

► **Séance 1**

1. L'avant-projet

Cette étude est basée sur le thème « Chaîne de jointage et de tri de douelles » dont le cahier de charges est décrit dans le document « Theme-i1011-Jointage-Tri-Douelles.v***.odt ».

Le développé utile du plus grand tonneau qui peut être fabriqué est de 3,22 mètres.

1.1. Les composants d'automatisme

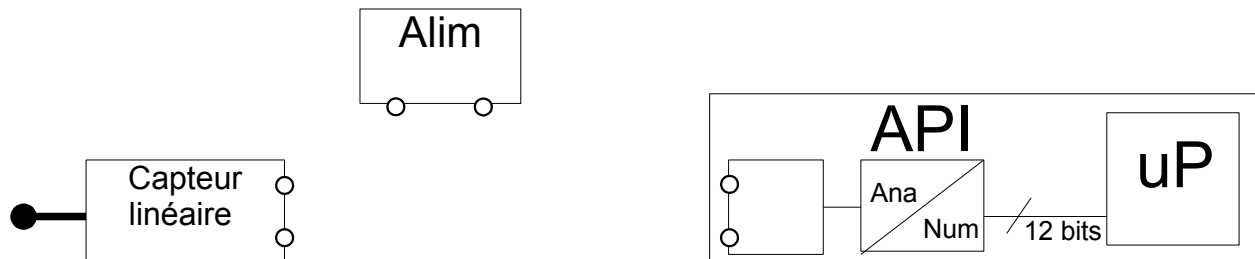
En entrée et en sortie de chaîne, un capteur de position linéaire permet de mesurer la largeur de la douelle.

Le modèle choisi dispose d'une gamme de mesure (« course ») de 0 à 200 mm maximum et d'une sortie analogique de type 4-20 mA 2 fils.

L'API dispose d'entrées analogiques 4-20mA avec conversion analogique/ numérique (C A/N) 12 bits unipolaire.

Sur le schéma ci-dessous :

- ◆ Placer et repérer (nommer) les bornes de connexion de chaque dispositif présenté ;
- ◆ tracer les connexions électriques nécessaires afin d'acquérir la mesure de largeur sur l'entrée API.



1.2. La préparation au choix de l'API et à la programmation

- ◆ Pour les largeurs de douelles admises en sortie de raboteuse, déterminer :
 - ✓ les valeurs mini et maxi des courants de mesure fournis par le capteur,
 - ✓ les valeurs numériques mini et maxi acquises par l'entrée analogique de l'API.
- ◆ Quelle est la précision absolue de la mesure de largeur par l'API en millimètres ? Est-elle acceptable ?
- ◆ Quel est le plus grand périmètre de tonneau dans la gamme à fabriquer ? A quelle valeur numérique correspond cette mesure, en prenant la même échelle que le capteur linéaire ? Cette valeur peut-elle être directement traitée par un API 16 bits de type Schneider qui ne sait travailler qu'en nombres entiers signés ?
- ◆ Proposer trois solutions pour permettre le traitement correct du développé de la série du plus gros tonneau.

2. Préparation de la programmation

- ◆ Afin de mettre en évidence les limites de stockage des mots 16bits, écrire un programme qui, chaque 1/10^e de seconde, augmente le développé (mot %MW1) de la valeur numérique acquise par le capteur pour une largeur de 60 mm, et incrémente %MW2 (comptage du nombre de douelles passées). On consultera l'aide en ligne (lancer Twidosoft et aller dans le menu « Aide », accéder à la rubrique « Bits système »). Tester le programme.
- ◆ Créer une table d'animation, afficher la valeur du développé %MW1 en décimal et commenter son évolution.
- ◆ Ajouter un bloc qui recopie %MW1 dans %MW3 et afficher %MW3 en binaire dans la table d'animation. Commenter.
- ◆ Compléter le programme afin d'augmenter en même temps un mot double %MD4 (des #60 mm), et commenter l'évolution de ce développé en format 32 bits dans la table d'animation.
- ◆ Modifier le programme pour faire évoluer le passage de douelle à chaque seconde (au lieu de 1/10 s) pour faciliter la lecture de l'animation.
- ◆ On envisage d'utiliser un API d'entrée de gamme ne gérant pas les doubles-mots. Modifier le programme afin de gérer le développé sur 2 mots distincts, l'un représentant les mètres accumulés, l'autre représentant le reste du développé inférieur à 1 mètre.

