

Sommaire

Chapitre: Programmation et Simulation des API Omron CQM1	2
I. Description logicielle	3
II. CX-Programmer pour Omron	3
1. Création d'un nouveau projet :	4
2. Compilation et contrôle d'un programme :	10
3. Mise en ligne :	10
4. Transfert de programmes :	11
Chapitre: Eléments de programmation des API Omron	13
I. Introduction	14
II. Les principes des schémas à relais	14
1. Exemple de programmation :	15
a. Programmation d'une fonction ET logique :	15
b. Programmation d'une fonction OU logique :	15
Exercice 1 :	15
Exercice 2 :	15
Exercice 3 :	15
Exercice 4 :	15
Exercice 5 :	15
Exercice 6 :	15
Exercice de synthèse :	15

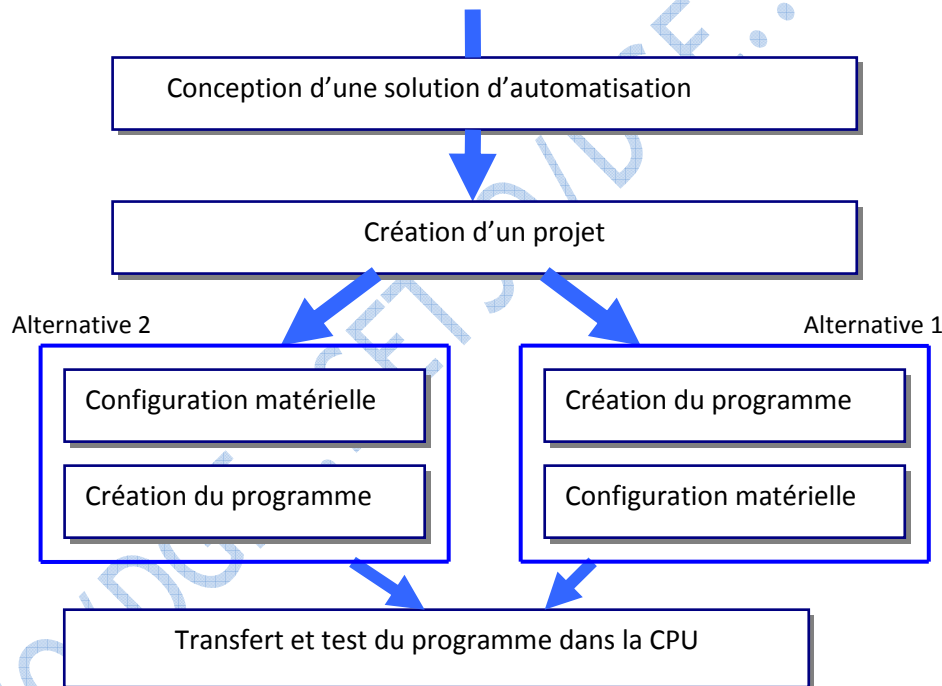
Chapitre

Programmation et Simulation des API Omron CQM1

I. Description logicielle

Les logiciels de base pour la configuration et la programmation de systèmes d'automatisation existent en plusieurs variétés, selon la marque et la gamme de l'automate programmable industriel. Les logiciels à présenter ci-dessous sont choisis conformément aux API disponibles :

- ✚ **STEP7-Micro/DOS** et **STEP7-Micro/Win** pour des applications autonomes simples sur Siemens **S7-200** ;
- ✚ **STEP7-SIMATIC Manager** pour des applications sur **S7-300/400**, **M7-300/400** et **C7-300/400** présentant des fonctionnalités supplémentaires ;
- ✚ **STEP7-TIA PORTAL** pour la gamme des API Siemens **S7-1200/1500** et aussi **S7-300** ;
- ✚ **CX-Programmer** pour programmer des API **OMRON** ;
- ✚ **GX Developer** pour des applications sur des API **Mitsubishi**.



II. CX-Programmer pour Omron

CX-Programmer constitue une plate-forme logicielle API commune pour tous les types d'automates programmables Omron, des micro-API aux systèmes à processeurs en duplex.

Il permet une conversion facile et la réutilisation de code API entre différents types d'API ainsi que la réutilisation totale des programmes de contrôle créés par une autre génération de logiciels de programmation d'API.

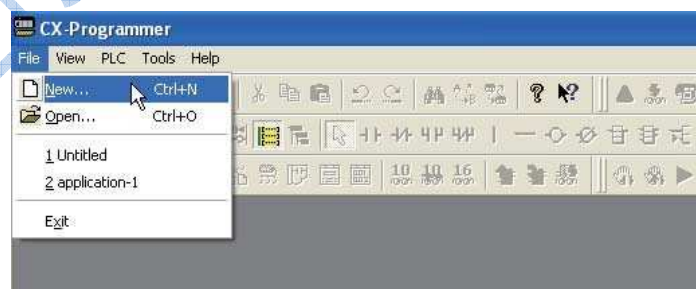


CX Programmer est un outil de programmation d'API pour créer, tester et procéder à la maintenance de programmes associés avec les API Omron des séries CS/CJ, CV et CQM. Il propose des installations de support d'API, des informations d'adresse et de communication avec les API Omron et leurs types de réseau supportés.

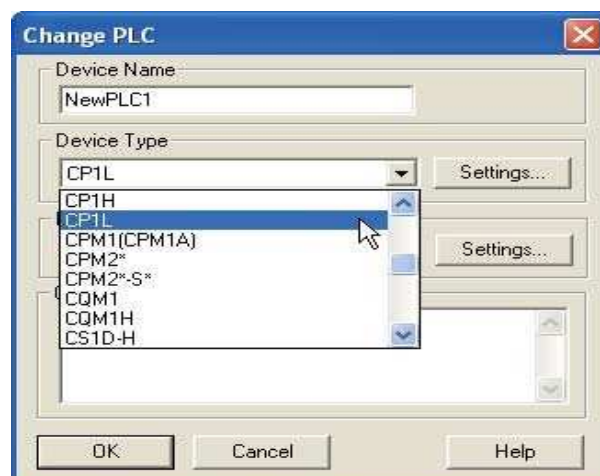
1. Création d'un nouveau projet :

Lors de la première utilisation de CX-Programmer, vous devez créer un nouveau projet. Lors de la création d'un nouveau projet, vous devez définir le type de périphérique cible et le type d'UC pour le programme et les données créés.

Sélectionnez [Fichier] – [Nouveau] dans le menu principal.
boîte de dialogue Modifier API s'affiche.



Sélectionner [CP1L] dans le menu déroulant Type de périphérique.

