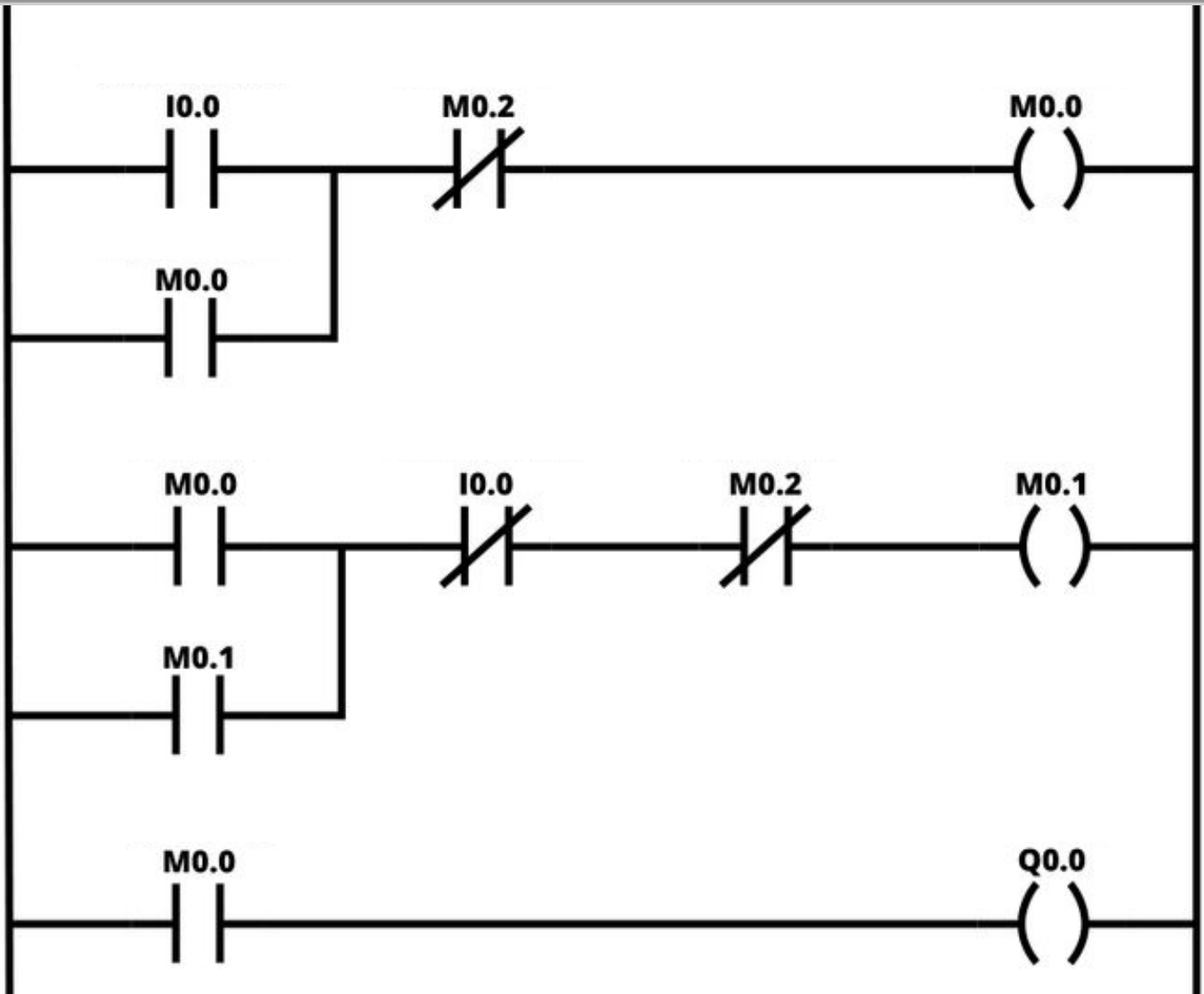


SEMAINE 4

LA PROGRAMMATION DES AUTOMATES

FICHE 16 : INTRODUCTION AU LANGAGE LADDER



Automation & Sense



Objectifs :

Après la consultation de cette fiche, vous serez en mesure:



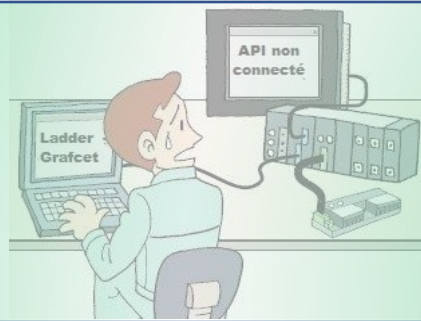
- De lire un schéma Ladder
- De concevoir un schéma Ladder

Les exercices d'applications viendront plutard. Nous utiliserons le logiciel TIA Portal pour la programmation en langage Ladder. Cependant, si vous ne disposez pas du logiciel TIA Portal, vous pourrez télécharger le logiciel gratuit Codesys.



