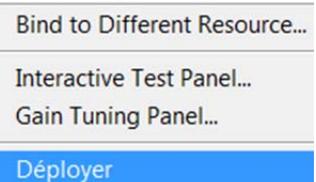
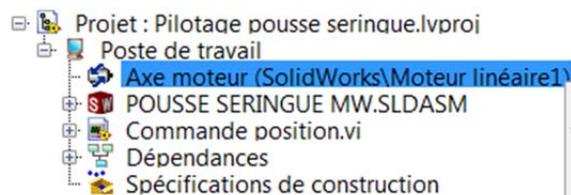
Quittez tous les documents ouverts dans solidworks, ainsi que dans Labview.

Ouvrez le fichier assemblage dans solidworks intitulé « POUSSE SERINGUE MW.SLDASM »

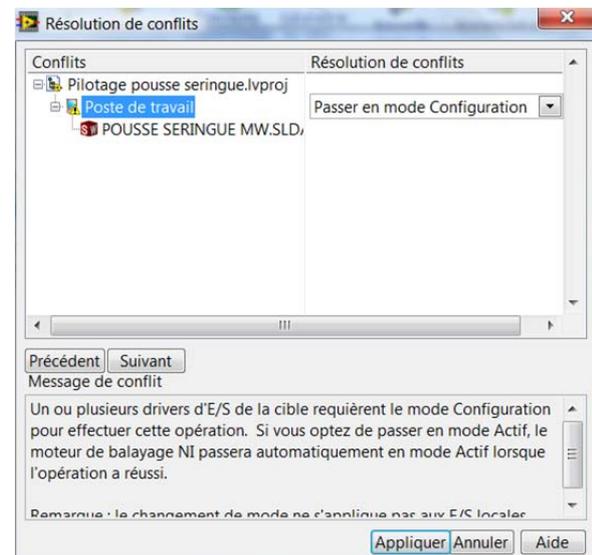
Ouvrez le projet Labview intitulé « Pilotage pousse seringue.lvproj », et double cliquez dans l'arbre du projet sur le VI nommé « Commande position.vi » afin de l'ouvrir.



Déployez l'axe moteur de Labview vers solidworks en cliquant bouton droit dans l'arbre du projet Labview sur « Axe moteur » puis *Déployer*.



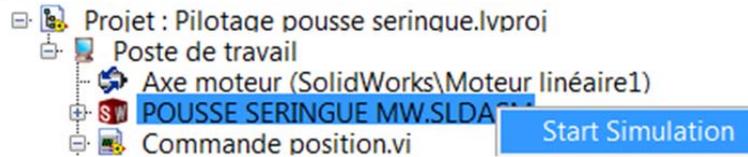
Validez la fenêtre qui apparait en cliquant sur « appliquer ».





Sciences et technologies de l'Industrie et du développement durable

Lancez le début de la simulation et du calcul dans solidworks motion en cliquant bouton droit dans l'arbre du projet Labview sur « POUSSE SERINGUE MW.SLDASM » puis *Start Simulation*.



Exécutez le VI , et injectez un volume compris entre 0 et 60 cm³. **Génial ça marche...** 😊

Pour ramener le piston de la seringue au point de départ, vous pouvez taper une position négative, ou Stopper la simulation et la relancer (Clic droit sur « POUSSE SERINGUE MW.SLDASM » dans le projet Labview).

3.2 Description du diagramme

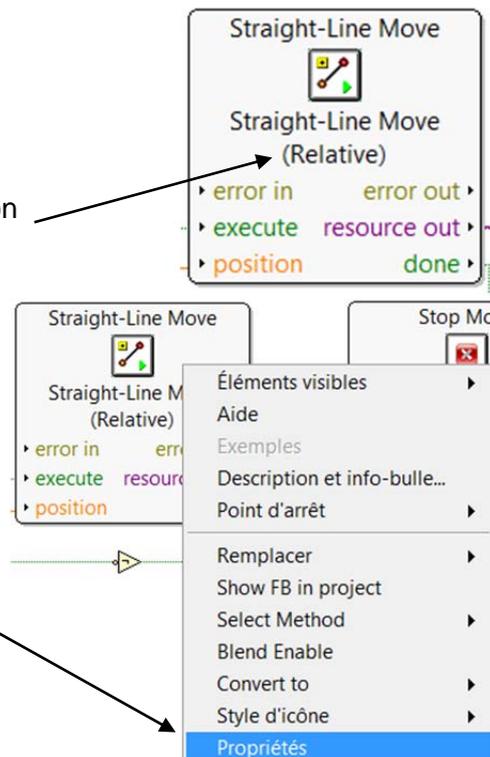
3.2.1 1^{ère} observation

Faites apparaître le diagramme du VI (ctrl + E).