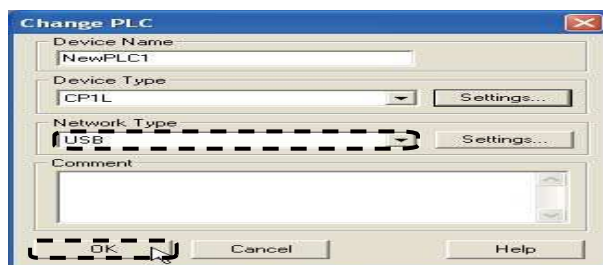
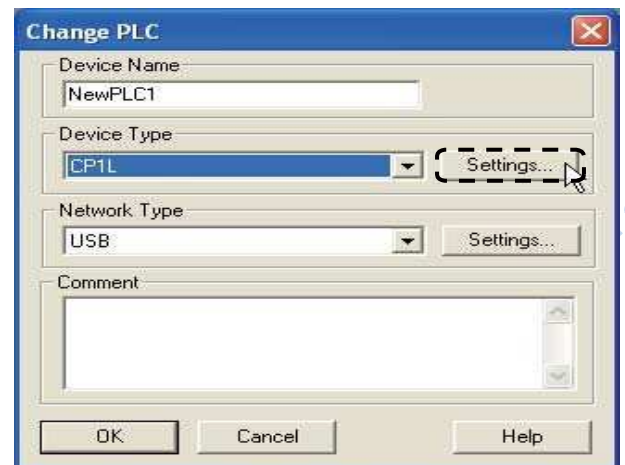


Cliquer sur [Paramètres].

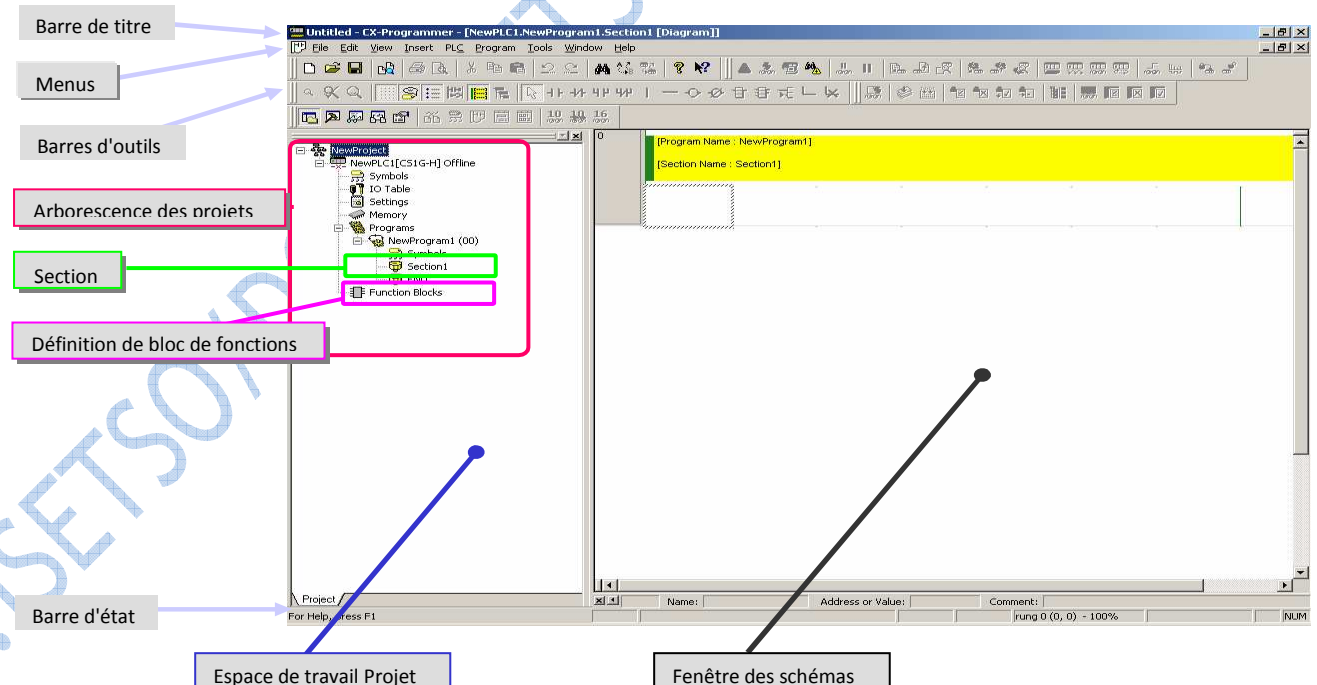
La boîte de dialogue Paramètres type de périphérique s'affiche.

Sélectionner l'UC dans le menu déroulant Type d'UC. Cliquez sur [OK].

La boîte de dialogue Paramètres type de périphérique se ferme.



La boîte de dialogue Modifier API se ferme. La fenêtre principale pour le nouveau projet s'affiche.



Saisie d'un contact normalement ouvert

Appuyer sur la touche [C] du clavier pour ouvrir la boîte de dialogue [Nouveau contact].

Sélectionner le symbole P_On dans la liste déroulante.



P_On est un symbole défini par le système. Il a toujours l'état ON (activé).

[0] ne s'affiche pas lorsqu'il est le premier chiffre d'une adresse.

Le point [.] sépare le numéro de canal et le numéro de relais.

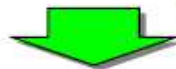
✚ Saisie d'une instance

Appuyer sur la touche [F] du clavier pour ouvrir la boîte de dialogue

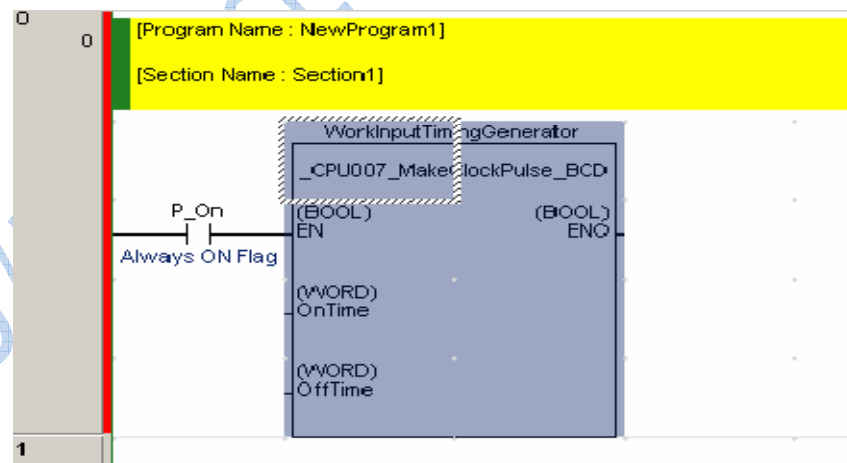
[Modifier une invocation de bloc de fonction].



Attribue un nom au processus spécifique dans le schéma.

