

Exemples de programmes Labview

Les programmes VI présentés ci-après, utilisent les composants de la bibliothèque TXLAB pour accéder au contrôleur Robo TX.

Programme N°1 : Test du port de branchement du boîtier USB2Dynamixel

Nom du VI : « Test-ports.vi »

Précisions sur le fonctionnement :

Après avoir appuyé sur le bouton INIT, le programme effectue une boucle pour tester à tour de rôle les ports COM de 1 à 40. La vitesse de transmission est réglable (1 = vitesse maximale par défaut) ainsi que le numéro de port maximal à tester (N°40 par défaut). Si un port valide est trouvé, la boucle est stoppée.

L'appui sur la touche STOP met fin au programme.

Diagramme et face avant :

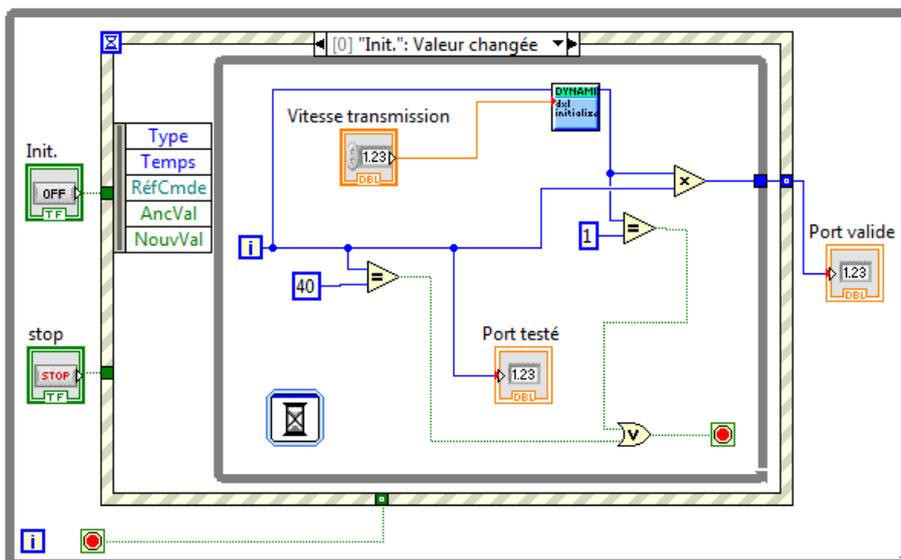
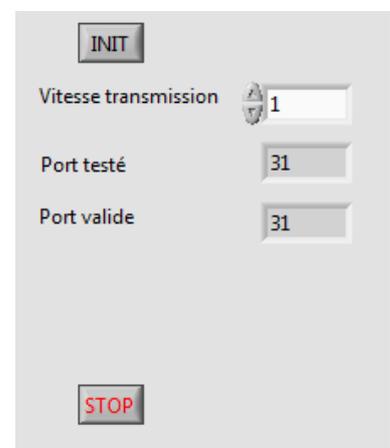


Diagramme si appui sur la touche INIT



Le boîtier USB2Dynamixel a été trouvé sur le port 31.

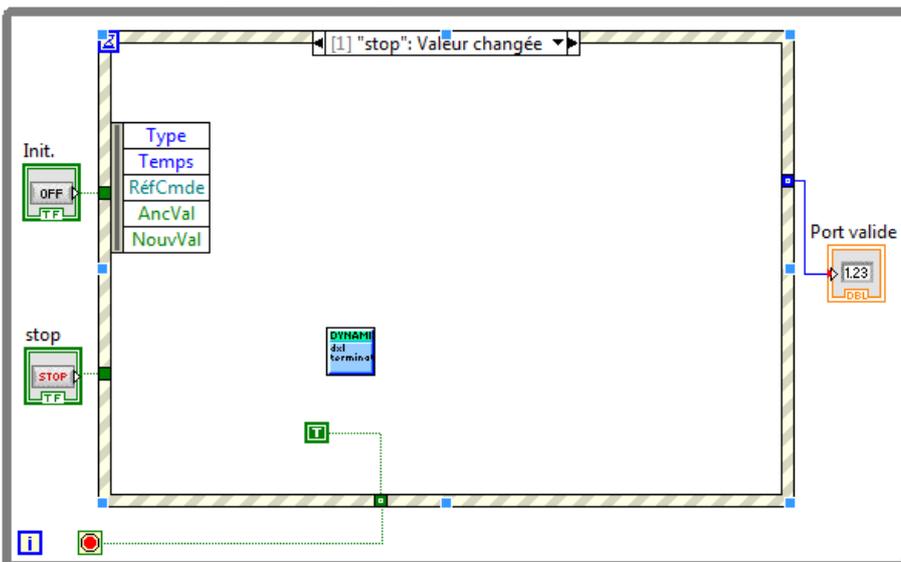


Diagramme si appui sur la touche STOP

Programme N°2 : Test la présence de servomoteurs et de capteurs sur le bus TTL

Nom du VI : « Ping servos.vi »

