

L'automate programme (API)



EL KACEM EL MOSTAFA

ARCHITECTURE DE L'AUTOMATE (API)

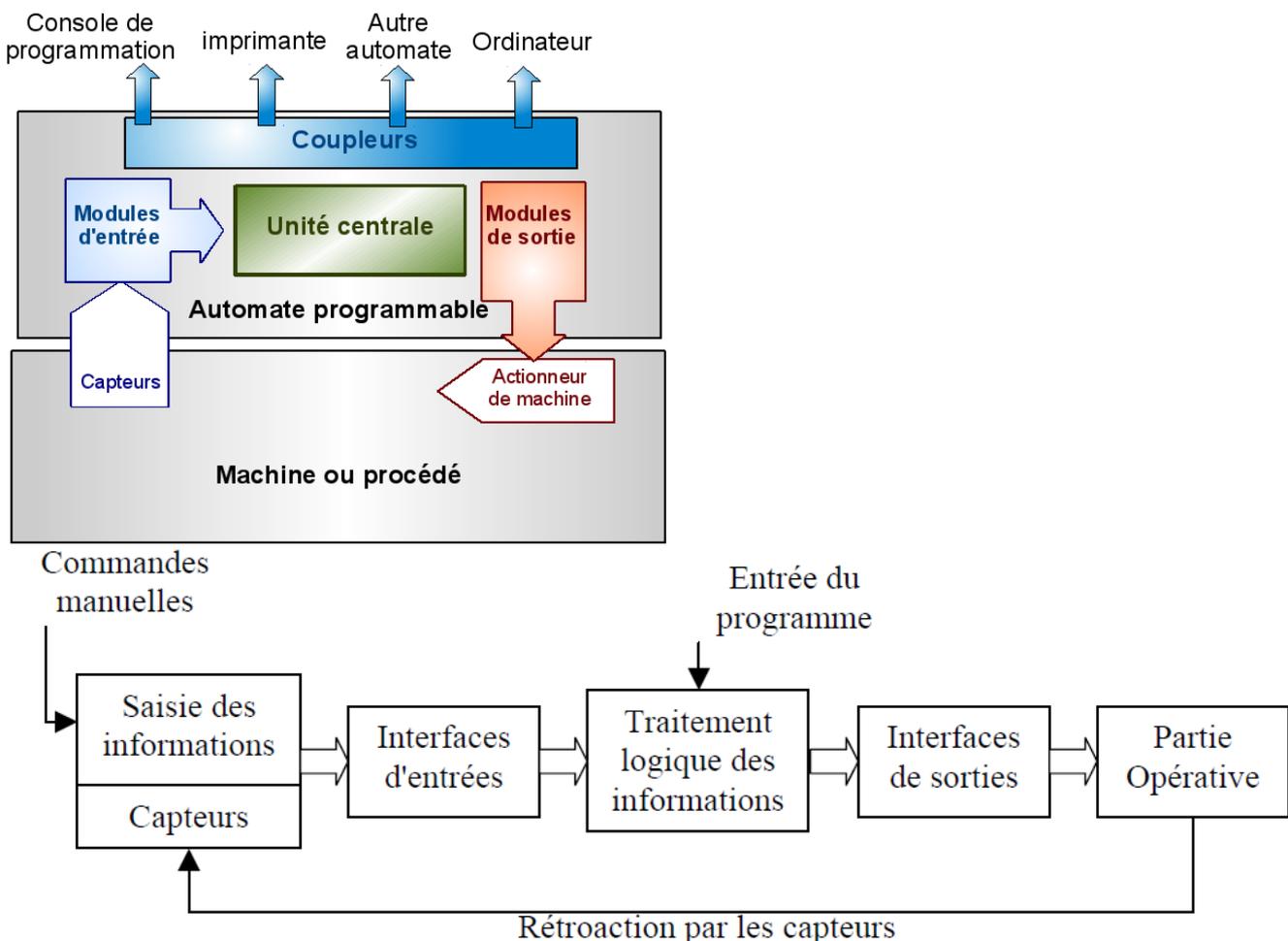
1- Définition :

Un Automate Programmable Industriel (API) est une machine électronique programmable par un personnel non informaticien et destiné à piloter et à faire fonctionner une installation industrielle automatisée. Il est en général manipulé par un personnel électrotechnicien.

2- Structure fonctionnelle de l'automate :

L'automate programmable industriel est un appareil qui traite les informations selon un programme préétabli.

Son fonctionnement est basé sur l'emploi d'un micro-processeur et de mémoires.



3- Structure d'un système de traitement :

a) Interface d'entrée

Elle permet d'acquérir les informations à partir des **organes de commande manuelle et des capteurs** (saisie de l'information) et de les transmettre à l'unité de traitement après avoir réalisé l'isolation galvanique par des opto-coupleurs.