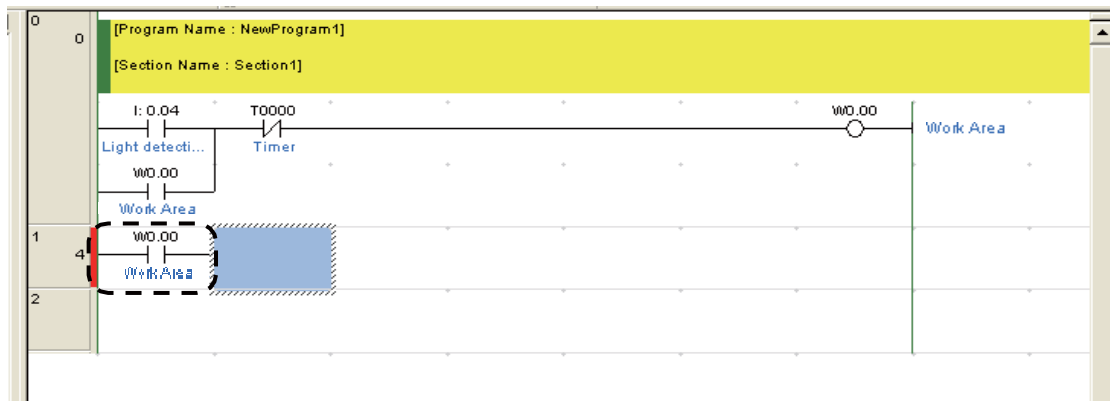


Affichage de l'instruction d'appel FB «Work Input Timing Generator».



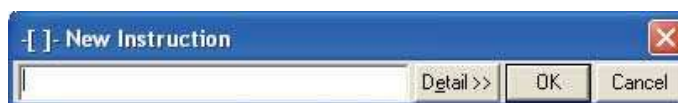
✚ Insertion d'un temporisateur

- Appuyer sur la touche [C]. Entrer contact «W000». Appuyer ensuite sur la touche [Entrée] pendant que la boîte de dialogue «Modification commentaire» soit présente :



Appuyer sur la touche [I].

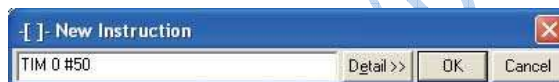
La boîte de dialogue Nouvelle instruction s'affiche.



Introduire une instruction temporisateur « TIM 0 #50 ». Appuyez sur la touche [Entrée].

« TIM 0 #50 » est entré. La boîte de dialogue Modification commentaire s'affiche avec le commentaire E/S déjà entré.

« TIM 0 #50 » indique un délai de 5,0 secondes, avec un drapeau de fin de temporisation de T0000.



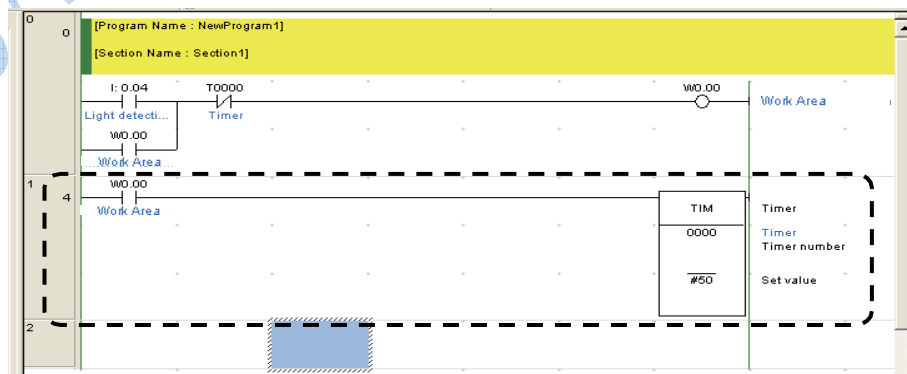
Appuyer sur la touche [Entrée].



Une instruction de temporisation sera affichée sur le programme contact.

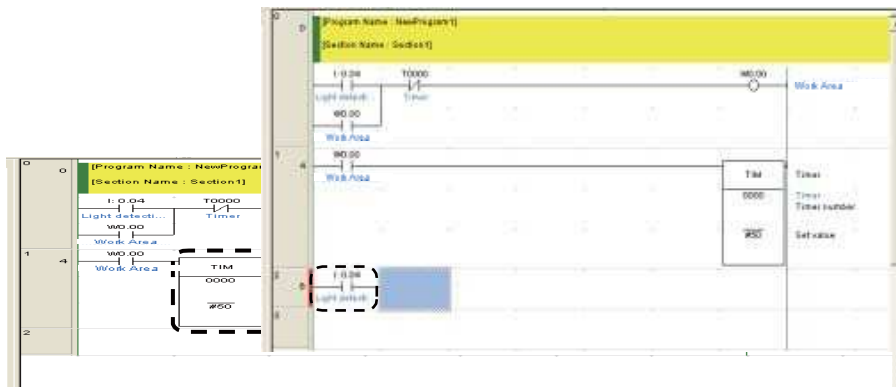
Appuyez sur la touche Bas 3 fois.

Lorsque le curseur est positionné sur le segment suivant, l'entrée de l'instruction de temporisation est terminée.



Saisie d'un compteur

Appuyer sur la touche [C]. Entrer contact «004». Appuyer ensuite sur la touche [Entrée] pendant que la boîte de dialogue «Modification commentaire» soit présente.

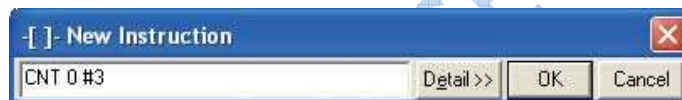


Appuyer sur la touche [I].

La boîte de dialogue Nouvelle instruction s'affiche.



Introduire une instruction compteur «CNT 0 #3 ». Appuyer sur la touche [Entrée].



«CNT 0 #3» est entré. La boîte de dialogue Modification commentaire s'affiche avec le commentaire E/S déjà saisi.

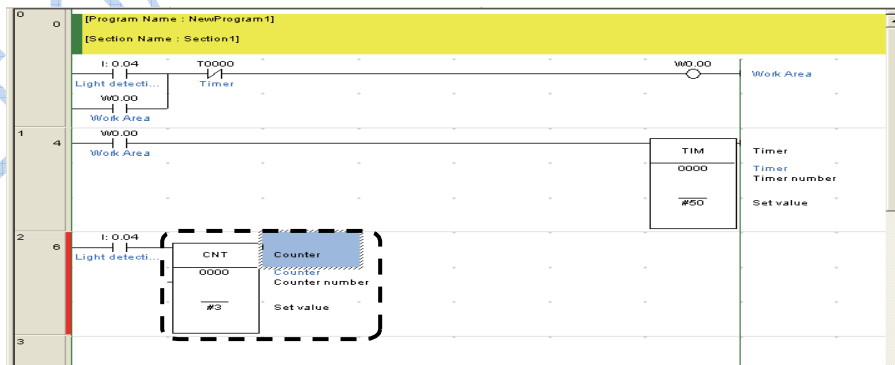
«CNT 0 #3» indique un début de compteur décrémental commençant à 3, avec un drapeau de fin de compteur de C0000.



Entrez « Compteur » comme commentaire E/S. Appuyez sur la touche [Entrée].



