



<p>Origine</p> <p>Accélération (kHz) <input type="text" value="20"/></p> <p>Vitesse Lente (mm/s) <input type="text" value="2"/></p> <p>Vitesse Rapide (mm/s) <input type="text" value="20"/></p>	<p>Aller</p> <p>Accélération (kHz) <input type="text" value="20"/></p> <p>Vitesse (mm/s) <input type="text" value="109"/></p> <p><input type="text" value=""/></p>	<p>Retour</p> <p>Accélération (kHz) <input type="text" value="50"/></p> <p>Vitesse (mm/s) <input type="text" value="127"/></p> <p><input type="text" value=""/></p>
<p>Paramètres</p> <p>Vitesse Min. (mm/s) <input type="text" value="1"/></p> <p>Vitesse Max. (mm/s) <input type="text" value="500"/></p> <p>Longueur Axe (mm) <input type="text" value="1500"/></p> <p>Résolution (pulses/mm) <input type="text" value="40"/></p>	<p>A/R</p> <p>Tempo Arrivée (ms) <input type="text" value="1000"/></p>	<p>Multi A/R</p> <p><input type="text" value="0"/></p> <p><input type="text" value="2"/></p> <p>Tempo Départ (ms) <input type="text" value="5000"/></p>
<p>STOP</p>	<p>Position Départ (mm) <input type="text" value="50"/></p> <p>Position en cours (mm) <input type="text" value="+50"/></p>	<p>Position Arrivée (mm) <input type="text" value="1450"/></p> <p>Vitesse en cours (mm/s) <input type="text" value="+0"/></p>
<p>SOPROLEC</p>		

Démo Axe Linéaire

1 Programme Basic

Le programme est con u pour pouvoir fonctionner sans IHM tout en  tant structur  pour fonctionner avec.

C'est une synth se des besoins exprim s par nos clients pour diff rentes applications.

Il est structur  de fa on   fournir des briques utilisables individuellement. Le cycle AR montre comment utiliser les cycles individuels Aller et Retour : on lance le cycle Aller en mettant le bit MBB_GALLER   1 et on sait qu'il est termin  lorsqu'il est revenu   0.

Dans cet exemple nous ne revenons pas sur la m thode de programmation s quentielle largement d taill e dans notre Livre de recette.

Les param tres sont stock s dans les registres en EEPROM et facilement accessibles avec TestCenter gr ce   l'Editeur EEPROM Utilisateur.

Pour les initialiser   des valeurs par d faut il suffit de mettre   1 le MBBit 415, il est remis automatiquement   0.

	320	328	336	344	352	360	368	376	384	392	400	408
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0

Adresse = 401

Eermer

Le programme est d compos  en 5 cycles :

1. Cycle de prise d'origine (Homing) recherche du capteur (IN_FDC_ORIGINE) puis positionnement   la position de d part
2. Cycle Aller depuis la Position de D part jusqu'  la Position d'Arriv e, la sortie 1 est activ e pendant le d placement
3. Cycle Retour depuis la Position d'Arriv e jusqu'  la Position de D part, la sortie 2 est activ e pendant le d placement
4. Cycle AR qui encha ne les 2 cycles pr c dents avec une pause en Position d'Arriv e, la sortie 3 est activ e pendant la pause
5. Cycle AR multiples qui ex cute en continu le cycle pr c dent avec une pause en Position D part, la sortie 4 est activ e pendant la pause.

Entr es :

IN_FDC_ORIGINE : entr e du capteur de prise d'origine, TYPE_FDC permet de d finir le type de capteur NO ou NF.

IN_ORIGINE : un front montant sur cette entr e d clenche le cycle de homing.

IN_ALLER, IN_RETOUR, IN_AR, IN_MULTI_AR : un front montant d clenche le cycle correspondant, un front descendant sur IN_MULTI_AR arr te le cycle.

Sorties :

OUT_ALLER : mise à 1 pendant le déplacement Aller

OUT_RETOUR : mise à 1 pendant le déplacement Retour

OUT_TOP_PA : mise à 1 pendant la temporisation en position d'arrivée

OUT_TOP_PD : mise à 1 pendant la temporisation en position de départ

Paramètres en EEPROM:

Les constantes indiquent le registre dans lequel est stockée la valeur. En mode 16 bits les registres sont numérotés de 0 à 511 pour les commandes GetEEData16 et SetEEData16 et de 5120 à 5631 pour leur accès en Modbus.