SOMMAIRE

- I) GENERALITES
- II) LES BASES DU LANGAGE LADDER
- III) QUELQUES REGLES A RESPECTER LORS DE LA CONCEPTION D'UN SCHEMA LADDER
- IV) CONVERSION DE PORTES LOGIQUES EN LANGAGE LADDER



I) GENERALITES

Le langage Ladder ou langage à contacts ou encore schéma à relais fait partie des cinq langages standards (Ladder, ST, IL, Grafcet, FBD) de la norme CEI 61131-3 défini par la commission d'électrotechnique internationale (CEI). Le langage Ladder est un langage de programmation graphique facile à comprendre et à prendre en main. C'est sans doute le langage de programmation d'automatisme le plus couramment utilisé pour la programmation d'automates. Le langage Ladder est composé d'une séquence de contacts (interrupteurs qui sont soit fermés, soit ouverts) et de bobines qui permettent de traduire les états logiques d'un système. Un schéma Ladder se lit de gauche à droite et de haut en bas.

Même si le langage Ladder est un langage standard, on peut retrouver quelques différences en fonction des logiciels de programmation d'automate. Par exemple, on peut trouver certaines fonctions qui existent sur le logiciel TIA Portal de Siemens et pas sur Studio 5000 de Rockwell.

Dans les formations que nous allons fournir, nous allons nous concentrer sur TIA Portal vu que les automates Siemens sont les automates les plus utilisés en Europe.

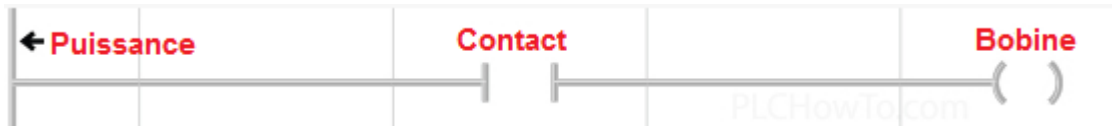
Les logiciels de programmation d'automates fonctionnent quasiment de la même manière, si vous maîtrisez TIA Portal, avec un peu d'adaptation et de pratique, vous pourrez facilement vous familiariser avec Unity Pro de Schneider ou Studio 5000 de Rockwell.

I) Les bases du langage Ladder

Le langage Ladder fonctionne quasiment de la même manière qu'un schéma de commande électrique. Il est composé de contacts et de bobines. Selon la logique définie par les contacts qui représentent les entrées, on pourra activer ou désactiver les bobines qui représentent des sorties. Par exemple, on peut affecter une bobine d'un schéma Ladder à une sortie d'un automate qui va commander à son tour une électrovanne ou un voyant.

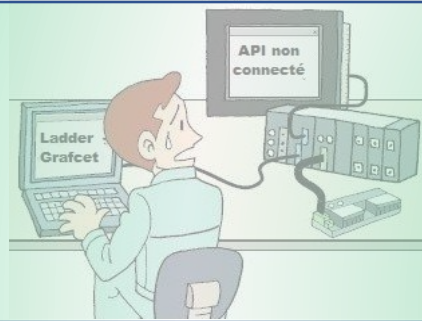


Dans l'image ci-dessous, un appuie sur "contact" laisse passer le courant sur la ligne ce qui permet d'exciter la bobine.



Il existe plusieurs types de contacts selon la logique voulue :

- | |— Contact normalement ouvert
- |/|— Contact normalement fermé
- |↑|— ou —|P|— Contact agissant sur front montant
- |↓|— ou —|N|— Contact agissant sur front descendant
- |<|— Contact comparatif infériorité
- |>|— Contact comparatif supériorité
- |≤|— Contact inférieur ou égal
- |≥|— Contact supérieur ou égal
- |=|— Contact égalité
- |≠|— Contact différent de



Et plusieurs types de bobines :

- () — **Le résultat de la fonction logique active la bobine**
- (S) — **Le résultat de la fonction logique met à "1" la bobine (Set)**
- (R) — **Le résultat de la fonction logique met à "0" la bobine (Reset)**
- (/) — **L'inverse du résultat de la fonction logique active la bobine**

II) Quelques règles à observer lors de la conception d'un schéma Ladder

- Au niveau d'une même ligne d'un schéma Ladder, on ne peut pas avoir deux sorties en série, on ne peut avoir que des sorties montées en parallèles. Par contre on peut avoir plusieurs contacts d'entrées qui peuvent être montés en séries comme en parallèles.