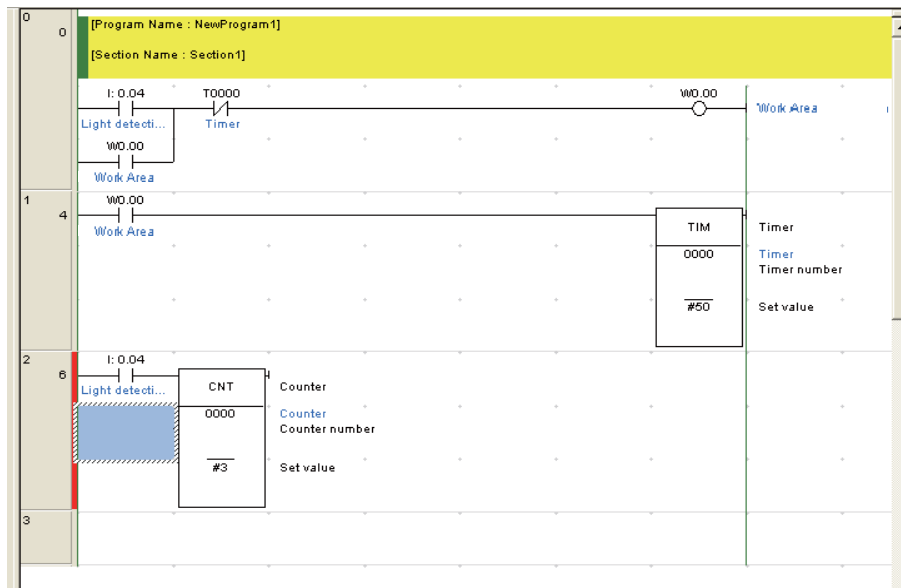
Une instruction de compteur sera affichée sur le programme contact.



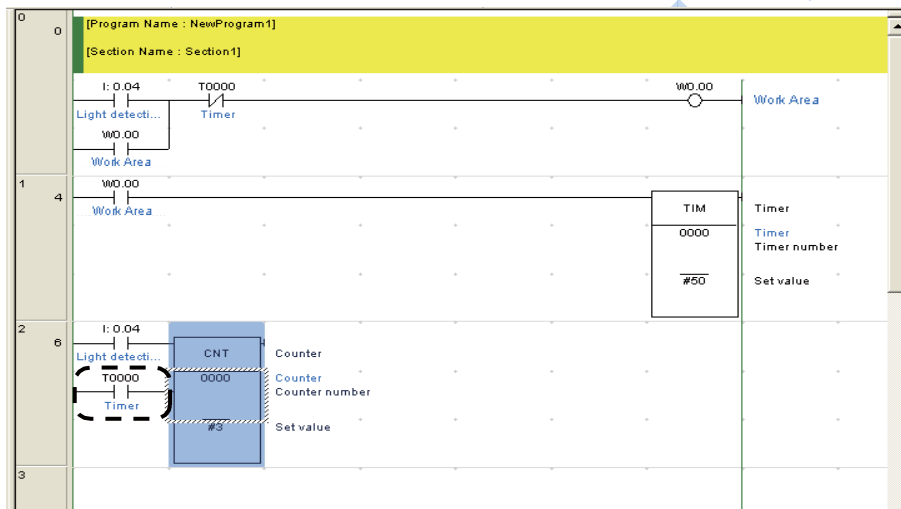
Introduire ensuite une entrée de réinitialisation pour l'instruction compteur.

Le contact minuterie (TIM 0000) est utilisé comme entrée de réinitialisation.

Placez le curseur sous le contact créé à l'étape 1.

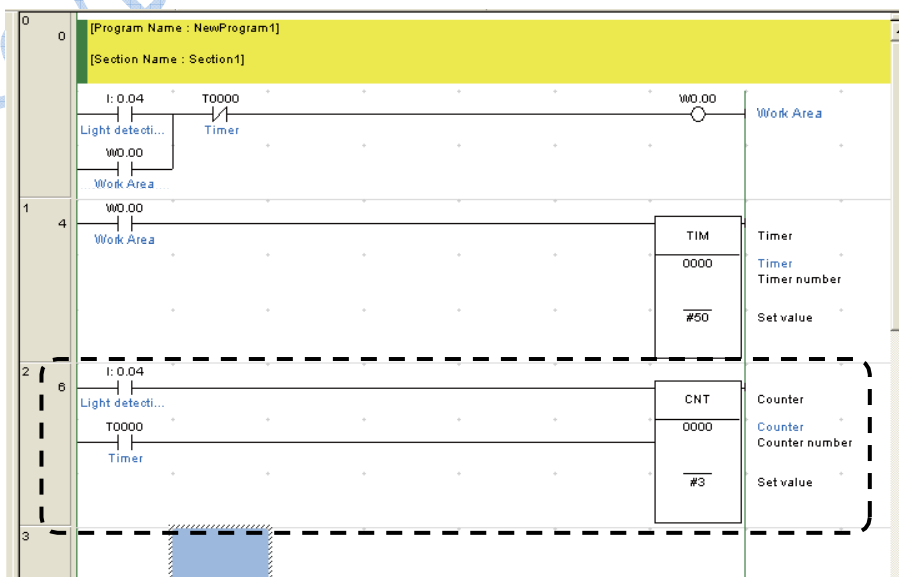


Exploitation de la variable interne du Timer



Appuyer sur la touche Bas 2 fois.

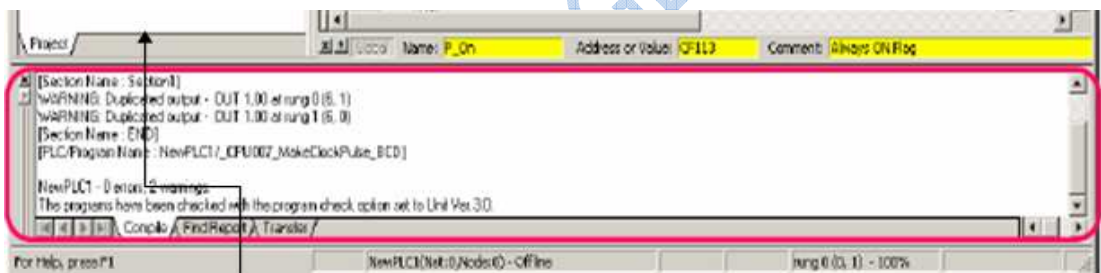
Lorsque le curseur est positionné sur le segment suivant, l'entrée de l'instruction de compteur est terminée.



2. Compilation et contrôle d'un programme :

Avant de transférer le programme, contrôler les erreurs éventuelles à l'aide de la fonction de compilation du programme.


- ✚ La fenêtre Sortie s'ouvre automatiquement lors du contrôle du programme;
- ✚ Pour déplacer le curseur sur une erreur, appuyez sur la touche J ou F4;
- ✚ Pour fermer la fenêtre, appuyer sur la touche Echap.




Les erreurs et les adresses correspondantes s'affichent dans la fenêtre Sortie.

