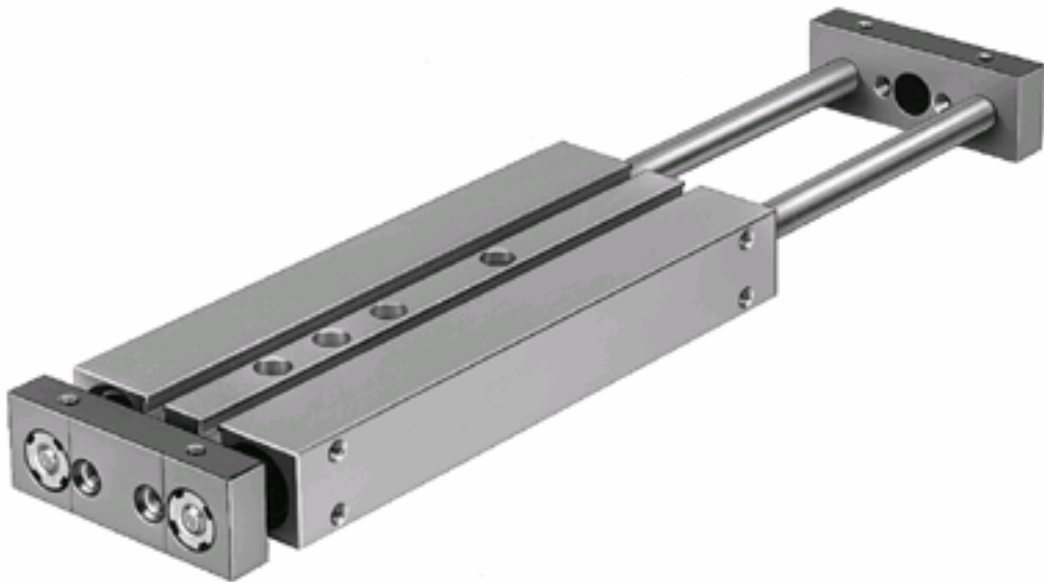


# Cours ENIP

- Applications API
- Langages API



Cours : M. Etienne

Date : oct. 2000

Version : v04

La table des matières :

<b>1. Logigramme structuré .....</b>	<b>2</b>
1.1. Introduction .....	2
1.2. Programmation d'une coulisse .....	2
1.3. Buts .....	3
1.4. Ecriture .....	3

## 1. Logigramme structuré

### 1.1. Introduction

Principalement en automatisation industrielle, nous sommes confrontés à réaliser 3 sortes de travaux :

- 1) Le développement de machines ou de processus à haute production. Exemple : vous fabriquez une machine automatique polyvalente à raison de 500 unités par année.
- 2) Le développement de machines ou de processus à raison de 500 unités par année, mais avec de légers différents entre chaque machine. Le client fait une demande pour diverses options qui lui sont propre.
- 3) Le développement de machines ou de processus comme prototypes, toutes différentes les unes des autres.

Dans notre application du "Logigramme structuré", nous nous trouverons dans une production industrielle décrite dans les points 1 et 2 précités.

### 1.2. Programmation d'une coulisse

Réaliser sur la page suivante le programme pour piloter une coulisse hydraulique.

- 1) Faire une copie du logigramme structuré
- 2) Mettre sur la copie les valeurs pour les entrées, les sorties et les mnémoniques
- 3) Comprendre le fonctionnement. Pour ce faire appuyer sur le start et le maintenir. Observer le fonctionnement. Ensuite faite d'autres variantes avec les touches Avance et Retour.
- 4) Expliquer chaque Label (uniquement son but) exemples : pourquoi c'est un set/reset, pourquoi start retour et retour, etc...

## 1.3. Buts

La mise en réalisation d'un programme n'est pas réalisée à l'aide du "Logigramme structuré". L'étude de la programmation est traitée dans les cours d'informatique et d'analyse. Dans notre cas précis, nous désirons écrire notre programme d'une telle manière qu'il nous sera très facile de reprendre le fil conducteur de ce même programme d'ici plusieurs semaines, plusieurs machines différentes, mais similaires ou encore plusieurs domaines d'activités différents.

Grâce à ce système, vous retrouverez votre "message" dans votre programme. Vous réaliserez le même programme sur une autre machine avec la conviction de sa crédibilité et de sa fonctionnalité.

Dans le cas où la machine serait légèrement différente, ou qu'une option viendrait se greffer sur l'original, vous n'aurez que peu de peine à manipuler, transformer, modifier votre programme avec une excellente compréhension et facilité.

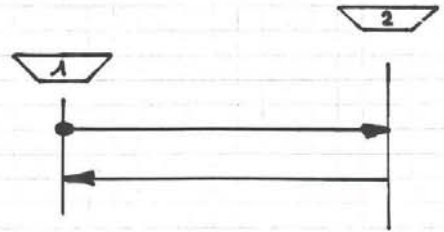
## 1.4. Ecriture

Selon l'exemple en annexe, réalisez votre soft de base selon les points suivants :