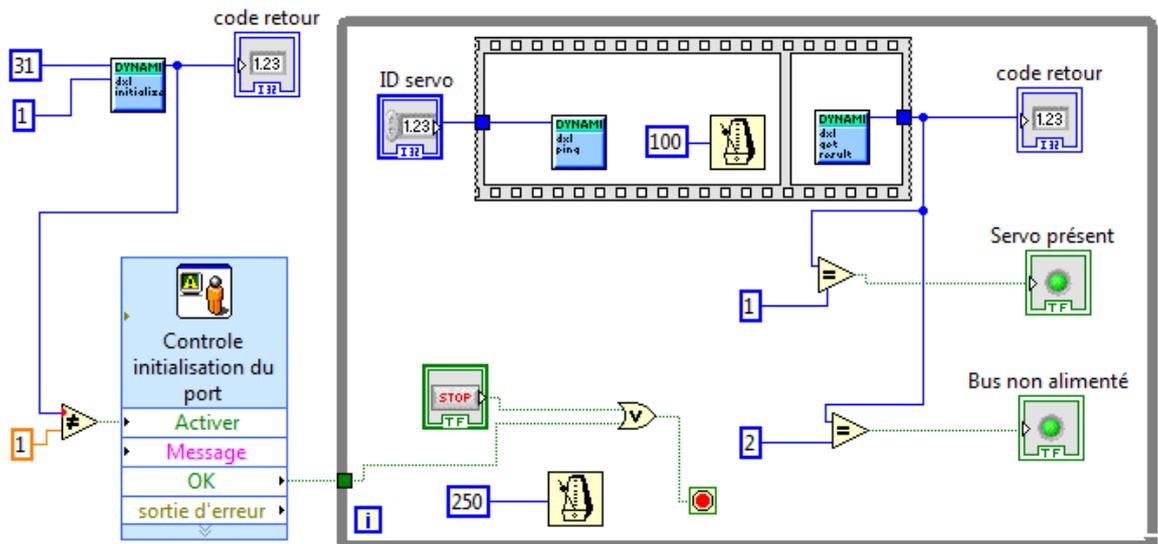
Précisions sur le fonctionnement :

Le programme initialise d'abord la connexion avec le boîtier USB2Dynamixel (port COM 31 par défaut à modifier éventuellement, vitesse de communication maximale). Chaque identificateur de servomoteur ou capteur est ensuite testé manuellement en faisant varier l'entrée « ID servo ». Le code de statut de la communication est affiché (voir plus haut la fonction « dxl_get_result » pour les codes de statut disponibles).

Si le code renvoyé est 1 (Paquet d'instructions reçu avec succès) le voyant « Servo présent » s'allume).

Si le code renvoyé est 2 (Erreur lors de la transmission du paquet d'instructions) le voyant « Bus non alimenté » s'allume.

Diagramme :



Face avant :



