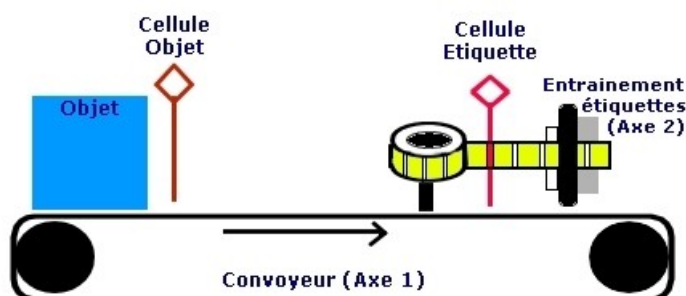


## Cas pratique de mise en œuvre d'une étiqueteuse simple, pilotée par la carte InterpCNC 5 axes SOPROLEC, avec réalisation d'un écran tactile (exemple ici: IHM de type Kinco GL70E).

### Principe de fonctionnement :

- un convoyeur indépendant (axe 1, ON/OFF manuel) déplace les objets déposés à l'entrée, vers le poste d'étiquetage.
- une 1ère cellule optique (front montant sur l'entrée 1 de la carte) détecte la présence d'un objet à étiqueter.
- à ce moment là, une temporisation prédéfinie (timer) est lancée afin de déplacer l'objet jusqu'au point de contact avec une étiquette.
- à l'issue de cette temporisation, une étiquette est lancée (NB : supposons que le rouleau d'étiquettes est entraîné par un axe monté directement sur l'arbre d'un moteur pas à pas (axe 2), en appui contre un rouleau presseur.)
- une 2ème cellule optique (front montant sur l'entrée 2 de la carte), détecte le moment du passage de l'étiquette suivante, ce qui permet en fonction de la position actuelle relevée sur l'axe, de définir le nombre de pulses restant à envoyer au moteur pour accomplir le mouvement jusqu'à la position souhaitée de l'étiquette (distance entre la cellule et la fin de l'étiquette lancée).



### Calculs et pré-requis :

Nous utiliserons le logiciel InterpCNC Test Center (téléchargeable sur le site [www.soprolec.com](http://www.soprolec.com)), lequel permet de modifier, monitorer, debugger, envoyer en Ram et dans l'EEPROM de la carte votre programme Basic pour qu'il soit exécuté.

Vous pouvez visualiser, modifier, forcer l'état des bits et adresses mémoire utilisateur afin de tester votre programme.

L'utilitaire DTools de Kinco, sera nécessaire pour l'édition de vos pages d'écrans et la configuration des IHM de cette marque.

Nous avons tout d'abord besoin de déterminer la résolution de l'axe d'étiquetage (axe 2). Elle s'exprime en pulses par millimètre.

Nous avons également besoin de connaître la longueur de nos étiquettes, exprimée en millimètres.

1) Supposons que notre moteur (axe 2, entraînement direct) ait une résolution de 1600 pulses/tour, et qu'il fasse 50mm de diamètre, sa résolution est de :

$$1600 / (50 \cdot \pi) = 10,2 \text{ pulses/mm}$$

2) Supposons que nos étiquettes aient une longueur de 40mm, leur longueur correspond alors à :  
 $40 \times 10,2 = 408$  pulses

Le mouvement d'un axe est défini par une accélération(khz/s), une vitesse d'exécution(hz), une décélération(khz/s), et enfin une distance cible (en pulses).

Nous allons aussi prévoir une gestion de recettes (= pré-réglages pour différents objets à étiqueter).

Nous stockerons dans l'EEPROM de la carte ces réglages concernant l'étiquetage :

- vitesse d'accélération, d'exécution, et de décélération de l'axe.
- temporisation nécessaire au déplacement de l'objet.
- paramètres des recettes (nom de la recette, longueur maxi de l'étiquette, position de l'étiquette, ...)

L'écran communique avec la carte InterpCNC via une connexion Modbus RTU dont il est « Maître ». Il peut donc lire et écrire directement aux adresses Modbus de la carte (Voir la documentation Modbus sur notre site [www.soprolec.com](http://www.soprolec.com)).

Ainsi, les valeurs visibles l'écran sont celles lues aux adresses de la carte, et les valeurs saisies à l'écran iront remplacer directement ces mêmes valeurs.

NB : les registres en EEPROM ayant un nombre d'écritures limité (de l'ordre d' 1 million de cycles), il convient de les utiliser avec précaution (surtout pas de boucle sans fin avec des écritures, lors de vos tests de programmes !...).

### **Le programme Basic :**

Notre programme Basic débute par la définition de constantes, des bits, et adresses Modbus requis. Ainsi pour limiter le nombre d'écritures en EEPROM, nous faisons le test si nous y avons déjà écrit les paramètres, sinon nous procédons.

Ensuite nous initialisons la carte (Unlock, et remise à zéro des bits utilisateurs), ainsi que nos variables (sub-routine « Initialisation »).

