

LYCEE TECHNIQUE D'ABIDJAN-COCODY	PROGRAMMATION DES AUTOMATES SYSMAC C28H EN LANGAGE MNEMONIQUE	AUTOMATISME
TE-TF1		DOC : 1/12

1-DEFINITION

Le schéma à contacts ne peut être introduit directement dans l'API au moyen d'une console de programmation, il faut donc un SSS (SSYSWIN). Pour l'introduire à l'aide d'une console de programmation, il faut convertir le schéma à contacts en code mnémonique.

Le code mnémonique fournit exactement la même information que le schéma à contacts mais sous une forme qui peut être tapé directement dans l'API.

En effet, le langage en code mnémonique est une suite ou liste d'instructions constituées de fonctions logiques (LD, AND, OR) ; en réalité, un programme direct en code mnémonique est possible bien que ce ne soit pas recommandé pour des débutants ou pour des programmes complexes. Et quelque soit le périphérique de programmation utilisé, le programme est mis en mémoire sous forme mnémonique, ce qui fait qu'il est important de comprendre ce code.

2-PRINCIPE D'ECRITURE DU PROGRAMME

2-1- Les instructions à contacts

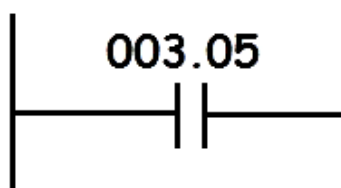
Pour comprendre complètement un programme, nous présenterons et décriront le code mnémonique en même temps que les schémas à contacts.

Les instructions à contacts correspondent aux conditions qu'elles soient indépendantes ou combinées avec le bloc logique ; elles fournissent les conditions sur lesquelles toutes les autres instructions sont fondées.

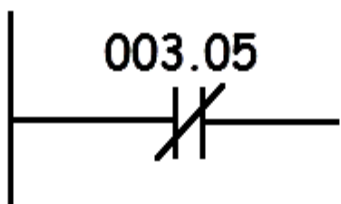
2-1-1- Les instructions LOAD (LD) et LOAD NOT (LD NOT)

La première condition qui commence un bloc logique quelconque dans un schéma à contacts correspond à une instruction LOAD ou LOAD NOT.

Chacune de ces instructions demande une ligne de code mnémonique.



Instruction	Adresse



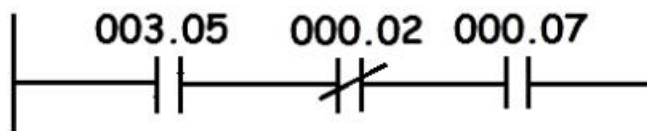
Instruction	Adresse

LYCEE TECHNIQUE D'ABIDJAN-COCODY	PROGRAMMATION DES AUTOMATES SYSMAC C28H EN LANGAGE MNEMONIQUE	AUTOMATISME
TE-TF1		DOC : 2/12

2-1-2- Les instructions AND et AND NOT

Quand deux ou plusieurs conditions sont en série sur la même ligne secondaire, la première correspond à une instruction LOAD ou LOAD NOT ; et le reste des conditions à des instructions AND ou AND NOT.

Chacune de ces instructions demande une ligne de code mnémonique.

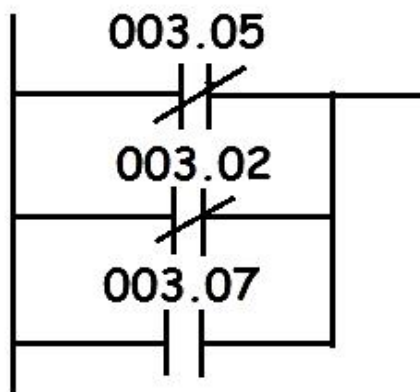


Instruction	Adresse

2-1-2- Les instructions OR et OR NOT

Quand deux ou plusieurs conditions sont en série sur des lignes secondaires séparées mais parallèles et se rejoignent ensuite, la première correspond à une instruction LOAD ou LOAD NOT ; et le reste des conditions à des instructions OR ou OR NOT.