Vous pouvez vous apercevoir que la méthode sélectionnée pour la fonction « Straight-Line Move » est : « Relative ».

Dans ce cas, la fonction est commandée en entrée par la position relative à la position actuelle de la maquette (*position en ° si moteur circulaire, ou mm si moteur linéaire dans motion*), mais également par l'accélération et la vitesse (Elles ont été configurées par défaut à 30 mm/s et 60 mm/s²).

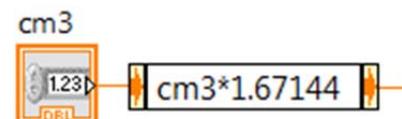
L'accélération et la vitesse sont visibles et modifiables dans les propriétés de la fonction « Straight-Line Move » (clic droit propriétés), et peuvent être paramétrées/câblées en cochant la case visible.



Terminal name	Visible	Variable scope	Data source	Default value
resource	<input type="checkbox"/>	Unsupported	Default	Motion Resource
execute	<input checked="" type="checkbox"/>	No Variable	Terminal	FALSE
position	<input checked="" type="checkbox"/>	No Variable	Terminal	0,000000
velocity	<input type="checkbox"/>	No Variable	Default	30,000000
acceleration	<input type="checkbox"/>	No Variable	Default	60,000000
deceleration	<input type="checkbox"/>	No Variable	Default	0,000000
accel. jerk	<input type="checkbox"/>	No Variable	Default	0,000000
decel. jerk	<input type="checkbox"/>	No Variable	Default	0,000000
smoothing	<input type="checkbox"/>	No Variable	Default	0,500000

3.2.2 2^{ème} observation

La valeur saisie pour le volume à injecter est multipliée par 1,67144 (coefficient multiplicateur calculé en fonction du diamètre de la seringue). Cette multiplication s'effectue par une fonction « Nœud d'expression » présente dans la palette à *Programmation / Numérique / Nœud d'expression*.





Sciences et technologies de l'Industrie et du développement durable

4 Lecture de données (vitesses, position, couple...) : Fonction « READ »

Il est possible dans SoftMotion de lire certaines données provenant des axes pilotés telles que la vitesse de déplacement du moteur ($^{\circ}/s$, ou mm/s), sa position ($^{\circ}$, ou mm), le couple (Nmm),...

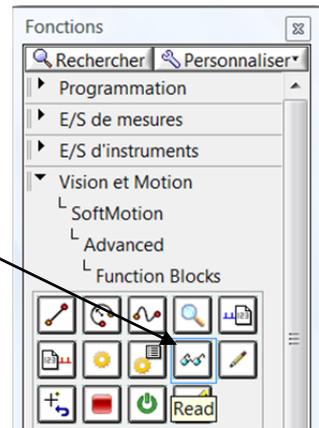
Ces données peuvent être affichées numériquement ou sous forme de graphes en temps réel dans la face avant du VI, mais aussi sauvegardées dans un fichier ou exportées sous excel.