Le cœur de notre programme se situe dans une boucle « DO ... LOOP » dans laquelle nous implémentons 2 cycles, précédés par une partie combinatoire (test de bits déterminant ensuite l'exécution à l'intérieur des cycles).

Après toute modification détectée de registres Modbus de la carte, nous actualisons la valeur de chacune de nos variables (sub-routine « CalculParametres »).

Pendant l'exécution de chaque cycle, l'utilisation de l'instruction « select case » est idéale pour reproduire les étapes des évolutions séquentielles, et les tests « if...elseif... endif » les transitions.

### 1<sup>er</sup> cycle : GCYCLE

→ Sur l'écran, le bouton « Départ Cycle » activera ou non le bit utilisateur MBB\_GCYCLE, dont le test (case 10) permettra d'entrer dans le cycle principal GCYCLE (et si nous ne sommes pas dans un test d'étiquette)

→ case 20 : Si un objet est détecté sur l'entrée 1 (front montant), alors je lance la temporisation qui définit l'avancée de l'objet. Passage à l'étape suivante

→ case 30 : si le timer est terminé, passage à l'étape suivante

→ case 40 : je définis la position actuelle de l'axe-étiquettes comme point 0, et je lance un mouvement d'étiquette sur une longueur de 2 étiquettes. Passage à l'étape suivante

→ case 50 : si je détecte le défilement de l'étiquette suivante (front montant sur l'entrée 2), alors je relève la position actuelle de l'axe-étiquettes, et je lance un mouvement de cet axe jusqu'à la position définie pour la pose. Passage à l'étape suivante (case 60).

Sinon, si l'axe-étiquettes a pu terminer le mouvement initial pour 2 étiquettes(=timeout), on génère une alarme car une anomalie s'est alors produite dans la détection du défilement des étiquettes. Dans ce cas, arrêt du cycle et passage à l'étape 10.

→ case 60 : si l'axe-étiquettes a fini son mouvement pour 1 étiquette, alors j'incrémente le compteur partiel d'étiquettes (en Ram). Passage à l'étape 10

→ case else : dans tout autre cas, on affiche une erreur avec la valeur d'étape.

### 2<sup>ème</sup> cycle : GTESTETIQUETTE

→ sur l'écran, le bouton « Test Etiquette » activera ou non le bit MBB\_TEST\_ETIQUETTE dont le test (case 10) permettra d'entrer dans le cycle GTESTETIQUETTE.

→ case 20 : je définis la position actuelle de l'axe-étiquettes comme point 0, et je lance un mouvement d'étiquette sur une longueur de 2 étiquettes. Passage à l'étape suivante.

→ case 30 : si je détecte le défilement de l'étiquette suivante (front montant sur l'entrée 2), alors je relève la position actuelle de l'axe-étiquettes, et je lance un mouvement de cet axe jusqu'à la position définie pour la pose. Remise à zéro du bit MBB\_TEST\_ETIQUETTE. Passage à l'étape 10.

Sinon, si l'axe-étiquettes a pu terminer le mouvement initial pour 2 étiquettes (=timeout), on génère une alarme car une anomalie s'est alors produite dans la détection du défilement des étiquettes. Dans ce cas, Remise à zéro du bit MBB\_TEST\_ETIQUETTE. Passage à l'étape 10.

→ case else : dans tout autre cas, on affiche une erreur avec la valeur d'étape.

## L'IHM

Rappel : l' IHM étant maître dans la connexion Modbus, il peut accéder directement en lecture et écriture aux adresse Modbus de la carte InterpCNC.

Ecran PRODUCTION :



Les voyants Cellule Objet et Cellule Etiquette accèdent en lecture aux bits d'état des entrées 1 et 2.

Les champs de valeurs pour « Longueur Max Etiquette », « Position Etiquette », et sous « Recette », sont saisissables et accèdent en lecture/écriture aux adresses correspondantes en zone de recettes dans l'EEPROM de la carte (RCP\_NOM\_RECETTE, RCP\_LG\_MAX\_ETIQUETTE, et RCP\_POS\_ETIQUETTE). Idem pour Tempo Avance Objet, sur EE\_TEMPO\_AV\_OBJET.

