MODULES USB 6009 SOUS LABVIEW

Durée : 2 H.

Plan du T.P.

A. OBJECTIF DU T.P.

On désire faire des programmes sous *labview* en utilisant les modules USB 6009.

B. TRAVAIL DEMANDÉ

B.1. But :

A l'aide d'un module USB 6009, nous souhaitons créer deux sous-vi « param_mvt.vi » et « detection_butee.vi » utilisables en TP d'électronique.

B.2. Cahier des charges :

Lors des Tps d'électronique, vous apprenez à câbler une carte qui permet le pilotage d'un moteur pas à pas. Pour permettre le bon fonctionnement de celui-ci et connaître la position angulaire, vous devez brancher 4 éléments extérieurs supplémentaires (voir **Erreur ! Source du renvoi introuvable.**) :

- Un interrupteur I.1 (borne 8).
- Un compteur CP48 D2 (bornes 15 et 10).
- Un interrupteur I.3 (borne 14).
- Un potentiomètre relié à une alimentation 5V (borne 11).

Le but du Tp d'électronique est de remplacer ces 4 éléments pas une seule carte d'acquisition : module USB 6009).

Lors du TP13, vous avez créé un sous-vi « générateur de tension 0-5V », celui-ci permettra d'envoyer une tension réglable à l'aide du module USB6009 donc : sur la borne 11 vous brancherez la « borne + » de sortie analogique de votre générateur 0-5V.

Lors du TP14, pour remplacer le compteur, on utilisera le compteur du module USB6009 : PT1 qui sera gérer par un sous-vi « config_comptage.vi ». Celui-ci permettra le comptage en ° lors du déplacement du moteur (borne 15) et une remise à 0 de celui-ci si L1 est faux.

Pour remplacer les deux interrupteurs, nous allons utiliser deux sorties logiques L1 et L2 pilotées informatiquement. L1 correspondra à la Marche/Arrêt du moteur et RAZ compteur (borne 8) et L2 correspondra à la marche Avant/arrière du moteur (borne 14).Il faudra une troisième sortie logique L3 qui permettra d'arrêter le moteur si la position souhaitée est atteinte ou

si l'utilisateur souhaite arrêter puis reprendre le mouvement sans initialiser le compteur (fin de comptage : borne 10). Le pilotage de ces trois sorties logiques sera géré dans un sous-vi « param_mvt.vi ».

Dans un deuxième temps, nous ferons un sous-vi qui permet de détecter, à l'aide d'une entrée analogique, l'arrivée de la platine en butée sur le système de guidage linéaire vu en électronique.

B.3. Programme principal : paramètres de mouvement d'un moteur pas à pas :

- Créer le fichier TP15_a_votrenom.vi

B.3.1. Face avant :

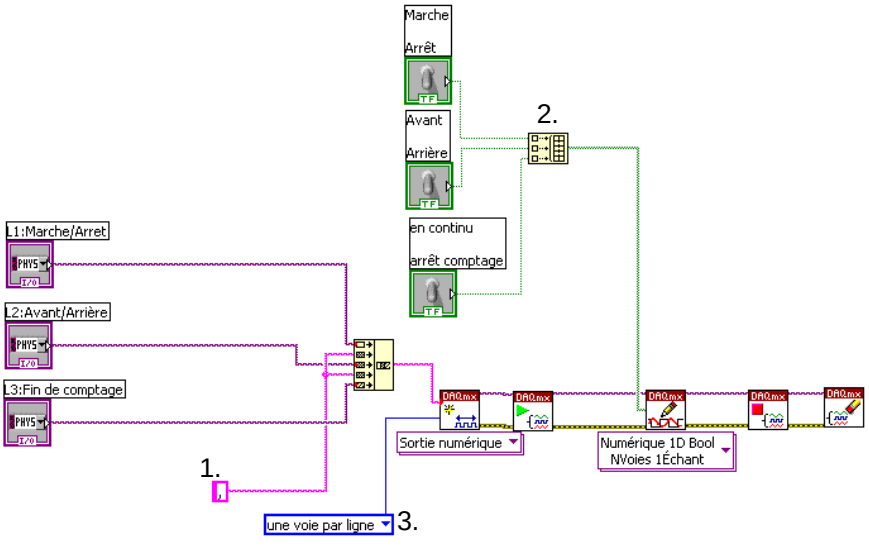
- Créer trois voies physiques DAQMX « L1 : Marche/Arrêt moteur », « L2 :Avant/Arrière » et « L3 : en continu » (A l'aide du bouton choisir « Filtrage du nom d'E/S… » et sélectionner type d'E/S = « sortie numérique»)
- Créer 1 bouton FIN DE PROGRAMME.(action mécanique = armement au relâchement)
- Créer 3 interrupteurs à bascule verticale « Marche/Arrêt moteur », « Avant/Arrière » et « En continu ».