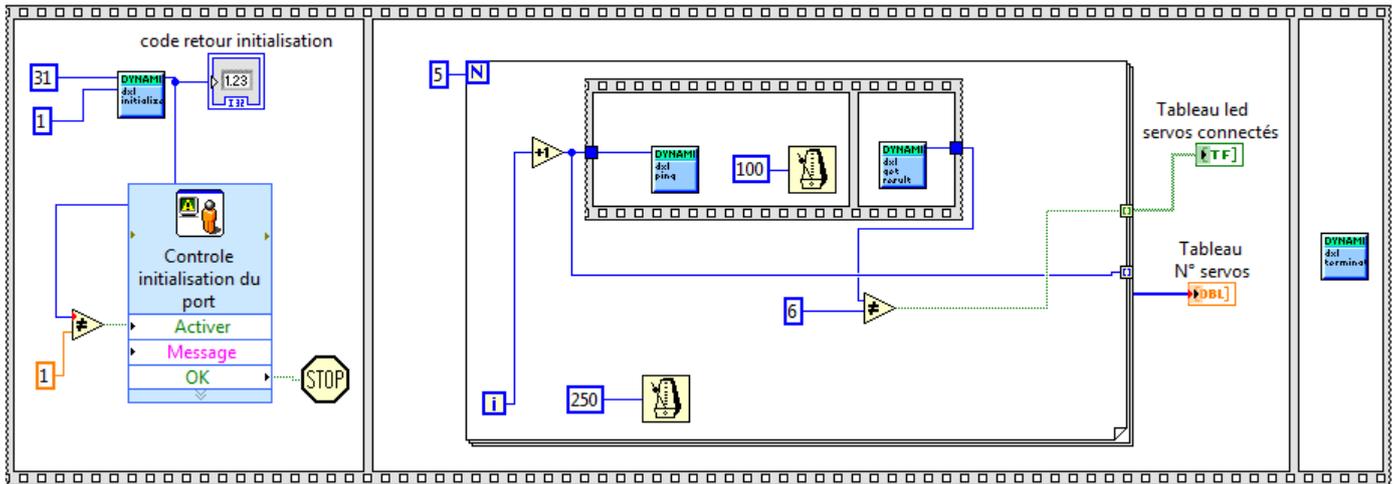
Programme N°3 : Test automatiquement la présence de 5 servomoteurs.

Nom du VI : « Ping 5 servos.vi »

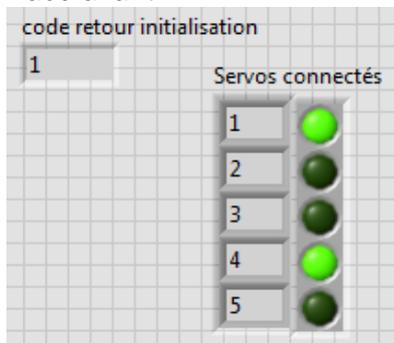
Précisions sur le fonctionnement :

Ce programme est une variante du programme précédent. Une boucle est effectuée 5 fois pour un test sur les cinq premiers ID (ce qui correspond aux 5 servomoteurs de la boîte « BIOLOID Beginner Kit »). Le résultat est affiché sous forme de 5 leds.

Diagramme :



Face avant :



Programme N°4 : Identifie si l'élément connecté est un servomoteur ou un capteur.

Nom du VI : « Servo ou capteur.vi »

Précisions sur le fonctionnement :

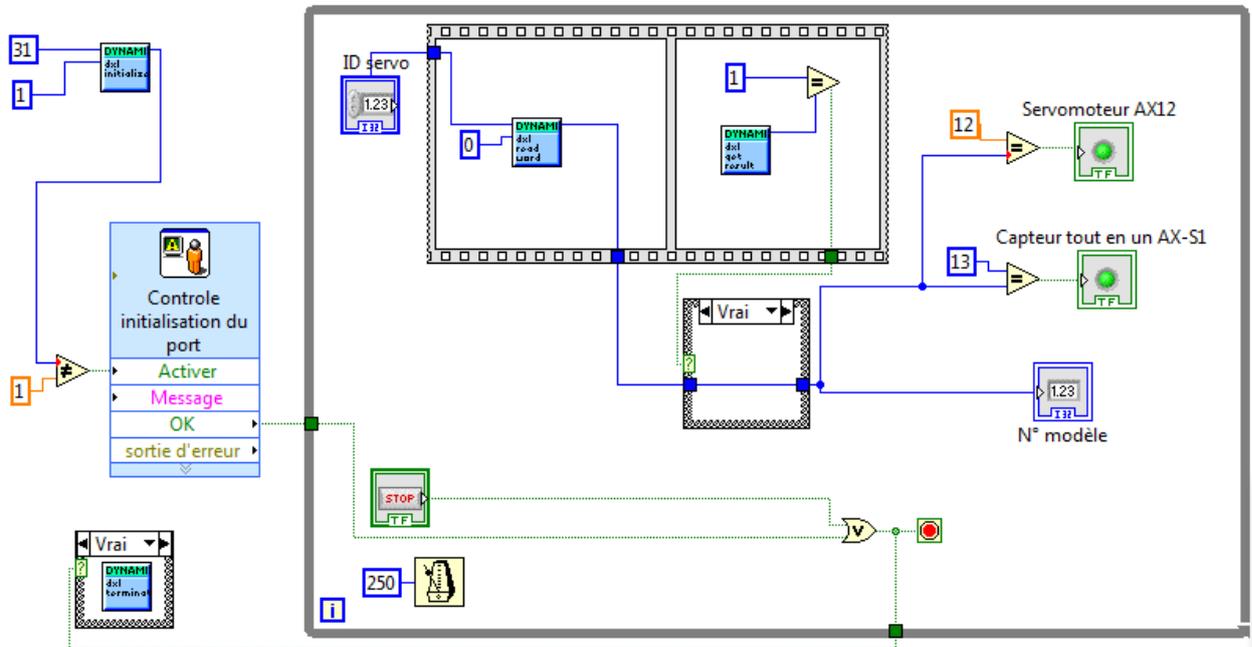
Le programme suivant effectue une lecture dans la mémoire de l'élément branché sur le bus. Le numéro de modèle identifiant les appareils branchés sur le bus Dynamixel se situe à l'adresse 0x00 et est codé sur deux octets (voir documentations sur le servomoteur AX12 et sur le capteur tout-en-un AX-S1). Les numéros sont les suivants :