EE_RESOLUTION (0 / 5120) : Résolution : indiquée en pulses par mm, par exemple un driver est réglé à 1600 pulses par tour et l'axe parcourt 40mm par tour. Il faudra donc 1600 pulses pour parcourir 40mm, soit $1600/40 = 40$ pulses par mm

EE_ACCEL_HOM (1 / 5121) : Accélération pour le cycle de Homing

EE_VITESSE_HOME_R (6 / 5126) : Vitesse Rapide pour la recherche du capteur

EE_VITESSE_HOME_L (7 / 5127) : Vitesse Lente pour la sortie du capteur

EE_ACCEL_ALLER (2 / 5122) : Accélération pour cycle aller

EE_ACCEL_RETOUR (3 / 5120) : Accélération pour cycle retour

EE_VITESSE_MIN / EE_VITESSE_MAX (4 / 5124) (5 / 5125) : borne pour sélection des vitesses aller et retour

EE_VITESSE_ALLER (8/ 5128) : Vitesse pour cycle aller

EE_VITESSE_RETOUR (9 / 5129) : Vitesse pour cycle retour

EE_LONGUEUR_AXE (11 / 5131) : Longueur de l'axe, utilisée pour définir la course maximale de la prise d'origine

EE_POSITION_DEPART (12 / 5132) : Position de départ pour le mouvement Aller

EE_POSITION_ARRIVEE (13 / 5133) : Position d'arrivée pour le mouvement Aller

EE_TEMPO_DEPART (14 / 5134) : Temporisation à la position départ dans le cycle d'Aller/Retour multiples. La sortie 4 est active pendant cette temporisation.

EE_TEMPO_ARRIVEE (15 / 5135) : Temporisation à la position arrivée dans le cycle d'Aller/Retour. La sortie 3 est active pendant cette temporisation.

EE_NB_AR (16 / 5136) : Nombre de cycles A/R dans le cycle A/R multiples. Si ce registre vaut 0 le nombre de cycles A/R est illimité.

Bits Modbus de cycle:

Ces bits sont mis à 1 par la détection d'un front montant sur les entrées associées ou par les boutons de l'IHM.

MBB_GHOME : mis à 1 il lance le cycle de prise d'origine et y reste tant que le cycle n'est pas terminé

MBB_GALLER : mis à 1 pendant un cycle Aller, lorsqu'il revient à 0 le cycle est terminé.

MBB_GRETOUR : mis à 1 pendant un cycle Retour, lorsqu'il revient à 0 le cycle est terminé.

MBB_GAR : mis à 1 pendant un cycle A/R, il met à 1 successivement MBB_GALLER et MBB_GRETOUR.

MBB_GMULTI_AR : mis à 1 pour exécuter des cycles A/R

Autres Bits Modbus :

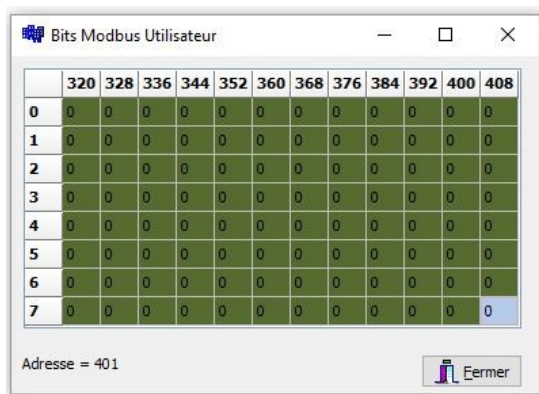
MBB_HOME_OK : mis à 1 si la prise d'origine s'est terminée avec succès

MBB_MODE_AUTO : mis à 1 pendant pendant un cycle automatique. Utilisé, notamment, pour empêcher les déplacements manuels.

MBB_STOP : accessible depuis l'IHM, pour arrêter les cycles

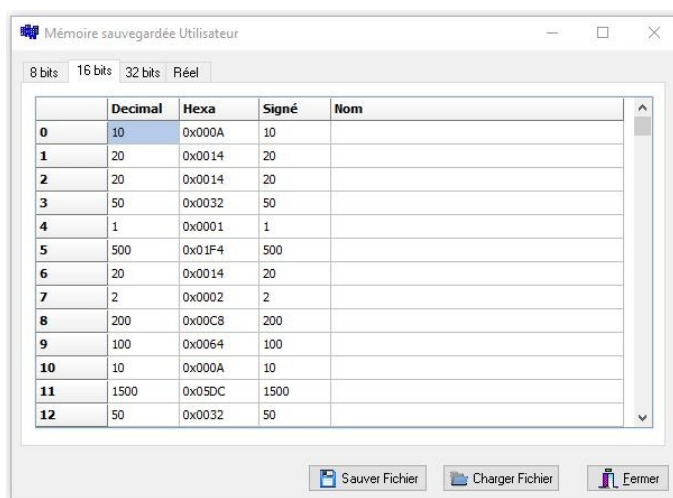
MBB_X_PLUS & MBB_X_MOINS : lance un déplacement en mode manuel sur un front montant et l'arrête sur un front descendant