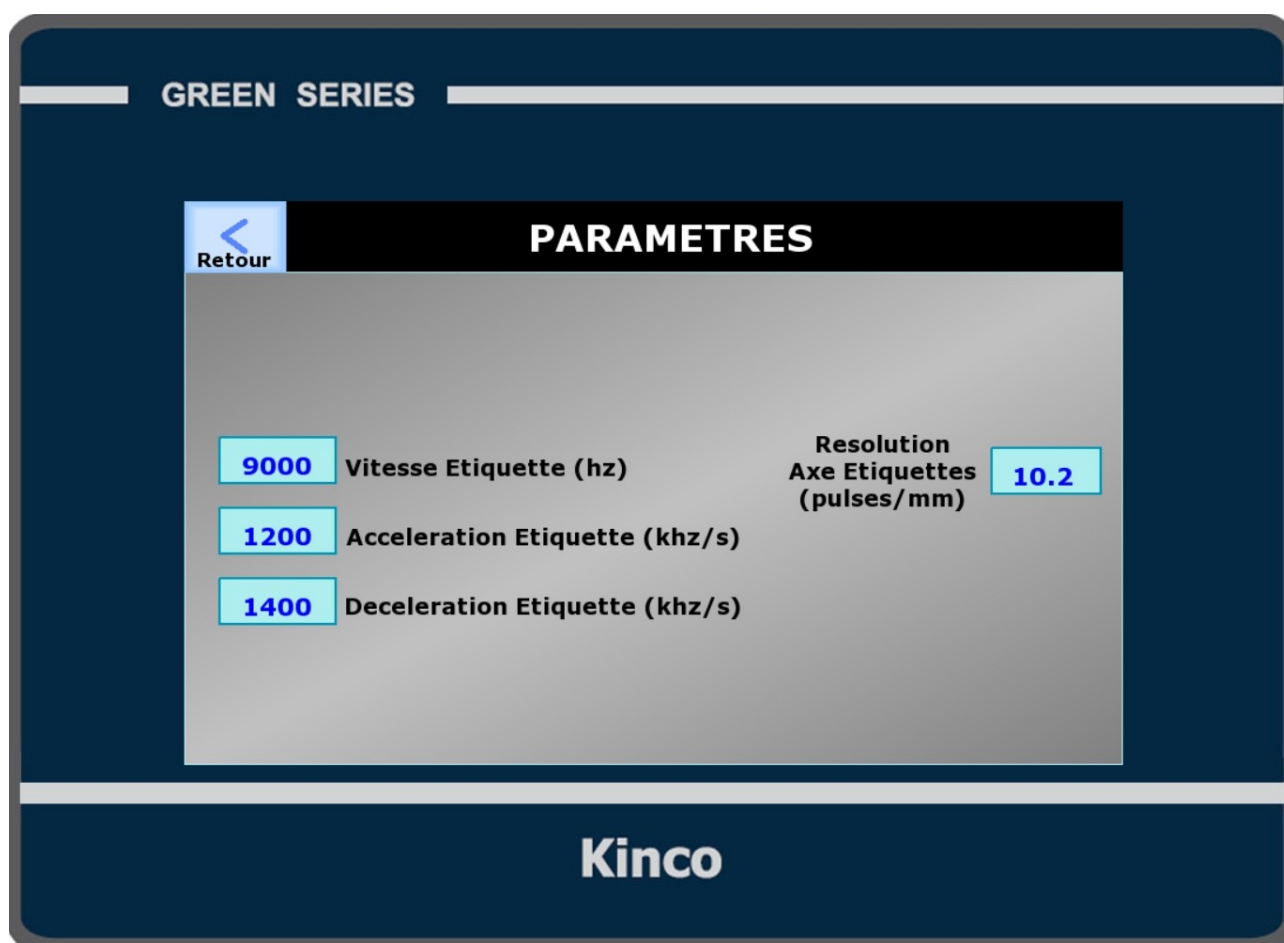
Les boutons « Convoyeur ON » et « DEPART CYCLE » sont de type 'toggle', c'est à dire qu'à chaque pression il font changer l'état des bits MBB\_ONOFF\_CONVOYEUR et MBB\_GCYCLE respectivement.

Les boutons « Test Etiquette » et « RAZ » sont de type 'Set 1', c'est à dire qu'à chaque pression ils forcent à 1 l'état des bits MBB\_TEST\_ETIQUETTE et MBB\_RAZ\_CPT, respectivement. Ces bits sont remis à zéro par le programme en fin d'exécution.

L'appui sur le bouton symbolisant un engrenage permet de passer à l'écran paramètres (Function key : Switch Window : Change Window : 10.Parametres).

Ce changement de fenêtre est soumis à mot de passe, nous appelons alors la « Login Window » : Control Setting/Conditional Enabling/ Security Level Minimum Level : 1  
et le mot de passe correspondant, en cliquant à la racine du projet sur l'écran : HMI Attribute → Security Levels Setting.

Ecran PARAMETRES :



Les champs de valeurs pour « Vitesse Etiquette », « Acceleration Etiquette », «Deceleration Etiquette » et « Resolution Etiquette » sont saisissables et accèdent en lecture/écriture aux adresses correspondantes dans l'EEPROM de la carte.

### **To do :**

- La gestion des alarmes n'a pas été traitée dans ce projet, vous pouvez juste forcer à zéro l'état du bit MBB\_ALARM\_ETIQ (bit utilisateur 324) pour acquitter l'alarme. Il conviendrait de le faire sur l'écran en y ajoutant une page d'alerte permettant aussi l'acquiescement de l'alarme.
- En l'état le compteur d'étiquettes ne travaille qu'en Ram (le compteur partiel est donc volatile). Une gestion d'un compteur total en EEPROM est restée commentée dans le programme, à vous de tester.