- 12 pour le servomoteur AX12
- 13 pour le capteur tout-en-un AX-S1

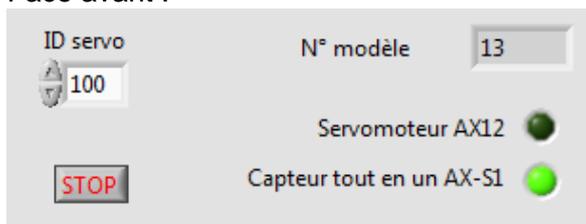
La fonction « dxl_read_word » renvoie un mot de deux octets depuis la mémoire de l'élément connecté sur le bus ; elle est associée à la fonction dxl_get_result qui permet de contrôler si la lecture est bien valide et a été effectuée avec succès.

Remarque : sur la boîte BIOLÖID Beginner Kit, les servomoteur AX12 ont les ID 1 à 5 et le capteur tout-en-un AX-S1 à l'ID 100.

Diagramme :



Face avant :



Programme N°5 : Récupération de la température et tension d'alimentation d'un servomoteur

Nom du VI : « température et tension servo.vi »

Précisions sur le fonctionnement :

Ce programme récupère depuis la mémoire d'un servomoteur connecté sur le bus, le firmware installé, la température interne et la tension d'alimentation de ce dernier. Ces différentes données sont stockées aux adresses suivantes :

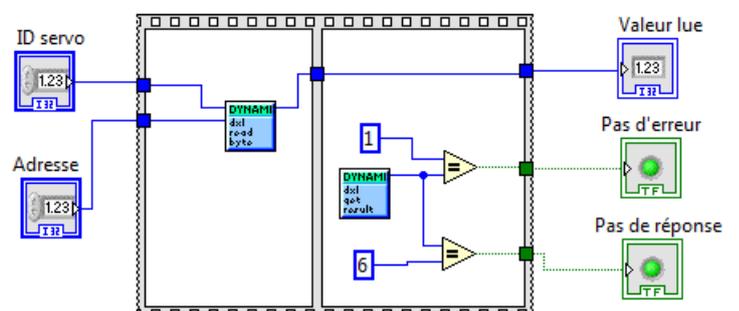
- Firmware : adresse 0x02 ; un octet
- Température : adresse 0x42 ; un octet
- Tension : adresse 0x43 ; un octet

Voir documentation sur le servomoteur AX12 pour plus de renseignements.

La lecture de la valeur mémoire d'un octet est traitée par le sous-programme « LIT OCTET » repéré sur le diagramme par l'icône