B.3.2. Diagramme :

- Exécuter « piloter les 3 voies numériques suivant les positions de trois interrupteurs à bascule verticale » tant que «FIN DE PROGRAMME » n'est pas relâché.

Remarque :

1. Pour piloter deux voies en même temps, il faut indiquer les trois voies en les concaténant séparer d'une virgule (,).
2. Les valeurs des niveaux sont alors donner par un tableau où la première valeur du tableau correspond à la première voie, la deuxième valeur à la deuxième voie et la troisième à la dernière voie sélectionnée.
3. Il faut créer une voie par lignes pour permettre le changement d'une seule voie à la fois.



- Sauvegarder sous TP15_a_ « votre nom ».vi
- On obtient :

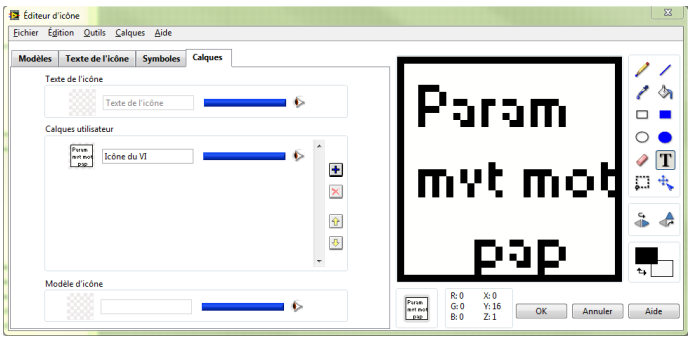
B.4. Création du sous-Vi « param_mvt.vi »:

B.4.1. Programme principal :

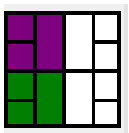
- Enregistrer votre fichier TP15_a_ « votre nom ».vi sous le nom suivant « param_mvt.vi » en utilisant enregistrer sous puis ouvrir une copie supplémentaire.
- Sur le diagramme, supprimer la boucle while en gardant juste ce qu'il y avait à l'intérieur.

B.4.2. Icône :

- Sur la face avant, cliquez à droite sur l'icône de LabView (Un éditeur d'icône s'ouvre).
- Modifier l'icône à votre façon exemple ci-contre plus cliquez dur OK



B.4.3. Connecteurs :

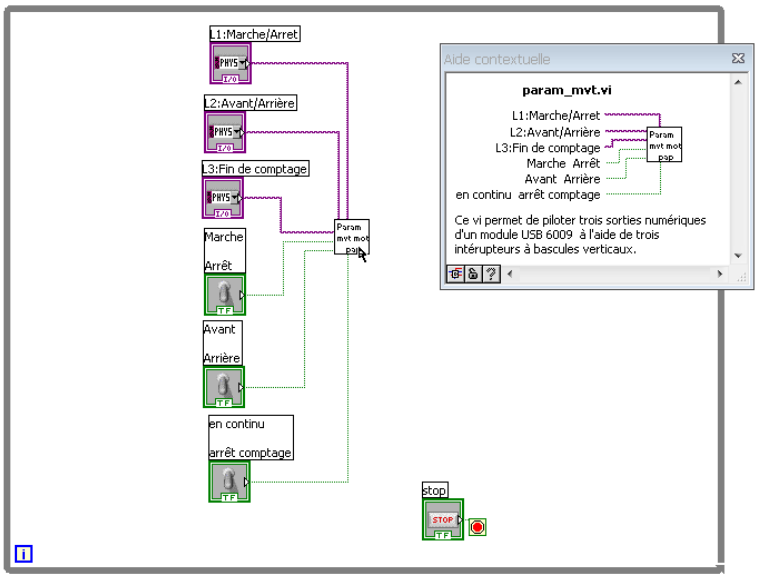


- Cliquez à nouveau sur cette nouvelle icône avec le bouton droit et choisir « Visualiser les connecteurs ».
- A l'aide du bouton droit, choisir le modèle suivant :
- Pour connecter chaque terminal, il faut choisir le connecteur avec la bobine de fil en haut à droite puis sélectionner l'élément correspondant sur la face avant. (en violet, se sont les voies DAQMX, en vert se sont les boutons)
- Enregistrer votre fichier (le sous-vi est créé)

- Dans Fichier/Propriétés du VI puis Documentation, on peut compléter à quoi servira le sous-vi.