Chacune de ces instructions demande une ligne de code mnémonique.



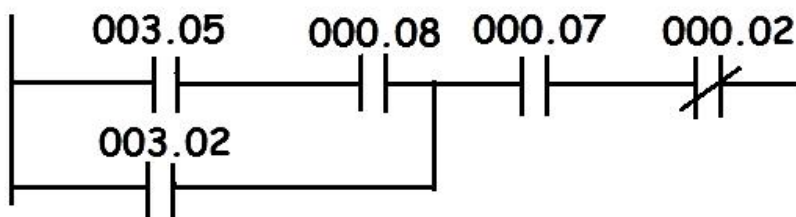
Instruction	Adresse

2-1-3- Combinaisons des instruction AND et OR

Si des instructions AND et OR sont combinées dans des schémas plus compliqués, elles peuvent être parfois considérées individuellement, chaque instruction effectuant une opération logique sur la condition d'exécution et l'état du bit opérande.

Ici il faut surtout comprendre que les résultats des opérations précédentes détermine les conditions d'exécution pour l'instruction suivante.

LYCEE TECHNIQUE D'ABIDJAN-COCODY	PROGRAMMATION DES AUTOMATES SYSMAC C28H EN LANGAGE MNEMONIQUE	AUTOMATISME
TE-TF1		DOC : 3/12



Instruction	Adresse

2-2- Les instructions de blocs logiques

Un bloc logique est tout groupe de conditions qui vont ensemble pour créer un résultat logique. Ils commencent par une instruction LOAD ou LOAD NOT.

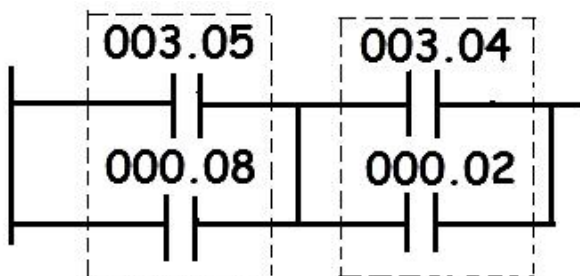
- Les instructions AND LOAD (AND LD) ou OR LOAD (OR LD)

Les instructions de bloc logique décrivent des relations entre des blocs logiques.

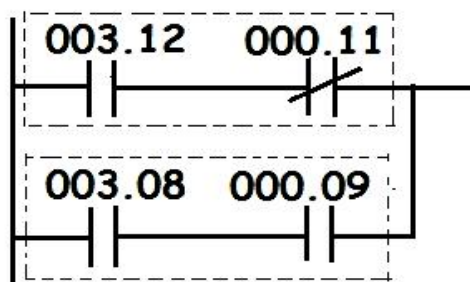
- L'instruction AND LOAD fait l'intersection logique (mise en série) des conditions d'exécutions produites par deux blocs logiques.
- L'instruction OR LOAD fait la réunion logique (mise en parallèle) des conditions d'exécutions produites par deux blocs logiques.

C'est une façon de faire des blocs logiques indépendamment et puis de combiner les résultats. Les instructions AND LOAD et OR LOAD ne nécessitent aucun bit opérande particulier parce qu'elles fonctionnent sur des conditions préalablement déterminées.

Chaque bloc est codé en utilisant une instruction LOAD ou LOAD NOT pour coder la première condition, puis AND LOAD ou OR LOAD est utilisée pour combiner les blocs.



Instruction	Adresse



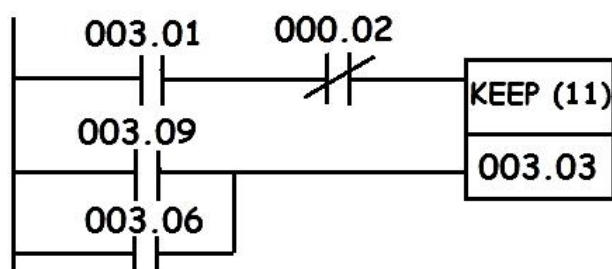
Instruction	Adresse

2-3- L'instruction KEEP

L'instruction KEEP est utilisée pour maintenir l'état du bit opérande sur la base de deux conditions d'exécution.