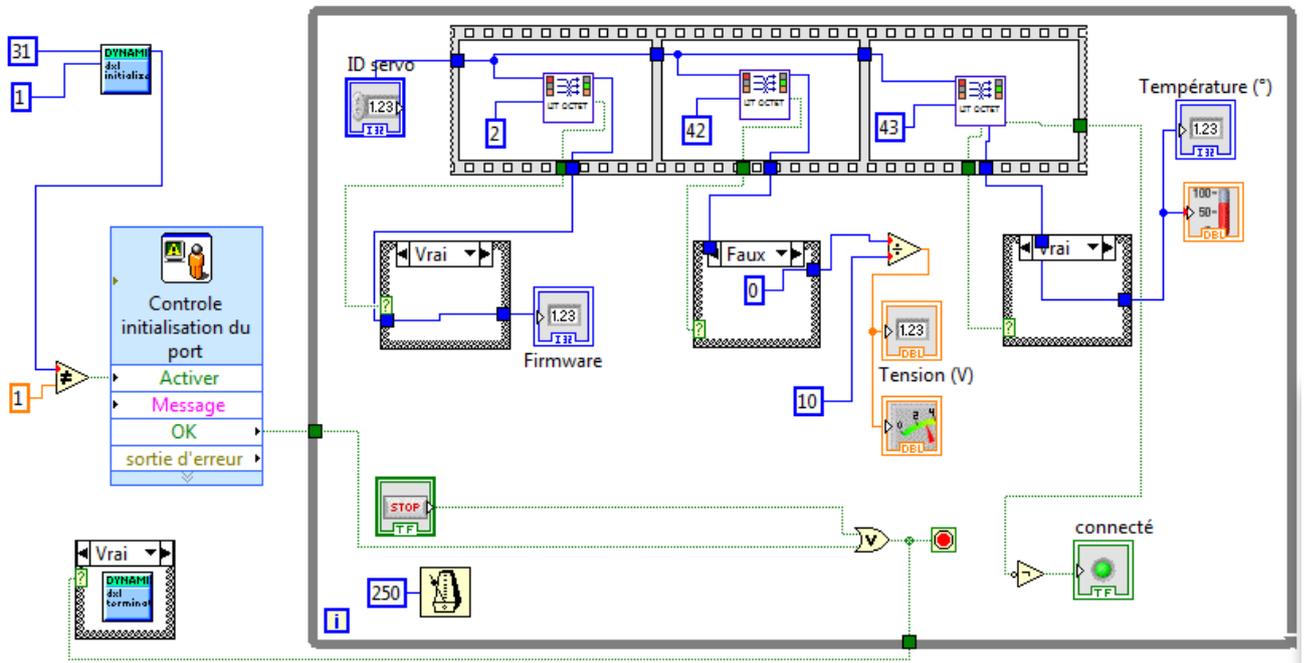
Ce sous-programme (DynaUSB – lit octet.vi) traite la lecture d'un octet et le contrôle de la validité de la transmission (voir explications données dans le programme N°4). A partir de l'ID du servomoteur et de l'adresse à lire dans sa mémoire, il renvoie la valeur de l'octet lu et deux indicateurs booléens indiquant



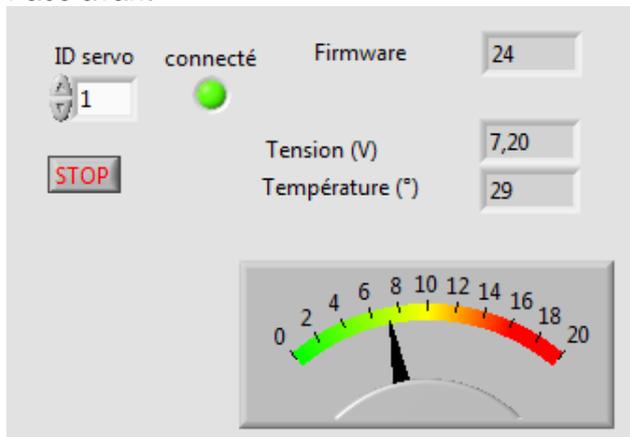
s'il y a eu une erreur et une réponse.

Dans le programme principal (voir diagramme ci-dessous), si l'indicateur « Pas d'erreur » renvoie la valeur vrai, la valeur lue est traitée et affichée ; sinon elle est remplacée par la valeur zéro.

Diagramme :



Face avant



Programme N°6 : Affichage et modification de l'état de la LED du servomoteur AX12

Nom du VI : « Allumage LED.vi »

Précisions sur le fonctionnement

Le programme récupère l'état d'allumage de la LED situé à l'adresse 25 du servomoteur grâce à la fonction « LIT OCTET » (vue précédemment). Un bouton ON/OFF permet de modifier cet état par écriture à l'adresse 25. Le déclenchement de la modification est effectué à condition que l'état effectif de la LED et la position du bouton ON/OFF soient différents grâce à une structure condition.

L'écriture vers la mémoire du servomoteur se fait avec la fonction « dxl_write_byte ». comme pour la lecture le contrôle de validité de l'écriture se fait ensuite grâce à la fonction « dxl_get_result ».

Face avant :