4.1 Affichage de la vitesse et position du piston de la seringue, ainsi que du débit

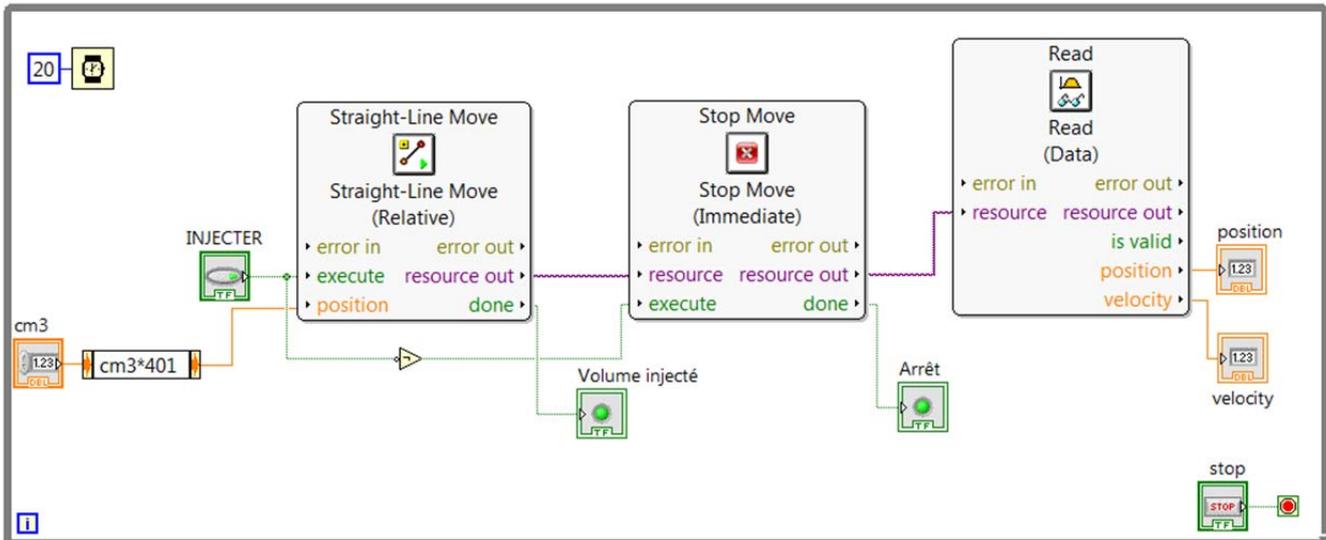
Vous allez insérer une fonction « Read » dans le diagramme afin de lire la vitesse et la position du piston de la seringue. Puis les afficher à l'aide d'un graphe déroulant.

Pour cela :

- Insérer dans le diagramme du VI une fonction « READ » présente dans la palette à *Vision et Motion / SoftMotion / Advanced / Function Blocks / Read*.
- Connectez la « resource out » du bloc fonction « Stop Move » à l'entrée « resource » du bloc « Read ».
- Insérez un indicateur en cliquant bouton droit sur position puis *Créer / Indicateur*.
- Renouveler l'opération en créant un indicateur pour la position.

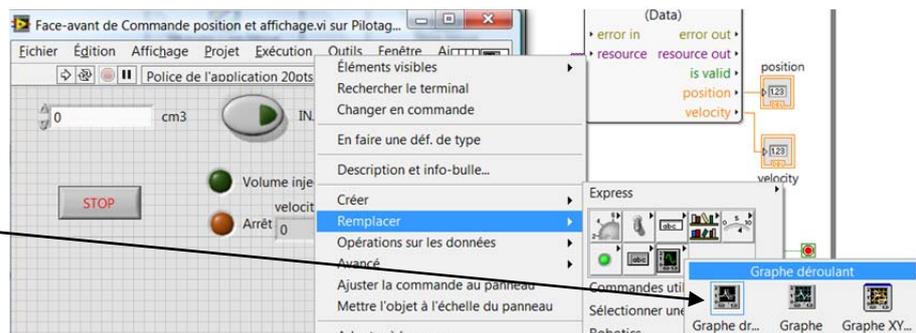


Votre diagramme devrait ressembler à cela :



Retournez maintenant dans la face avant du VI (ctrl + E).

Remplacez l'indicateur de vitesse par un graphe déroulant en cliquant bouton droit dessus, puis *Remplacer / Graphe déroulant*.