3. Mise en ligne :

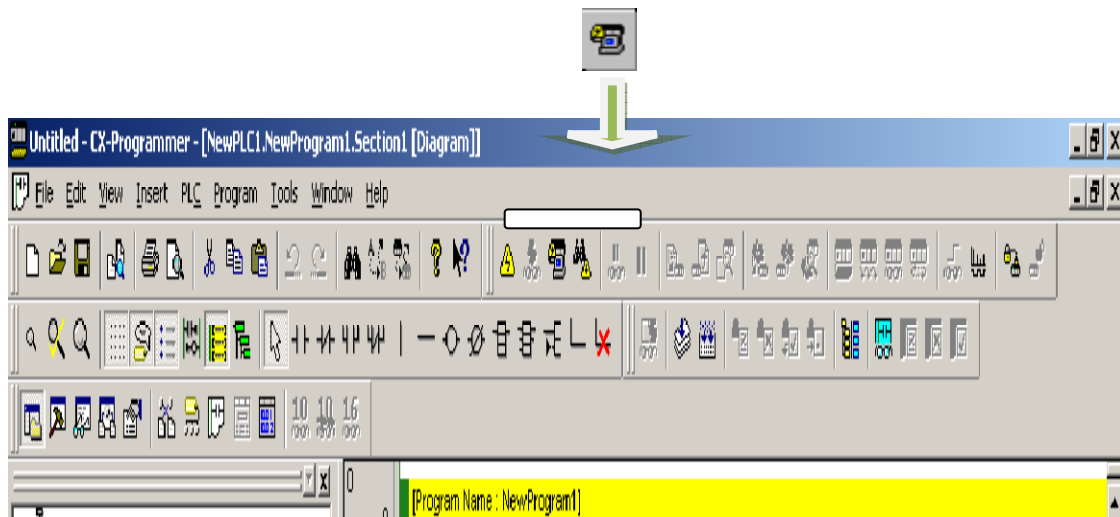
CX-Programmer propose trois modes de connexion qui varient en fonction de l'application.

- ✚ Online normal, permet d'activer le mode en ligne pour un API avec la méthode indiqués lors de l'ouverture d'un projet: 

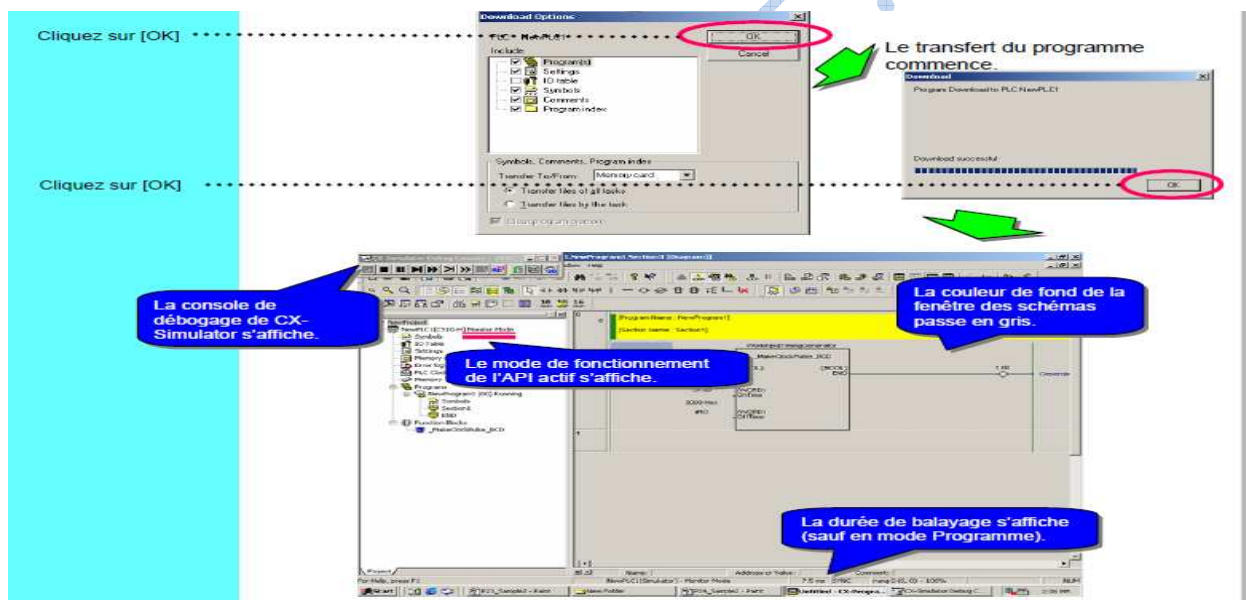
- ✚ Mode online automatique, ce mode reconnaît l'API connecté et vous permet d'activer le mode en ligne de l'API d'un seul bouton: 

Ce qui permet de télécharger toutes les données, comme les programmes, à partir de l'API.

- ✚ Online avec simulateur, permet aussi d'activer le mode en ligne de CX-Simulator d'un seul bouton (CX-Simulator doit être installé):



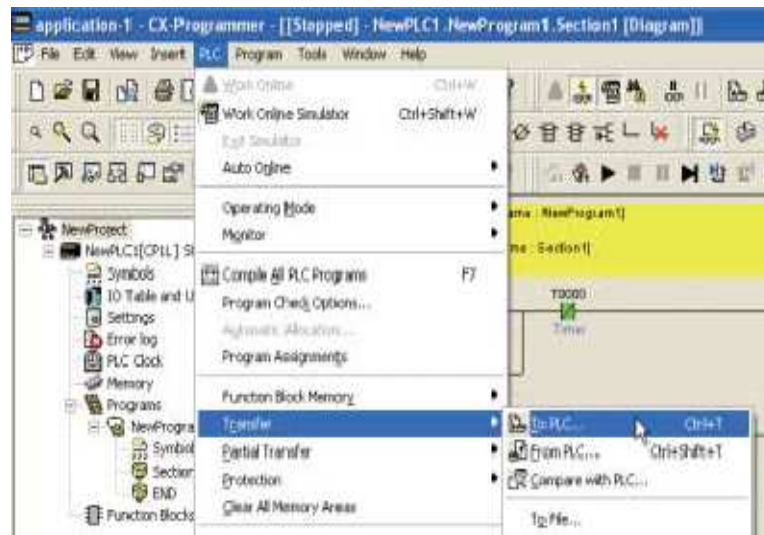
Les fonctions de mise en ligne et de débogage lors de l'utilisation en ligne de CX-Simulator sont expliquées par la figure suivante :



4. Transfert de programmes :

Un programme créé avec CX-Programmer peut être transféré vers le CPU comme suit :

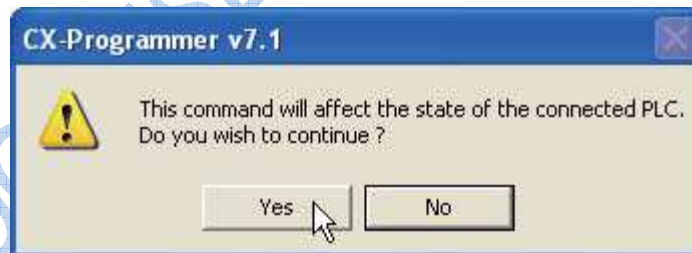
Sélectionner [API] – [Transfert] – [Vers API] dans le menu principal. La boîte de dialogue Options de téléchargement s'affiche.