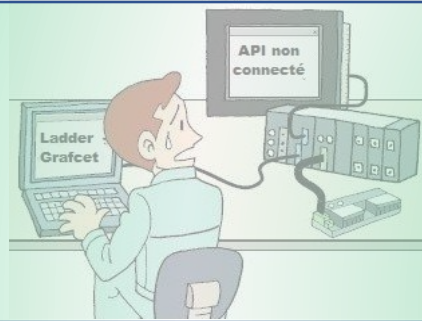
Ce qu'il ne faut pas faire :



Ce qui est possible de faire :



- Sur un schéma Ladder, les contacts d'entrée sont toujours à gauche et les sorties ou bobines toujours à droite.



III) Conversion de portes logiques en langage Ladder

Le "ET logique"

Dans le cas du "ET logique", les deux contacts A et B sont placés en série. La bobine X n'est active que si les contacts A et B sont simultanément fermés.

ET logique		
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

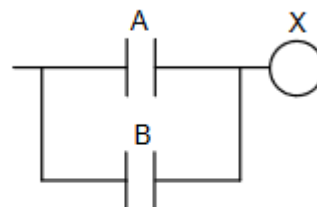


$$X = A \cdot B$$

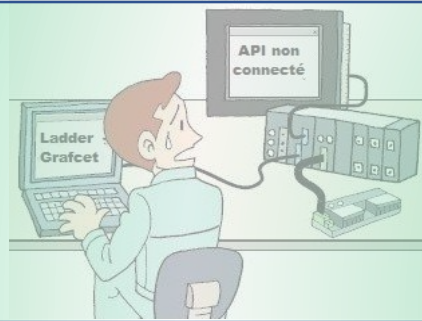
Le "OU logique"

Dans le cas du "Ou logique", les deux contacts A et B sont placés en parallèle. La bobine X est activée soit par appuie sur le contact A soit par appuie sur le contact B.

OU logique		
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1



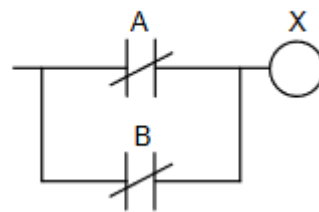
$$X = A + B$$



Le NAND (Non – ET)

Comme son nom l'indique, le **NAND** ou **Non-ET** est un « ET logique » inversé. Ici la sortie X est activée si l'inverse de l'entrée A ou l'inverse de l'entrée B est égal à « 1 ».

NAND		
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0



$$X = \overline{A \cdot B}$$

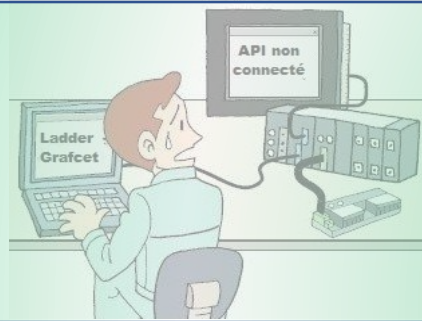
Le NOR (NON OU)

Le **NOR** ou « **NON-OU** » est un « OU logique » inversé. Ici l'entrée X est activée si l'entrée A et l'entrée B sont simultanément égales à « 0 ».

NOR		
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0



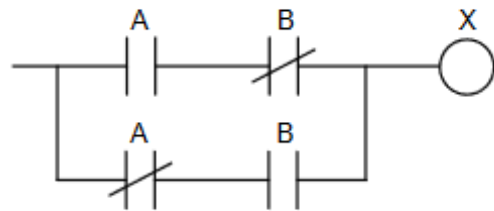
$$X = \overline{A + B}$$



Le OU Exclusif

Ici la sortie X est activée si l'entrée A est égale à 1 ou l'entrée B est égale à 1. Si les deux entrées A et B sont égales à 1 de manière simultanée, cela n'active pas la sortie X (c'est pourquoi on parle de « OU Exclusif »).

OU EXCLUSIF		
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

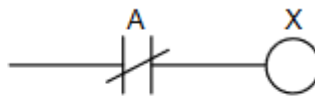


$$X = (\bar{A} \cdot B) + (A \cdot \bar{B})$$

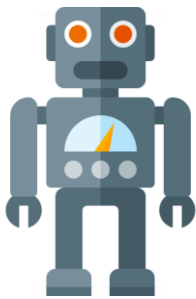
LE « NOT logique »

Le rôle du NOT est d'inverser la variable logique qui lui est passée en entrée. Ici la sortie X est activée si l'entrée A est égale à 0.

NOT	
A	X
0	1
1	0



$$X = \bar{A}$$



Dans cette fiche, vous avez pu découvrir le langage Ladder.

Nous verrons les autres fonctions du Langage Ladder, lorsque nous aborderons la programmation d'automate sur TIA Portal.