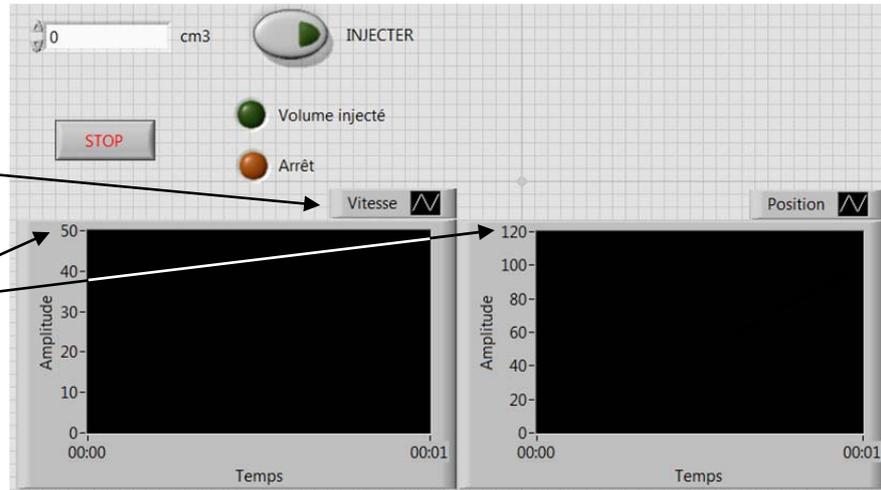


Sciences et technologies de l'Industrie et du développement durable

Faites de même avec l'indicateur de position.

Pour ces deux graphes :

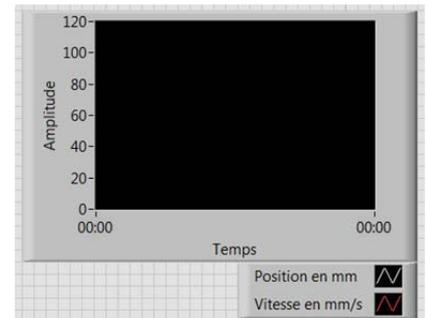
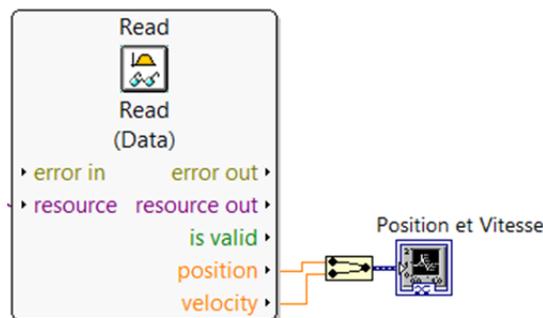
- Renommez ou supprimez les étiquettes.
- Modifiez les valeurs extrêmes de l'axe des ordonnées (0 à 120 pour la position, et 0 à 50 pour la vitesse).
- Dans l'onglet « Format d'affichage » des propriétés de ces graphes (clic droit « propriétés »), sélectionnez « Temps relatif » dans la fenêtre type.
- Décochez la mise à l'échelle automatique des Y en cliquant bouton droit sur l'axe des ordonnées et en décochant l'option.
- De la même façon cochez la mise à l'échelle des X.



Vous pouvez maintenant tester votre VI, la position affichée est la distance en mm depuis le point de départ (seringue pleine). Testez différents modes de rafraîchissement des graphes (Clic droit et *Avancé / Mode de rafraîchissement / Oscillographe*).

Afin de gagner en place et lisibilité, il est possible d'afficher plusieurs données sur un même graphe en utilisant la fonction « Assembler des signaux »  présente dans la palette à *Express / Manipulation de signaux / Assembler des signaux*.

Connectez les sorties position et vitesse de la fonction Read au même graphe (voir ci-contre), et re-paramétrez le graphe si nécessaire. Préférez le mode de rafraîchissement « Graphe déroulant ». Testez votre VI.



Avec vos connaissances, et sachant que la section du piston de la seringue est de 598,3 mm², affichez sous forme d'un graphe ou d'un indicateur simple le débit instantané injecté. Testez votre VI.

4.2 Sauvegarde des données

Vous pouvez enregistrer les données visualisées précédemment en les exportant sous excel par exemple. L'opération est simple, il suffit de cliquer bouton droit sur le graphe, puis *Exporter / Exporter les données vers Excel*.

Il est aussi possible d'enregistrer les données dans un fichier en utilisant la fonction « Ecrire dans un fichier de mesures » déjà utilisée dans durant le TD de formation sur les VI. Cette fonction se trouve dans la palette à *Express / Sortie / Ecrire dans un fichier de mesures*.