L'instruction KEEP est connectée à deux lignes secondaires S (Set) et R (Reset).

Comme pour toutes les instructions qui demandent plus d'une ligne secondaire, les lignes secondaires sont codées d'abord, avant l'instruction qu'elles commandent.



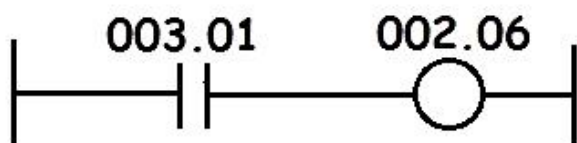
Instruction	Adresse

2-4- Les instructions OUT et OUT NOT

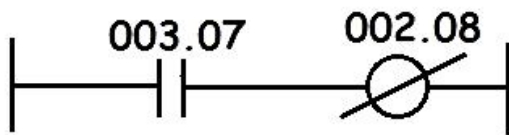
Avec l'instruction OUT, l'opérande sera mis à l'état 1 aussi longtemps que la condition d'exécution est à l'état 1 et sera mis à l'état 0 aussi longtemps que la condition d'exécution est à l'état 0.

Avec l'instruction OUT NOT, l'opérande sera mis à l'état 1 aussi longtemps que la condition d'exécution est à l'état 0 et sera mis à l'état 0 aussi longtemps que la condition d'exécution est à l'état 1.

E code mnémorique, chacune de ces instructions demande une ligne



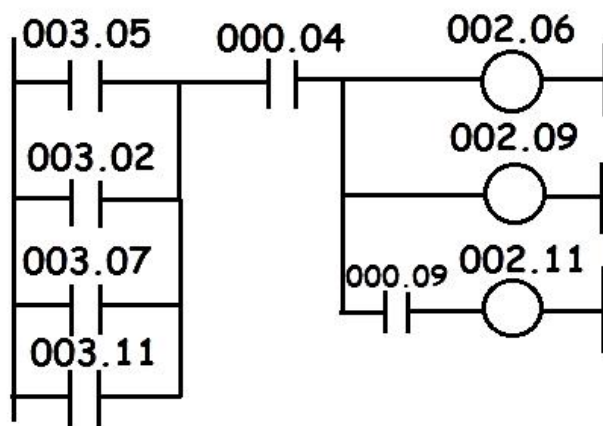
Instruction	Adresse



Instruction	Adresse

- Codage d'instructions multiples de droite

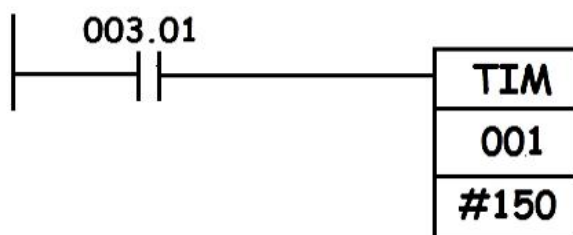
S'il ya plus d'une instruction de droites exécutées avec la meme condition, elles sont codées consécutivement en suivant la dernière condition de la ligne secondaire.



Instruction	Adresse

2-5- Les instructions TIM et CNT

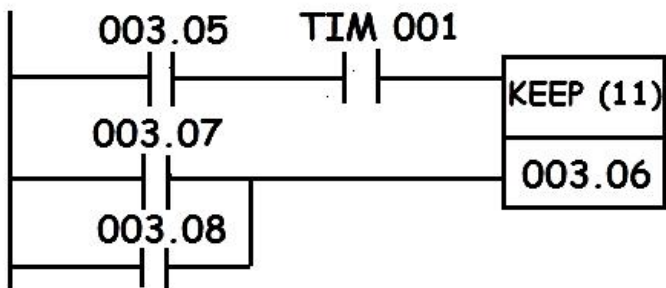
- Une temporisation est activée quand sa condition d'exécution passe à l'état 1 puis est réinitialisée (à la valeur de consigne SV) quand sa condition d'exécution passe à l'état 0. Une fois activée, TIM se mesure en unités de 0,1 seconde à partir de la valeur de consigne SV.
- Le terme « TIM 001 » peut désigner une instruction (cas de TIM en tant que variable interne) :