► **Séance 2**

3. L'étude

3.1. Séquencement et calcul numérique

Compléter le programme selon les fonctionnalités désirées ci-dessous :

Détecter l'arrivée d'une nouvelle douelle, acquérir sa largeur

La largeur de la douelle est acquise sur une entrée analogique en %IWm.v en la recopiant dans un mot interne %MW5.

Si la partie opérative n'est pas en place :

On simulera le passage d'une douelle par l'état du bit système qui change de valeur à une période de 1 seconde.

La valeur analogique acquise sur l'entrée sera remplacée par une valeur constante comprise entre les limites de largeurs de douelles admissibles (en points, ex. 1986).

Détecter si la largeur acquise correspond aux limites minimum et maximum admises

Éjecter dans le cas contraire la douelle par l'activation d'une sortie TOR

La sortie %Qm.v sera choisie sur le module TOR disponible sur votre API.

Si le module TOR n'est pas en place :

On activera un bit interne %Mi.

Calculer le développé par la méthode des 2 mots distincts (m | points)

On utilisera les mots %MW40 et %MW41.

Détecter si le développé à atteindre a été obtenu

La valeur du développé à atteindre est disponible dans 2 mots stockés aux adresses %MW50 pour les points et %MW51 pour les mètres.

Envoyer dans ce cas un ordre d'arrêt de la chaîne

L'ordre d'arrêt sera défini par l'allumage d'un voyant sur une sortie TOR %Qm.v

Si la partie opérative n'est pas en place :

L'ordre d'arrêt sera associé à l'activation d'un bit interne %M7. Ce bit interne à 1 sera utilisé pour inhiber la détection du bit système, et stopper ainsi la détection de l'arrivée des douelles.

Un nouveau cycle sera lancé par l'appui sur le bouton-poussoir Dcy en entrée TOR %Im.v.

Si la partie opérative n'est pas en place :

DCy sera simulé par la mise à 1 du bit %M9 dans la table d'animation.

La remise à 0 de ce bit %M9 sera assurée automatiquement par le programme.

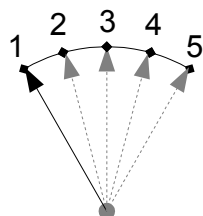
Amélioration de la lisibilité du programme

Nommer les objets mémoire, constantes et entrées/ sorties utilisés.

► **Séance 3**

Sélection des modèles de tonneaux de développé supérieur à 3 m

La valeur de consigne %MW50 représentant les centimètres du développé à atteindre doit « recevoir » la valeur choisie dans la table selon la position de l'index.



On installe sur la machine un sélecteur de modèle de tonneau (image ci-contre), qui va définir la valeur de développé à atteindre avant d'arrêter la production sur la série. Le câblage de ce sélecteur sur l'API et l'acquisition des entrées TOR correspondantes permettent de récupérer dans %MW55 une valeur d'index numérique de 1 à 5 par : %MW55:= %I0.0:3

La développé maximum à atteindre ne sera plus une valeur constante de 3 m et 9009 points, mais de 3 m et un nombre de points défini dans un tableau de constantes (ci-contre).

Adresse	Valeur
%KW60	
%KW61	1547
%KW62	4468
%KW63	7162
%KW64	9009
%KW65	15504

Les constantes font partie du programme et sont chargées dans l'API en même temps que celui-ci. Depuis le navigateur application, installer les valeurs ci-contre dans les constantes %KW61 à %KW65.

Voir dans l'aide les rubriques « Tables d'objets » et « Objets indexés ».

Écrire la ligne de programme qui permet de récupérer dans %MW50 la valeur du tableau sélectionnée selon l'index.

Sélection des modèles de tonneaux de tous développés

Le cas précédent ne permettait de sélectionner que des tonneaux de développé supérieur à 3 m. Or les différents modèles peuvent avoir des développés de 1 m à 3,44 m.

La table des développés va donc être modifiée pour contenir, pour chaque modèle, la limite décomposée en mètres d'une part et en points d'autre part (voir ci-contre).

Modifier le programme pour tenir compte de ces nouvelles données.

Adresse	Valeur
%KW70	
%KW71	
%KW72	1547
%KW73	1
%KW74	4468
%KW75	2
%KW76	7162
%KW77	2
%KW78	9009
%KW79	3
%KW80	15504
%KW81	3

Calcul numérique sur mots entiers

On revient sur une valeur de développé que l'on considère pouvoir stocker sur un seul mot.