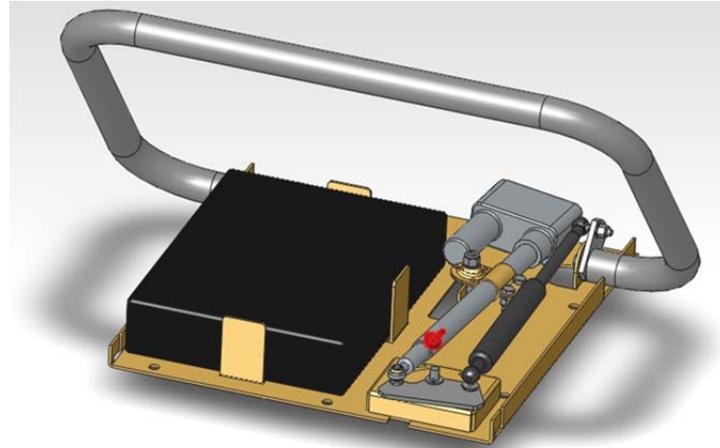
Sciences et technologies de l'Industrie et du développement durable

5 Implantation d'un capteur

Il est possible dans solidworks d'implanter des capteurs (position, vitesse, contraintes,...), et de signaler une alerte lorsque la grandeur relevée par le capteur est supérieure, inférieure, ... à une valeur donnée.

Dans Labview, l'état de ces capteurs peut être récupéré et utilisé dans un programme.

Vous allez maintenant implanter un capteur dans solidworks sur la maquette de barrière autonome de parking « Privapark » ci-contre. Ce capteur renverra une alerte quand la barrière est levée à 90° par exemple. Puis par programmation dans Labview, vous récupèrerez l'état de ce capteur, et stoppez la sortie de tige du vérin lorsque la barrière sera levée.

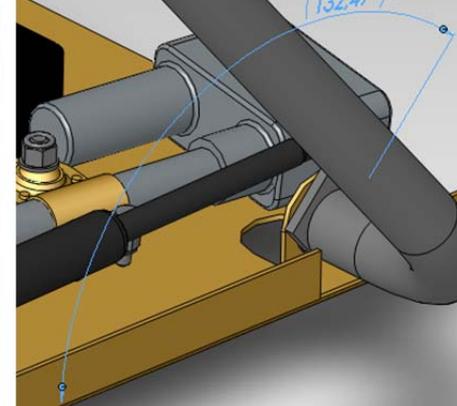
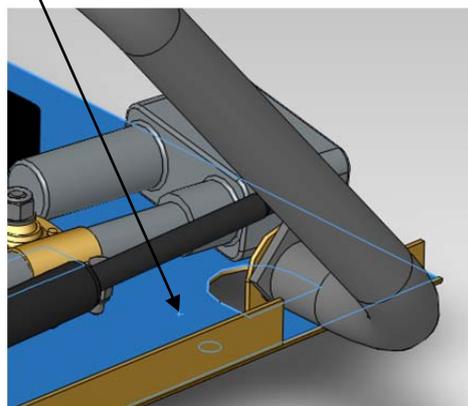
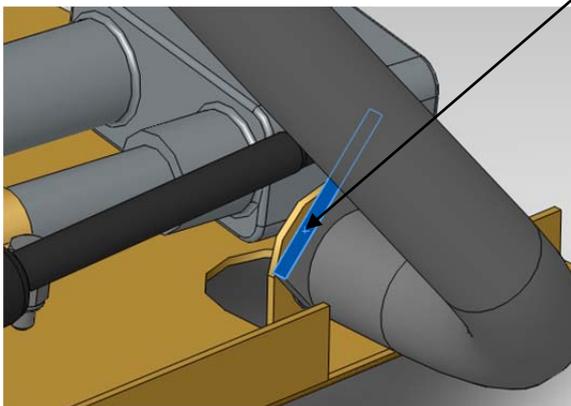


5.1 Création d'un capteur dans solidworks

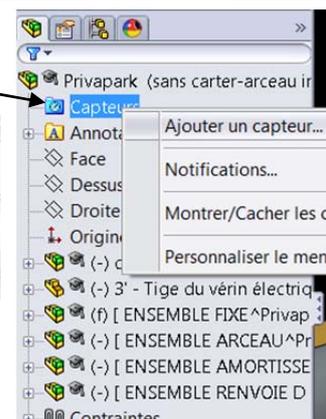
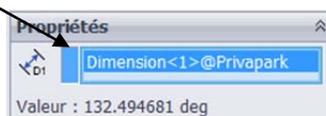
Dans un 1^{er} temps vous allez insérer une cote dans le modèle, puis vous créez le capteur.