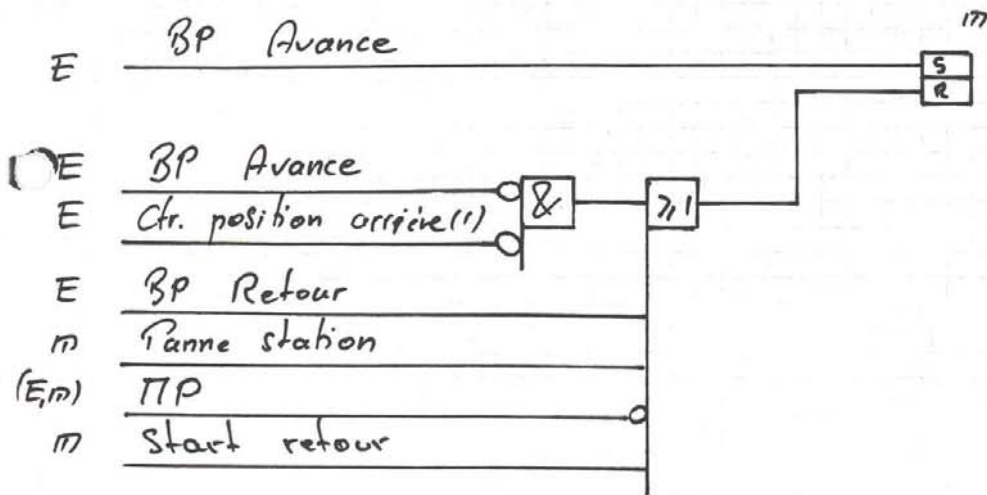
- a) Mettre un titre et faire un léger dessin si nécessaire.
- b) Sur le dessin, mettre les entrées et les sorties qui pourraient porter à confusion dans votre "Logigramme structuré".
- c) Garder une proportionnalité et un alignement de vos symboles
- d) Mettre un titre pour chaque label ou segment
- e) La sortie (côté droit) porte toujours le même nom que le titre du label
- f) Les noms des entrées (côté gauche) seront écrits sur leur ligne respective
- g) Nous mettrons une indication entrée (E ou I), sortie (O ou Q ou S) ou encore un marqueur interne (B ou M) éventuellement une Tempo ( $T_n, D$ ) etc...
- h) Ne jamais écrire de numérotation sur le document original

Pour faire une machine, faire une copie de l'original et compléter les informations propres au projet, tels que numérotation des entrées, des sorties, des marqueurs, des temporisations, etc...

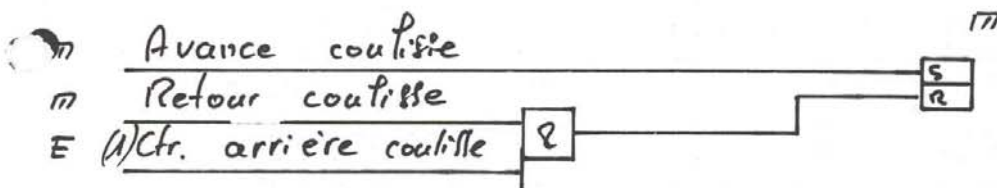
## Coulisse hydraulique



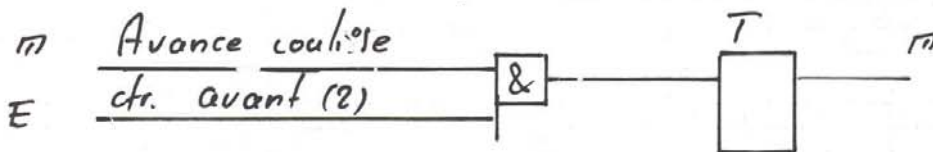
### L1 Avance coulisse



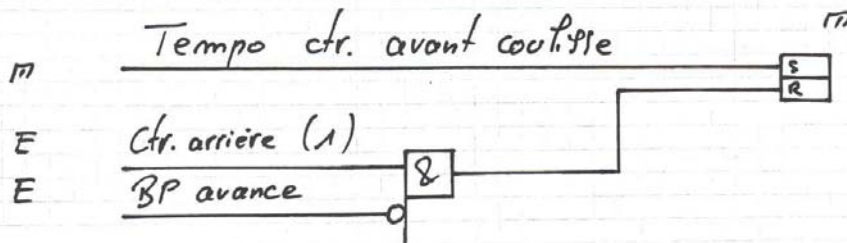
### L2 Mémo cycle station



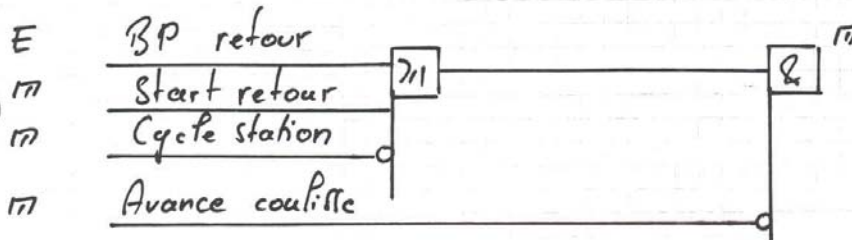
### L3 Tempo ctr. avant coulisse



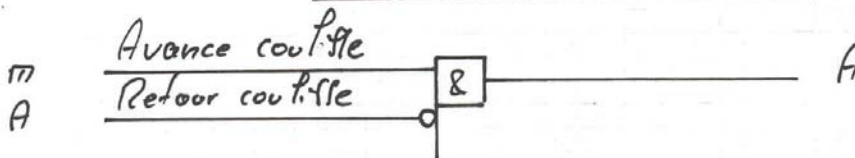
## L4 Start retour



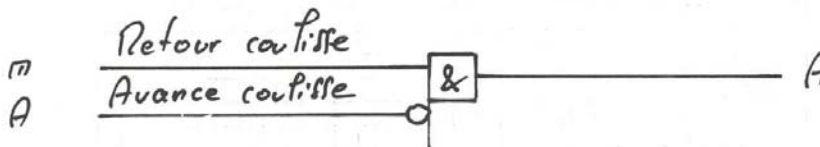
## L5 Retour coulisse



## L6 Sortie Avance coulisse



## L7 Sortie Retour coulisse



## L8 Panne station (pauze fin de course)