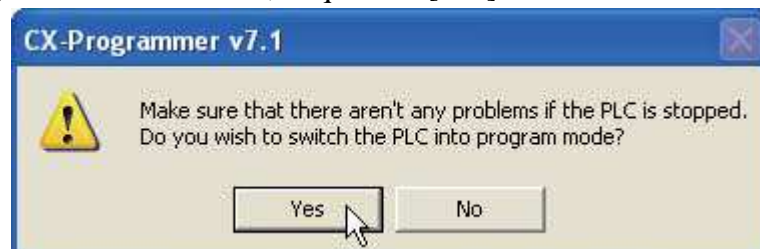
Une boîte de dialogue de confirmation du transfert apparaît.



Cliquer sur [Oui].



Si la boîte de dialogue suivante s'affiche, cliquer sur [Oui].



Le transfert commence et la boîte de dialogue Téléchargement s'affiche. Cliquer sur [OK].

Le transfert du programme est maintenant terminé.

Chapitre

Éléments de programmation des API Omron

I. Introduction

La programmation en schéma à relais empreinte la schématisation habituellement utilisée en schéma électrique pour modéliser les automatismes industriels. Cette représentation à l'avantage d'être compréhensible par des automatismes ayant le réflexe des systèmes à relais électromagnétiques.

II. Les principes des schémas à relais

Un schéma à relais compte une ligne et des branchements. La ligne de gauche est la ligne de bus et les lignes d'embranchements sont les lignes d'instructions ou lignes secondaires. Deux lignes d'instructions sont reliées par au moins une condition. L'exécution d'un schéma se fait toujours de gauche à droite.

Les conditions sont représentées par deux lignes verticales ou encore par deux lignes verticales traversées par une diagonale. Dans le premier cas la condition est initialement ouverte elle est transcrite par une mnémonique tel que (LD, AND, OR) ; dans le second cas c'est une condition initialement fermée (fermé au repos) elle est transcrite par une mnémonique du type (LDNOT, OR NOT, AND NOT).

1- La ligne de départ verticale placée à l'extrémité gauche du réseau est le point de départ de ce dernier.

2- Les lignes verticales parallèles à celle ci sont des lignes représentant des sous routines par rapport à la ligne principale.

3- Les conditions sont représentées par deux lignes verticales.

✚ Terminologie fondamentale :

En schéma à relais à chaque condition est attribuée un bit ce bit est à (un) lorsque la condition est vraie, la condition est alors en position ON. Une condition initialement fermée est initialement en position OFF, elle passera en position ON avec un état complémentaire à celui d'une condition initialement ouverte.

✚ Conditions d'exécution :

Dans un schéma à relais l'exécution d'une instruction se fait à partir du moment où la combinaison des conditions qui la précède est vraie.

✚ Bits d'opérande :

Une condition peut être déterminée par :

- 1- Un bit d'entrée/sortie ;
- 2- Un drapeau (bit d'état) ;
- 3- Un bit de travail ;
- 4- Un temporisateur ou un compteur.

✚ Blocs logiques et blocs d'instruction :

On appelle bloc logique un groupe d'instructions qui génère un résultat logique. La décomposition d'un programme en plusieurs blocs logiques simplifie la synthèse et le test des applications. Un bloc d'instruction comprend toutes les instructions reliées entre elles dans un même schéma.

✚ Codes mnémoniques :

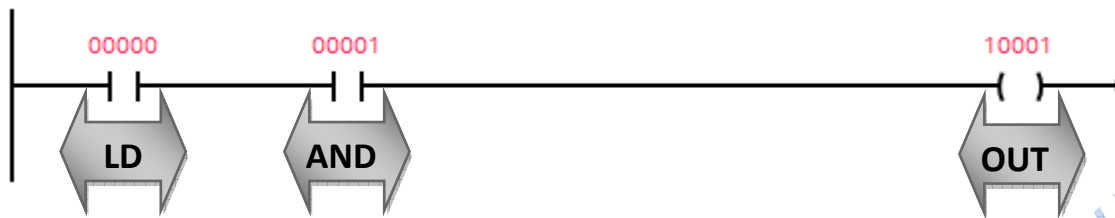
Un schéma à relais est non compréhensible par l'automate, ainsi pour pouvoir le saisir par une console de programmation il faut le transcrire en mnémonique intelligible par l'API.

Le code mnémonique fait correspondre à chaque formalisme du schéma à relais une instruction équivalente qui réclamera des opérandes en conséquence. Le programme résultant de cette transcription est chargée puis exécuté à partir de la mémoire de l'automate.

1. Exemple de programmation :

a. Programmation d'une fonction ET logique :

Le programme suivant réalise une fonction ET logique entre deux entrées de l'automate. Si la condition est vraie la sortie sera active. Le schéma à relais qui représente cette application est :



On vous propose de programmer le code numérique comme le montre la table suivante :

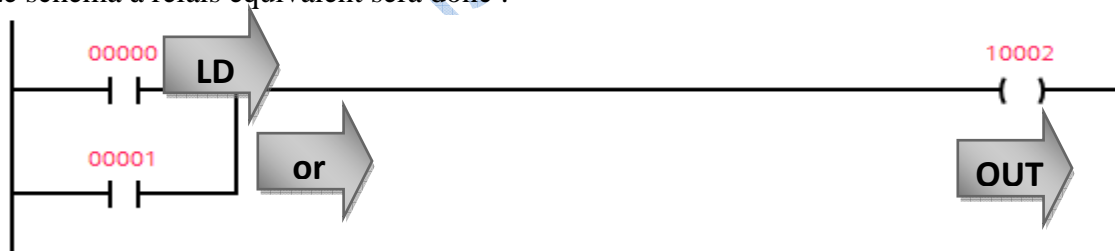
Adresse	Instruction	Opérande
00000	LD	00000
00001	AND NOT	00001
00002	OUT	10001
00003	END	-----

🔧 Questions :

- Q1- Vérifier le fonctionnement de votre programme et déduire la fonction qu'il réalise.
 Q2- Modifier le schéma à relais pour que la condition devienne une fonction ET entre les deux entrées.
 Q3- Modifier directement votre programme et tester-le.

b. Programmation d'une fonction OU logique :

La fonction OU se résume par deux contacts en parallèle qui aboutissent au même résultat. Le schéma à relais équivalent sera donc :



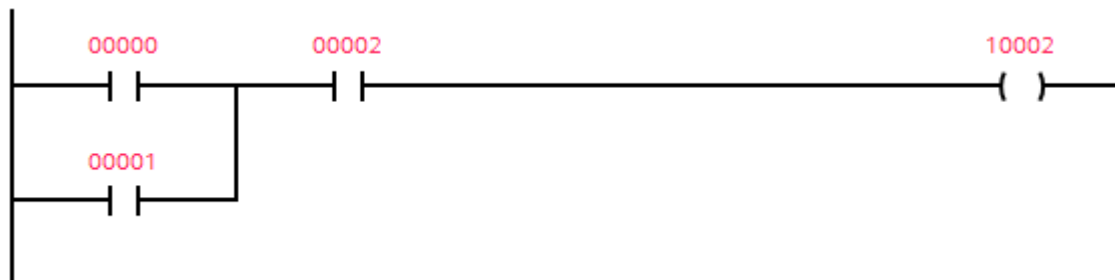
Adresse	Instruction	Opérande
00000	LD	00000
00001	OR	00001
00002	OUT	10002
000003	END	-----

🔧 Questions :

- Q1- Programmer une fonction OU avec les entrées d'adresses 4 et 5 et la sortie d'adresse 0.
 Q2- Modifier le programme pour qu'il réalise la fonction NAND.