Dans cet écran on peut suivre l'état des différents bits Modbus :



2 Utilisation sans IHM

Les paramètres de fonctionnement étant stockés en EEPROM ils ne sont pas perdus en cas d'extinction de l'automate. Il est très simple d'y accéder et de les modifier dans l'écran **Mémoire sauvegardée Utilisateur** :

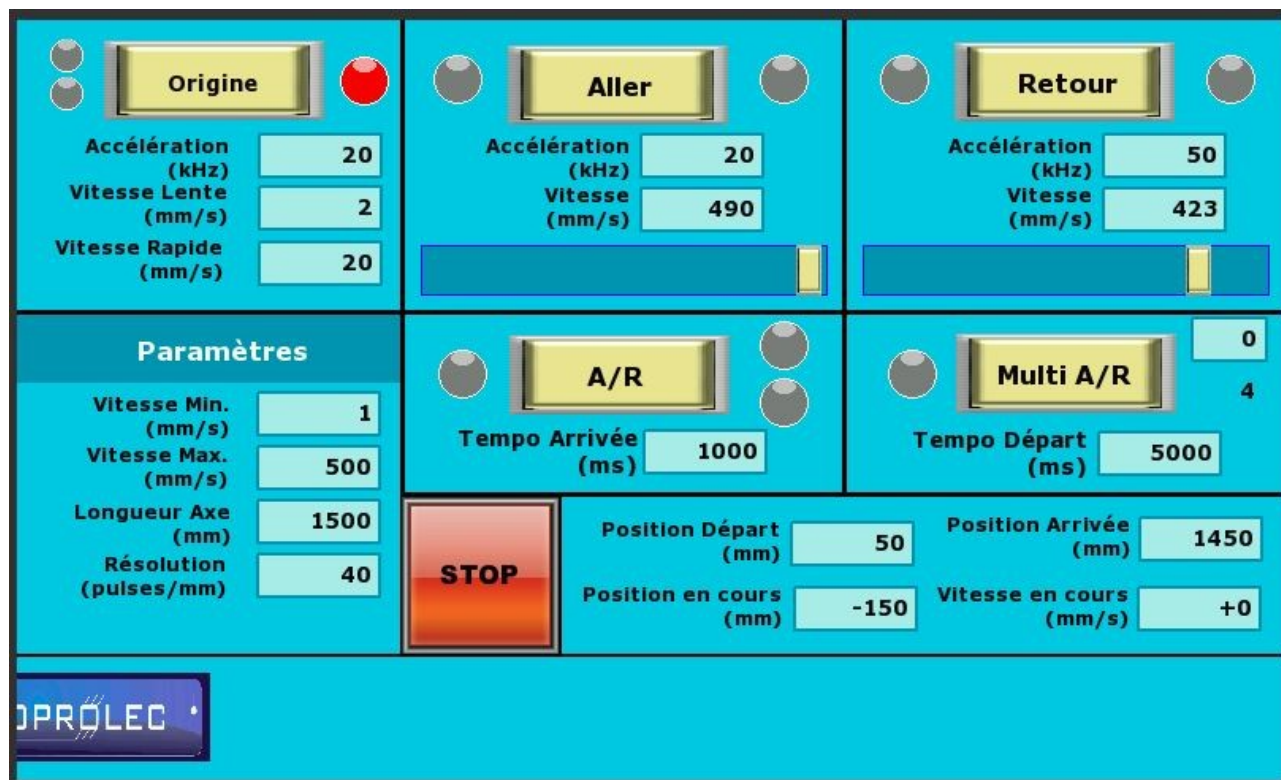


Les cycles sont déclenchés en mettant à 1 les différentes entrées décrites plus haut.

Si toute ou partie des paramètres sont changés fréquemment ou pour éviter la "boutonnerie", il est possible de connecter un IHM sur le port RS485 de l'automate pour le piloter.

3 IHM

Dans cette application nous avons tout regroupé sur un seul écran :



Avancer de commencer il faut régler les paramètres du système :



Il faut ensuite lancer la prise d'origine pour que le voyant Origine OK passe au vert :



Le bouton Origine est "relié" au bit Modbus 320 : MBB_GHOME

Ce voyant est "relié" au bit Modbus 321 : MBB_HOME_OK

Le principe est le même pour les autres boutons et voyants :