Calculer le développé brut admis en entrée (%MW30), c'est à dire le développé net acquis en sortie (%MW20) augmenté de 7%.

Modifier %MW20 par une table d'animation, vérifier le calcul avec comme exemple %MW20 = 26000, essayer différentes valeurs.

Corriger l'algorithme si nécessaire.

Calcul numérique sur mots flottants (réels)

Génération de signal

► **Séance 4**

3.2. Gestion temporelle

Préparation

- ◆ Rechercher dans la documentation en ligne de PL7-Pro l'aide à propos des fonctionnalités liées au temps : retard à l'enclenchement, au déclenchement, blocs de temporisation, ...

Note : NE PAS tenir compte des rubriques à propos de « Série 7 » (anciens automates), fonctions réflexe, ou fonctions évoluées « FPULSOR », « FTON », ...

Retards

A l'enclenchement

Dans le cas de la mesure des largeurs de douelles, il faut commencer la mesure de largeur seulement 1 seconde après que la douelle a été détectée par le capteur de position.

- ◆ Écrire le programme qui met un bit interne à 1 avec un retard à l'enclenchement de 0,4 seconde après l'activation d'une entrée TOR %Im.0.
- ◆ Le programme proposé permet-il de s'assurer que le détecteur de douelle est resté activé pendant toute la durée du retard de 1 seconde ? Que se passe-t-il si le détecteur est relâché intempestivement au bout de 0,3s, puis réenclenché ?

Pour mettre en évidence le fonctionnement de la temporisation, on choisira une durée plus grande, de l'ordre de 10 secondes, et on affichera dans une table d'animation les paramètres du bloc temporisateur %TMi.P et %TMi.V.

Au déclenchement

- ◆ Ajouter un bloc temporisateur afin mettre en œuvre un retard au déclenchement de 3 secondes.

Impulsions

Par temporisateur

On désire signaler par un voyant que chaque douelle a bien été prise en compte.

- ◆ Écrire un programme qui active la sortie du voyant %Qm.5 pendant 2 secondes après que la douelle a été prise en compte.
- ◆ Paramétrer cette même durée de 2 secondes à partir de différentes bases de temps : 1s, 100ms, 10ms, ...

Par Monostable

On suppose que beaucoup de temporisateurs sont nécessaires dans notre application, et que les ressources en blocs %TM sont insuffisantes.

- ◆ Reproduire le fonctionnement 'Impulsion' à partir d'un bloc monostable %MN, si ce type de blocs est disponibles sur votre API.
- ◆ Mettre en évidence la différence de fonctionnement entre %TM et %MN :
 - Lorsque l'entrée dure plus longtemps que la tempo,
 - Lorsque l'entrée dure moins longtemps que la tempo,
 - Lorsque l'entrée passe à 0, puis repasse à 1 avant la durée de la tempo.
- ◆ Quel bloc peut être considéré comme 'redéclenchable' ?

Astables

Par temporisateur

On désire nettoyer très régulièrement les copeaux issus de l'usinage des merrains. Il faut pour cela commander une électrovanne d'air comprimé qui va chasser la sciure de bois du plan de travail, à raison d'une durée de 2 secondes à des intervalles de 4 secondes.

- ◆ A partir des blocs temporisateurs, générer sur %M5 un signal astable (« clignotant ») qui reste d'abord à l'état '1' durant 2 secondes, puis à l'état '0' durant 4 secondes, dès qu'un bit interne %M2 passe à 1 et tant qu'il y reste.

Par monostable

- ◆ Reproduire le même fonctionnement Astable à partir de blocs monostables %MN, si ces blocs sont disponibles sur votre API.

Par mots internes et bits système

- ◆ Générer le même signal clignotant (2s/4s) en n'utilisant ni %TM, ni %MN, mais en utilisant seulement des blocs opérations sur mots, des blocs comparateurs de mots, et les bits systèmes d'horloge.

Mesure de durée

On désire mesurer les temps de fonctionnement des moteurs associés aux pompes hydrauliques afin de déclencher des opérations de maintenance après un certain nombre d'heures de fonctionnement.

- ◆ Créer un programme qui cumule le temps de fonctionnement d'un moteur (défini par un bit interne %M10 à '1') en secondes.
- ◆ Permettre l'affichage du résultat sous la forme Heures-minutes-secondes, en partageant ces 3 données dans 3 mots distincts.
- ◆ Évaluer la durée maximum qui peut être stockée dans un double mot exprimé en secondes.