

CH3-A-TP3

L'opérateur Mémoire

➤ *En autonomie : Lire et compléter pages 1 à 3 et répondre aux questions Q1 page 4 et 5.*

Introduction :

Dans tout système on trouve un bouton de mise en marche et un bouton d'arrêt.
Si ces boutons sont des poussoirs cela implique **de garder en mémoire** l'état du dernier bouton actionné si l'on veut assurer un bon fonctionnement du système .

Nous allons étudier ici les différents modes de commande :

- à MARCHE PRIORITAIRE
- à ARRET PRIORITAIRE

Dans un premier temps on étudiera les systèmes à relais électromagnétique (schéma à contacts) et cela nous conduira vers la réalisation de circuits mémoire en électronique intégrée.

Désignation

bouton de mise en marche noté : **m**

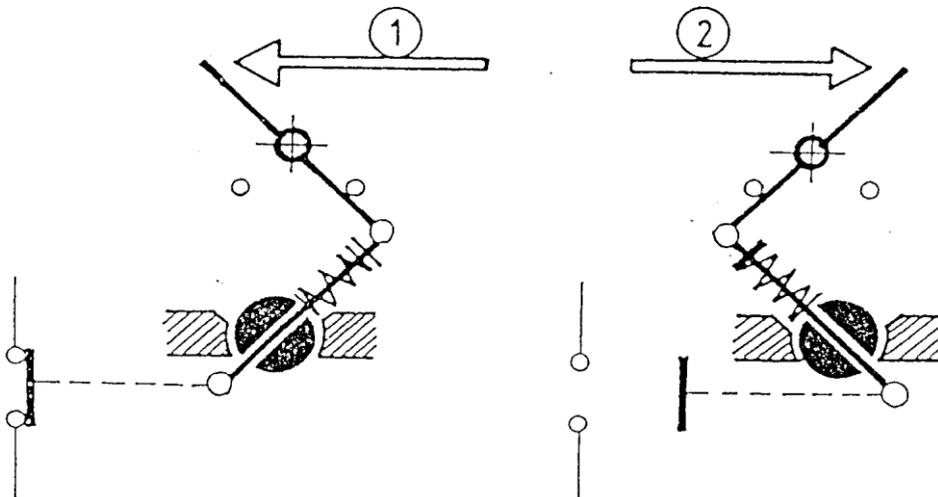
bouton d'arrêt noté : **a**

Différents types de mémoire

1 mémoire mécanique

C'est le cas de l'interrupteur bistable (à levier).

- 1 information de mise en mémoire;
- 2 information d'effacement de la mémoire.



2 mémoire électromécanique

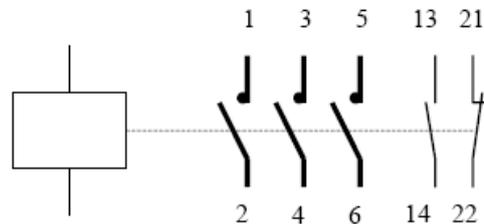
➤ Le relais électromagnétique (ou contacteur)

Fonction

Il permet d'établir ou d'interrompre l'alimentation d'un circuit électrique.

Un contacteur est pourvu en général de contacts de puissance (rep1..6), d'un ou plusieurs contacts auxiliaires de commande (rep13..22), et de 2 bornes d'alimentation de sa bobine interne.

Symbole

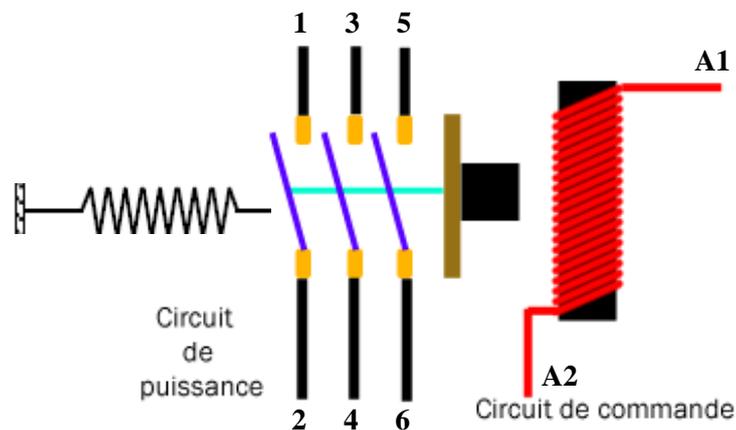


Principe de fonctionnement

Lorsque sa bobine est alimentée, elle crée un champ magnétique qui attire les contacts liés mécaniquement et ainsi ferme le circuit.