Instruction	Adresse

LYCEE TECHNIQUE D'ABIDJAN-COCODY	PROGRAMMATION DES AUTOMATES SYSMAC C28H EN LANGAGE MNEMONIQUE	AUTOMATISME
TE-TF1		DOC : 6/12

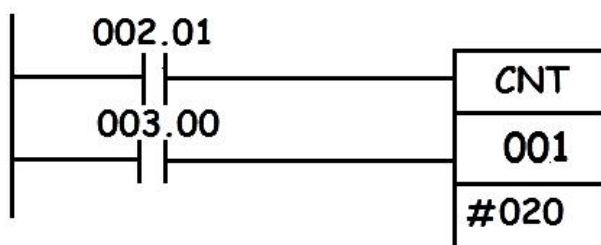
- Le terme « TIM 001 » peut désigner un bit opérande (cas de TIM en tant que contact « réceptivité ») : C'est le drapeau de fin du temporisateur.



Instruction	Adresse

- Le compteur CNT est utilisé pour déclencher un comptage dégressif à partir de la valeur de consigne SV quand la condition d'exécution de l'impulsion de comptage CP passe de l'état 0 à l'état 1. La valeur en cours est (PV) est décrémentée de un à chaque fois que CNT est exécuté avec une condition d'exécution à l'état 1 pour CP. CNT est remis à zéro par l'entrée de remise à 0 (R). Lorsque R passe de l'état 0 à l'état 1, la PV est remise à la valeur de SV. Le comptage dégressif à partir de la SV redémarre lorsque R est passé à l'état 0. La valeur en cours PV n'est pas remise à zéro dans le cas d'une coupure de courant.

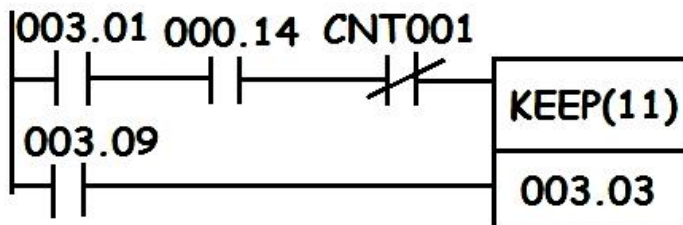
Le terme « CNT 001 » peut désigner une instruction (cas de CNT en tant que variable interne) :



Instruction	Adresse

LYCEE TECHNIQUE D'ABIDJAN-COCODY	PROGRAMMATION DES AUTOMATES SYSMAC C28H EN LANGAGE MNEMONIQUE	AUTOMATISME
TE-TF1		DOC : 7/12

- Le terme « CNT 001 » peut désigner un bit opérande (cas de CNT en tant que contact « réceptivité ») : C'est le drapeau de fin du compteur.



Instruction	Adresse

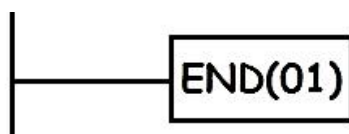
2-6- L'instruction END

La dernière instruction nécessaire pour terminer un programme simple est l'instruction END.

Quand l'Unité Centrale balaye le programme, elle exécute toutes les instructions jusqu'à la première instruction END avant de retourner au début du programme et de recommencer l'exécution.

Lorsqu'il existe un sous-programmes, END est placé après le dernier sous-programme. En l'absence d'instruction END dans le programme, il ne sera pas exécuté du tout et le message d'erreur « NO END INST » apparait.

L'instruction ne demande pas d'opérandes et aucune condition ne peut être placée avec elle sur la même ligne secondaire.



Instruction	Adresse

- GRAFCET de programmation (voir voir cours : Programmation en langage LADDER).
- Tableau d'activation et de désactivation des étapes (idem)
- Equations de pilotage des préactionneurs (idem)

[illegible]

[illegible]