B.4.4. Test du sous-vi :

- Créer une boucle while avec un bouton stop
- Sélectionner votre sous-vi puis les différentes commandes
- Enregistrer sous « TP15_b_test« votre nom »
- Tester votre nouveau programme.



B.5. Programme principal : détection de butée :

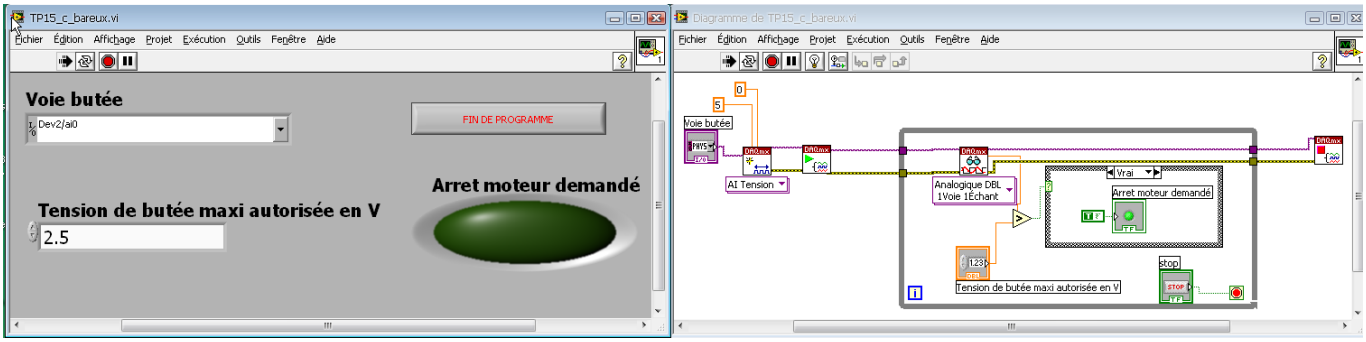
- Créer le fichier TP15_c_votrenom.vi

B.5.1. Face avant :

- Créer une voie physique DAQMX « Voie butée» (A l'aide du bouton choisir « Filtrage du nom d'E/S… » et sélectionner type d'E/S = « entrée analogique »)
- Créer 1 bouton « FIN DE PROGRAMME ».(action mécanique = armement au relâchement)
- Créer 1 voyant « Arrêt moteur demandé ».
- Une commande numérique « Tension de butée maxi autorisée en V ».

B.5.2. Diagramme :

- A l'initialisation : créer la tâche entrée analogique/tension comprise entre 0 et 5V puis démarrer la tâche.
- Exécuter « lire la tension », SI la tension lue est > à « Tension de butée maxi autorisée en V » ALORS allumer le voyant « Arrêt moteur demandé » SINON éteindre le voyant « Arrêt moteur demandé ».
- Si « FIN DE PROGRAMME » vrai Alors Arrêter la tâche sinon rien.
- Sauvegarder sous TP15_c_ « votre nom ».vi
- On obtient :



B.6. Création du sous-Vi « config_detection.vi »:

B.6.1. Programme principal :

- Enregistrer votre fichier TP15_c_ « votre nom ».vi sous le nom suivant « config_detection.vi » en utilisant enregistrer sous puis ouvrir une copie supplémentaire.

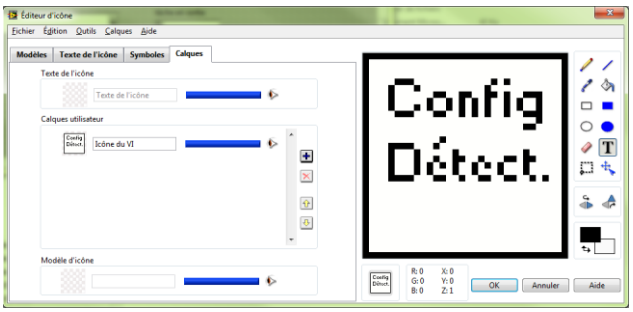
Remarque : On ne peut pas faire un sous-vi detection unique car il y a des éléments en dehors de la boucle while et que la valeur de la tension doit être lue continuellement c'est pourquoi nous allons faire deux sous-vi :

- l'une pour sélectionner la voie butée : « config_detection »
- l'autre pour détecter la demande de l'arrêt moteur suivant la valeur de la tension qui doit être lue continuellement : « detection»

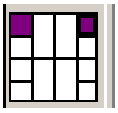
- Supprimer la boucle While et son contenu, ainsi que l'élément « Arrêter la tâche ».
- Il faut créer un indicateur en tâche de sortie.