La bobine est généralement branchée sur un circuit de commande.

Schéma de principe:



➤ Soit les montages suivants:

schéma 1

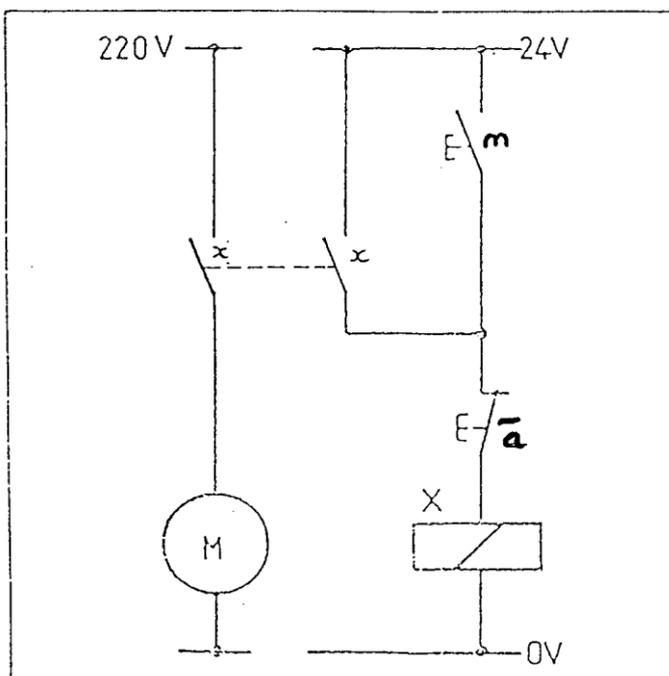
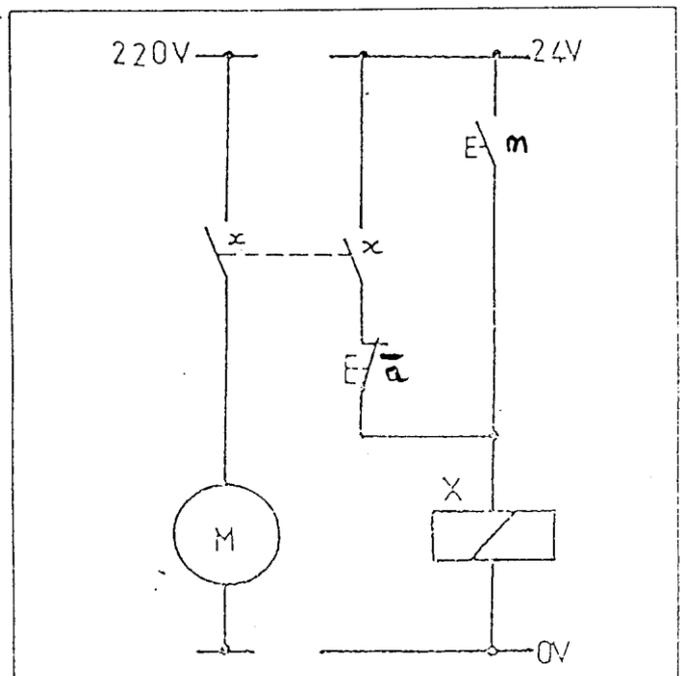


schéma 2



Question :

Compléter les phrases suivantes décrivant le fonctionnement général des montages ci-dessus.

Si l'on appui sur le contact "m", la bobine X est sous une tension de 24 V.
Celle ci les contacts "x" et le moteur M est alimenté sous une tension de V.

Le bouton poussoir "m" est donc le bouton de du moteur.

Quand le bouton "m" est relâché le courant continue de passer par l'intermédiaire des contacts "...." et "....". On dit qu'il y adu moteur.