5.1.1 Création de la cote

- Ouvrez le fichier « Privapark.SLDASM ».
- Insérer une cote  entre ces deux surfaces (cote d'angle).



- Dans l'arbre de création Feature manager de solidworks, cliquez bouton droit sur « Capteurs » puis *Ajouter un capteur...*
Sélectionner « cote » dans le menu déroulant « type de capteur »
Dans la rubrique propriété, cliquez à l'écran sur la valeur de la cote angulaire précédemment créée, elle doit apparaître dans le cadre.



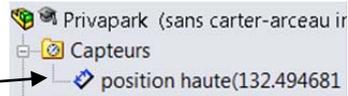


Sciences et technologies de l'Industrie et du développement durable

Cochez la case « Alerte » puis paramétrez de façon à être averti si la valeur de la cote est supérieure à 172 (Lorsque la cote implantée est de 172°, la barrière est un peu près à la verticale).

Validez de façon à créer le capteur.

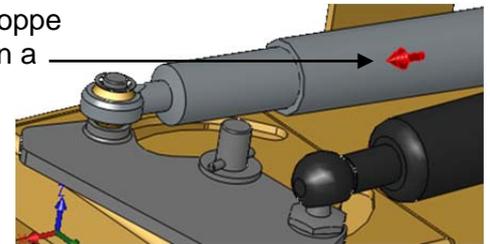
- Renommez ce capteur en « position haute »



Remarque : Vous auriez très bien pu implanter une cote entre la tige du vérin et le corps, ou autre.

5.1.2 Récupération est utilisation de l'état du capteur dans Labview

Vous allez compléter un VI qui commande la barrière en montée, et la stoppe lorsqu'elle est à la verticale. Pour cela un moteur linéaire simulant le vérin a été créé dans solidworks afin d'animer la barrière.



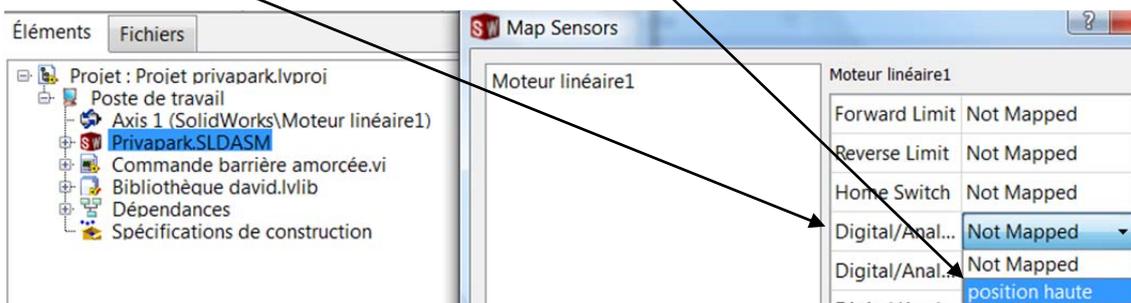
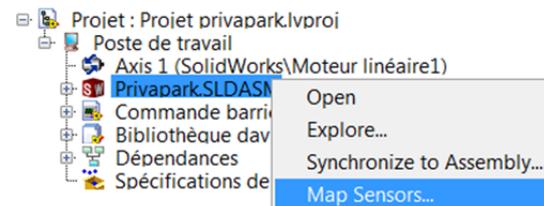
Ouvrez le projet « Projet privapark.lvproj »

Vous pouvez constater qu'un axe a déjà été créé, ainsi qu'un VI incomplet « Commande barrière amorcée.vi » afin de gagner du temps.

Déclaration du capteur

Vous allez déclarer ce capteur dans le projet Labview, en l'associant à l'axe (Axis1).

Cliquez bouton droit sur « Privapark.SLDASM » puis *Maps sensors...* Dans la fenêtre qui apparaît, attribuez le capteur « position haute » à la « Digital/Analog input 0 » du « Moteur linéaire1 ».



Remarque : les 3 premières lignes permettent d'attribuer un capteur pour définir les limites du moteur (Forward et Reverse Limit : le moteur s'arrêtera sans même avoir à le programmer) ainsi que la position zéro par défaut (Home switch).