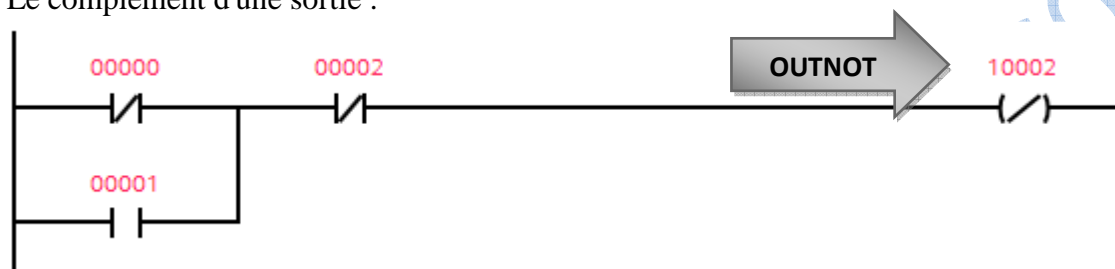
Exercice 1:

Ecrire et tester le programme équivalent au schéma suivant :



Exercice 2:

Le complément d'une sortie :

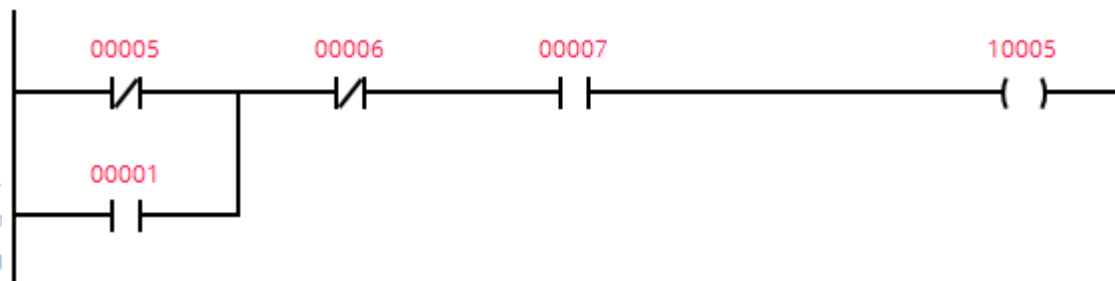


Compléter les opérands du programme suivant avant de le tester.

Adresse	Instruction	Opérande
00000	LD NOT	
00001	OR	
00002	AND NOT	
00003	OUT NOT	
00004	END	-----

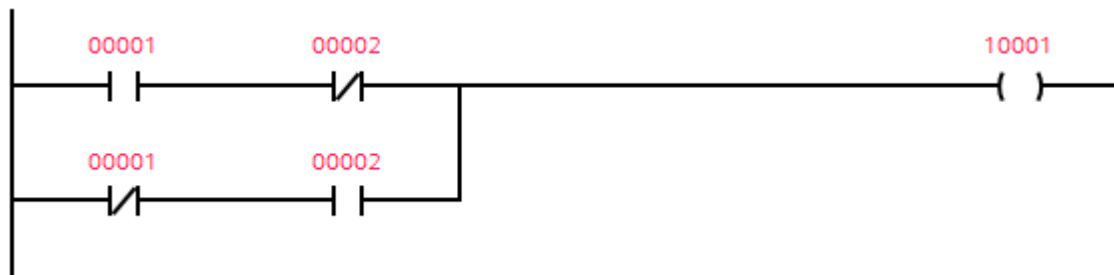
Exercice 3:

Ecrire le code mnémorique et tester le schéma suivant :



Exercice 4:

Le schéma à relais de la fonction OU EXCLUSIF entre les entrées d'adresse 1 et 2 est le suivant :



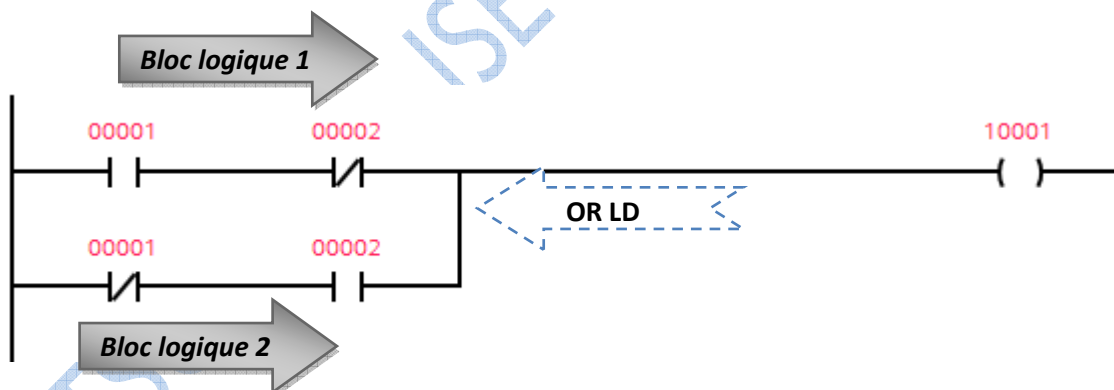
🚩 **Questions :**

Q1- Saisir le code mnémotique suivant et tester son fonctionnement.

Adresse	Instruction	Opérande
00000	LD	00001
00001	AND NOT	00002
00002	LD NOT	00001
00003	OR LD	
00004	OUT	10001
00005	END	-----

Q2- Donner le code mnémotique qui traduit le schéma à relais précédent puis tester son fonctionnement.

La fonction OU EXCLUSIF est un cas de programmation d'instruction en bloc logique en effet le schéma à relais correspondant peut être subdiviser comme suit .