Si l'on appui sur le poussoir "a", la bobine X alimentée, alors les contacts "...." s'ouvrent et l'auto alimentation du moteur M est supprimée.

Le bouton poussoir "a" est donc le bouton d' du moteur.

Question :

Que se passe t-il pour chacun des montages ci dessus si on appui simultanément sur les boutons poussoirs "a" et "m" ?

Schéma 1 :

L'appui simultané sur "m" et "a" entraîne une de l'alimentation de la bobine etdu moteur; c'est

donc un système dit à :

Schéma 2 :

L'appui simultané sur "m" et "a" entraîne de la bobine et par conséquent du moteur;

c'est donc un système dit à :

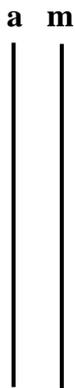
A) Etude de la MARCHE PRIORITAIRE

Travail demandé :

1. Rappeler l'équation de la commande de la bobine:

X =

2. Etablir le logigramme de l'équation ci dessus.

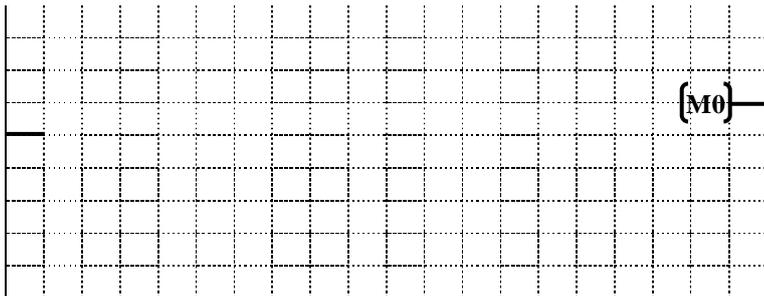


3. Etablir le diagramme à échelle correspondant à l'équation ci dessus et programmer en langage Ladder sous PL7-pro.

Affectations :

a ↔ %I1.0 ; m ↔ %I1.1 ;

Q ↔ %M0 ;



4. Transférer votre programme dans l'API puis créer une table d'animation vous permettant de compléter la table de vérité ci-contre

5. Simuler et compléter la table en respectant scrupuleusement l'enchaînement des combinaisons.

a	m	Q
%I1.0	%I1.1	%M0
1	1	
1	0	
0	0	
0	1	
0	0	
1	0	
1	1	

- Entourer en noir la combinaison 1 1.
- Entourer en rouge la combinaison qui mémorise l'état précédent;
- on en déduit:
 - en bleu la combinaison de mise à "1" de Q
 - en vert la combinaison de mise à "0" de Q

- Que peut-on dire de la combinaison 1 1 ?

- Si l'on note l'état précédent Q_{n-1} , en déduire la table de vérité réduite de la mémoire

Synthèse

Symbole général:

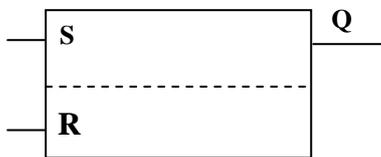


Table de vérité :

R	S	Q_n
0	0	Q_{n-1}
0	1	
1	0	
1	1	

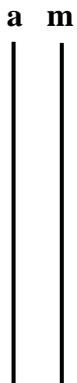
B) Etude de l'ARRET PRIORITAIRE

Travail demandé :

1. Rappeler l'équation de la commande de la bobine:

$$X =$$

2. Etablir le logigramme de l'équation ci dessus.

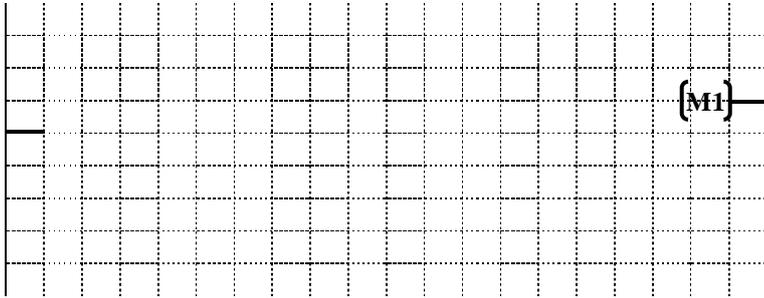


3. Etablir le diagramme à échelle correspondant à l'équation ci dessus et programmer en langage Ladder sous PL7-pro.

Affectations :

a ↔ %I1.0 ; m ↔ %I1.1 ;