B.6.2. Icône :

- Sur la face avant, cliquez à droite sur l'icône de LabView (Un éditeur d'icône s'ouvre).
- Modifier l'icône à votre façon exemple ci-contre plus cliquez dur OK



B.6.3. Connecteurs :



- Cliquez à nouveau sur cette nouvelle icône avec le bouton droit et choisir « Visualiser les connecteurs ».
- A l'aide du bouton droit, choisir le modèle suivant :
- Pour connecter chaque terminal, il faut choisir le connecteur avec la bobine de fil en haut à droite puis sélectionner l'élément correspondant sur la face avant.
- Enregistrer votre fichier (le sous-vi est créé)
- Dans Fichier/Propriétés du VI puis Documentation, on peut compléter à quoi servira le sous-vi.

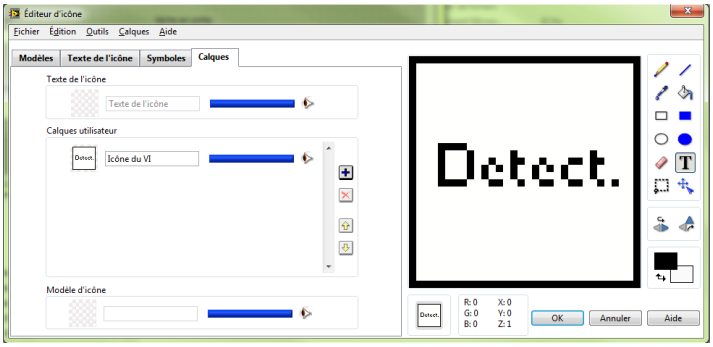
B.7. Création du sous-Vi « Detection.vi »:

B.7.1. Programme principal :

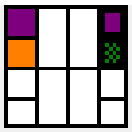
- Enregistrer votre fichier TP15_c_ « votre nom ».vi sous le nom suivant « detection.vi » en utilisant enregistrer sous puis ouvrir une copie supplémentaire.
- Supprimer tout ce qui ce fait à l'initialisation et à la fin de la boucle While. Puis supprimer la boucle While en gardant son contenu (sauf le bouton fin de programme).
- Pour la fonction « lire », il faut créer une commande « tâche en entrée » et un indicateur « tache en sortie ».

B.7.2. Icône :

- Sur la face avant, cliquez à droite sur l'icône de LabView (Un éditeur d'icône s'ouvre).
- Modifier l'icône à votre façon exemple ci-contre plus cliquez dur OK

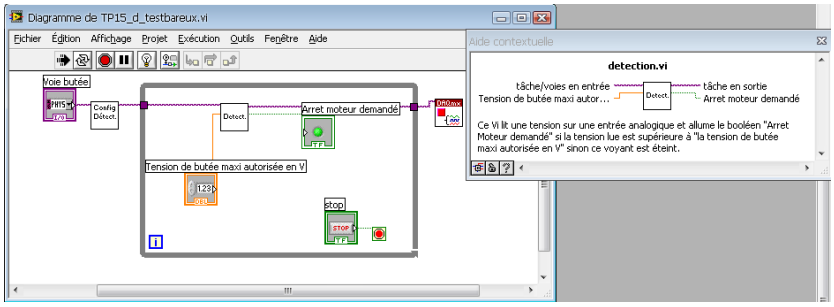


B.7.3. Connecteurs :



- Cliquez à nouveau sur cette nouvelle icône avec le bouton droit et choisir « Visualiser les connecteurs ».
- A l'aide du bouton droit, choisir le modèle suivant :
- Pour connecter chaque terminal, il faut choisir le connecteur avec la bobine de fil en haut à droite puis sélectionner l'élément correspondant sur la face avant.
- Enregistrer votre fichier (le sous-vi est créé)
- Dans Fichier/Propriétés du VI puis Documentation, on peut compléter à quoi servira le sous-vi.

B.7.4. Test du sous-vi :



- Créer une boucle while avec un bouton stop
- Sélectionner le sous-vi « config_detection »
- Sélectionner le sous-vi « detection »
- Ajouter l'élément arrêter la tâche puis les différentes commandes et indicateurs.
- Enregistrer sous « TP15_d_test« votre nom »
- Tester votre nouveau programme.