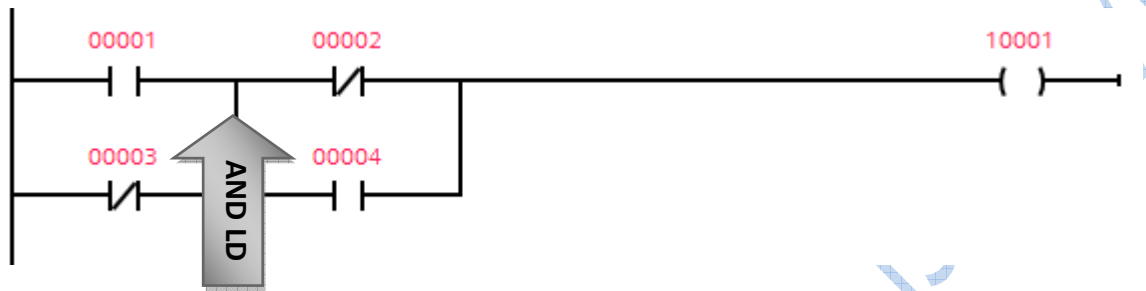
chaque bloc logique est seul, l'instruction **ORLD** permet de définir la condition qui relie le premier bloque au second, dans ce cas les deux bloques sont reliés avec une condition **OU**. le programme présenté sommairement à l'exercice précédent peut être détaillés comme suit :

Adresse	Instruction	Opérande
00000	LD	00001
00001	AND NOT	00002
00002	LD NOT	00001

00003	AND	00002
00004	OR LD	-----
00005	OUT	10001
00006	END	-----

Exercice 4:

Soit le schéma à relais suivant :

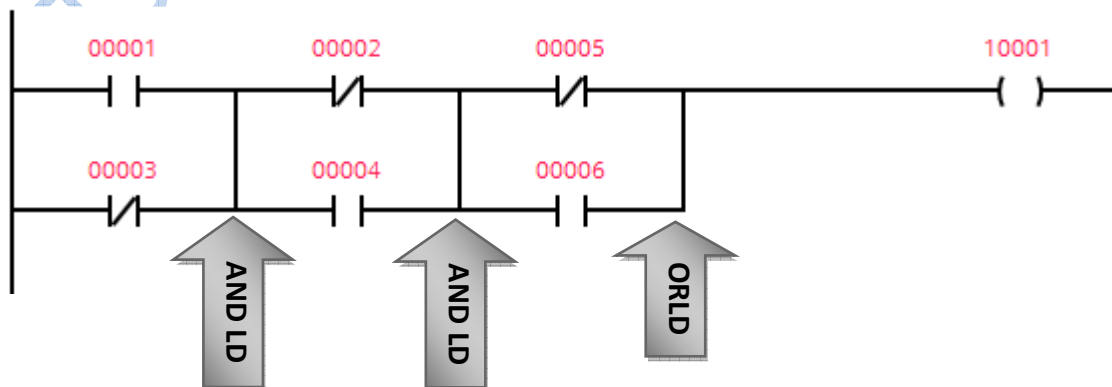


Questions :

Q1- Ecrire le code mnémotechnique du schéma en vous aidant du tableau suivant .

Adresse	Instruction	Opérande
00000	LD	0000X
00001	XX	XXXXX
00002	LD NOT	0000X
00003	XX	XXXXX
00004	AND LD	-----
00005	OUT	10001
00006	END	-----

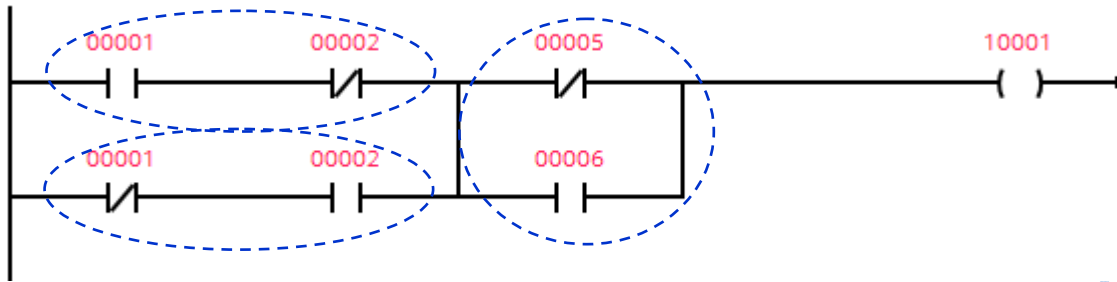
Exercice 5 : Assemblage de plusieurs blocs logiques



Ecrire le programme puis tester son fonctionnement.

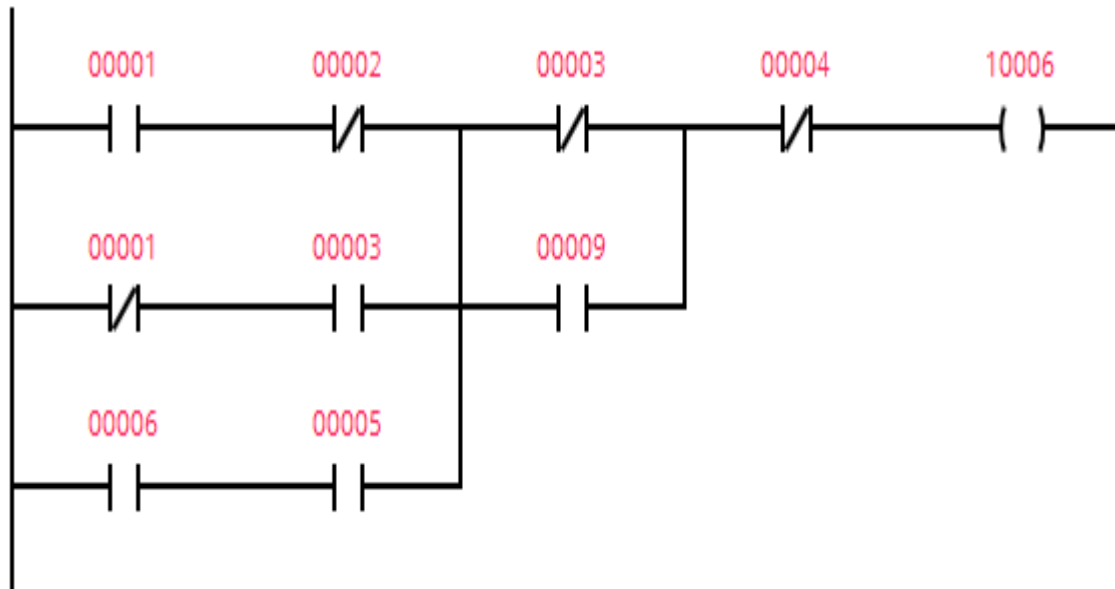
Exercice 6:

Pour la saisie de schémas assez complexes une décomposition en bloc logique est recommandée.



Questions :

- Q1- Rassembler les blocs 1 et 2 avec un ORLD.
 Q2- Former le bloc 3 et relier le, au bloc formé en Q1.
 Q3- Proposer une autre méthode pour refaire le même traitement.

Exercice de synthèse :

Questions :

- Q1- Décomposer le schéma en blocs logiques.
 Q2- Ecrire le code mnémotechnique équivalent et tester le programme.