Q ↔ %M1 ;



4. Transférer votre programme dans l'API puis créer une table d'animation vous permettant de compléter la table de vérité ci-contre

5. Simuler et compléter la table en respectant scrupuleusement l'enchaînement des combinaisons.

a	m	Q
%I1.0	%I1.1	%M1
1	1	
0	1	
0	0	
1	0	
0	0	
0	1	
1	1	

- Entourer en noir la combinaison 1 1.
- Entourer en rouge la combinaison qui mémorise l'état précédent;
- on en déduit:
 - en bleu la combinaison de mise à "1" de Q
 - en vert la combinaison de mise à "0" de Q
- Que peut-on dire de la combinaison 1 1 ?

➤ Si l'on note l'état précédent Q_{n-1} , en déduire la table de vérité réduite de la mémoire

Synthèse

Symbole général:

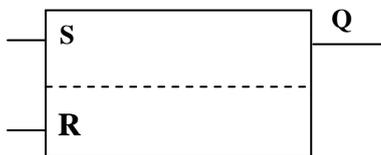


Table de vérité :

R	S	Q_n
0	0	Q_{n-1}
0	1	
1	0	
1	1	

Synthèse générale

Symbole général:

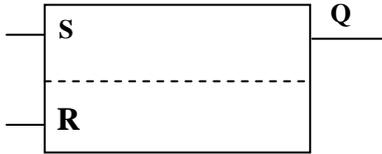
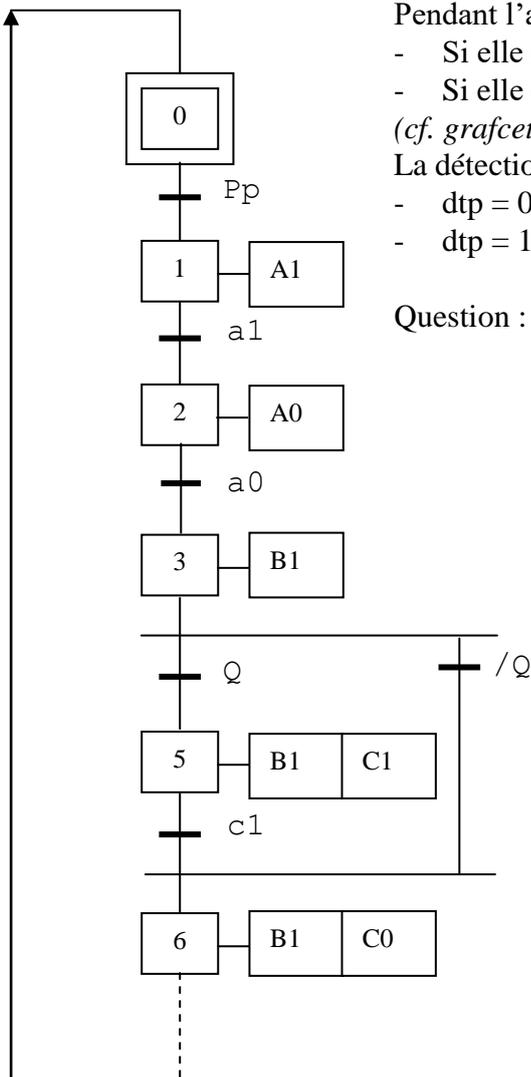


Table de vérité :

R	S	Q_n
0	0	Q_{n-1}
0	1	
1	0	
1	1	

Exemple d'application :



Pendant l'action A1 on doit mémoriser le type de pièce :

- Si elle est de type 1 Q reste à 0 et on ne lui applique pas l'action C1.
- Si elle est de type 2 Q passe à 1 et on lui applique l'action C1.

(cf. grafcet ci-contre)

La détection du type de pièce se fait grâce à un détecteur nommé dtp :

- dtp = 0 pour type 1
- dtp = 1 pour type 2

Question : Déterminer les conditions de mise à 1 et à 0 de la mémoire.