Programmation du VI :

Ouvrez le VI « Commande barrière amorcée.vi » en double cliquant dessus dans l'arbre du projet Labview et faites apparaître le diagramme (ctrl + E). Ce VI est incomplet, néanmoins un bloc fonction « Straight-Line Move », « Stop Move » ainsi que différents indicateurs et commandes y figurent déjà, et sont déjà paramétrés.



Sciences et technologies de l'Industrie et du développement durable

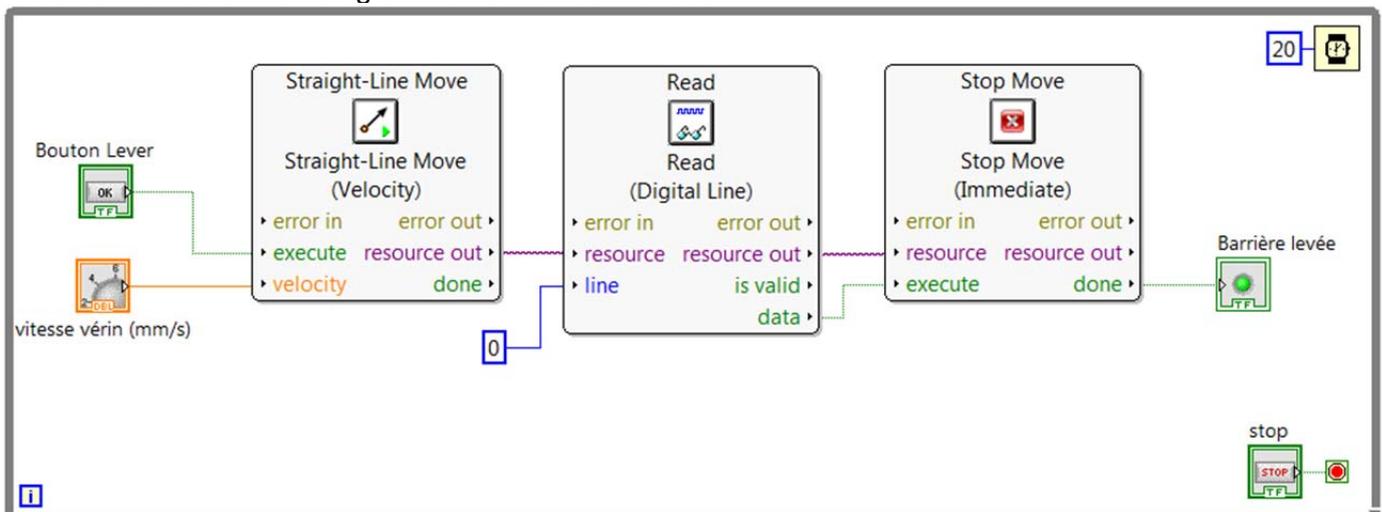
Vous allez lire l'état du capteur afin de l'utiliser dans le programme. Pour cela :

- Insérez une commande « Read » présente dans la palette fonctions à *Vision et Motion / SoftMotion / Advanced / Function Blocks / Read*.
- Sélectionnez pour la fonction « Read » la méthode « Digital Line » (Clic droit puis *Select Method / Digital Line*).
- Indiquez quelle « Digital/Analog input » la fonction « Read » doit elle lire. Pour cela cliquez bouton droit sur l'entrée « Line » puis *Créer / Constante*. Par défaut la constante est 0 ce qui nous convient puis que nous avons précédemment assigné le capteur « position haute » à la « Digital/Analog input » 0.

Connectez les éléments du diagramme entre eux de façon à :

- Commander la vitesse du vérin à l'aide du bouton rotatif « vitesse vérin (mm/s) ».
- Débuter le mouvement à l'aide du « Bouton Lever ».
- Relier les « resource out » et « resource » des différents blocs fonctions (La « resource » c'est-à-dire l'axe piloté par la fonction « Straight-Line Move » a déjà été paramétrée).
- Exécuter la fonction « Stop Move » lorsque la sortie « data » de la fonction « Read » est vrai (c'est-à-dire que le capteur de solidworks renvoie une alerte, la barrière est en position haute)
- Allumer le voyant « Barrière levée » lorsque le mouvement est stoppé.

Vous devriez obtenir un diagramme ressemblant à cela :



Testez le VI

Génial ça fonctionne...