Exemples : Boutons poussoirs, interrupteurs, commutateurs, clavier, potentiomètre etc....

Exemple d'organes de commande manuelle



Exemples de capteurs :



Interrupteur de position

thermostat

pressostat

b) Traitement des informations

L'unité de traitement CPU effectue toutes les opérations logiques et numériques selon un programme enregistré en mémoire, elle reçoit l'état logique et numérique de tous les entrées et suite au programme, elle affecte les états logiques et numériques des sorties .
Le travail de la CPU se fait d'une façon cyclique contrôlé par un chien de garde.



c) Interface de sortie

Elle permet de commander les sorties toute ou rien (TOR) (les contacteurs, les moteurs pas à pas, les électrovannes) et les sorties analogiques (boucle de régulation, débit, température et variateur de vitesse etc...)

La commande des actionneurs se fait grâce aux Préactionneurs.

Actionneurs :



Moteur

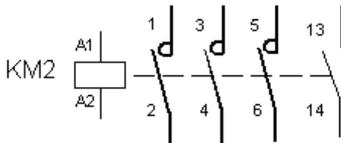


thermoplongeur

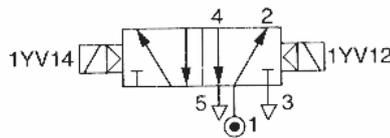


vérin

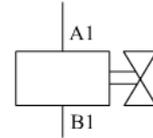
Préactionneurs :



Contacteurs



Distributeur

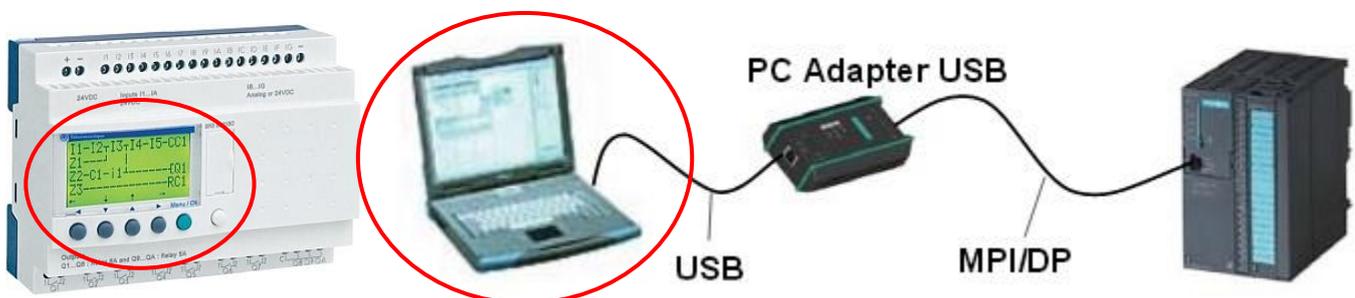


électrovanne



d) communication et dialogue

Elle est réalisée avec l'opérateur par un pupitre de dialogue ou par l'intermédiaire d'un ordinateur et avec les autres automates pour un réseau informatique local.



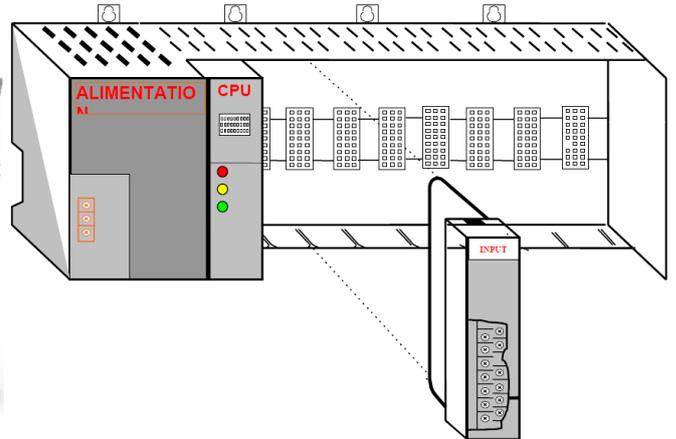
4- Description des automates

Il existe deux types d'automate programmable industriel :

- + L'automate monobloc
- + L'automate modulaire



Le type monobloc possède généralement un nombre d'entrées et de sorties limités. Bien qu'il soit parfois possible d'ajouter des extensions d'entrées/sorties, le type monobloc a pour fonction de résoudre des automatismes simples faisant appel à une logique séquentielle et utilisant des informations tout-ou-rien.



Le type modulaire est adaptable à toutes situations. Selon le besoin, des modules d'entrées/sorties TOR, analogiques avec possibilité d'extension et de communication avancée.

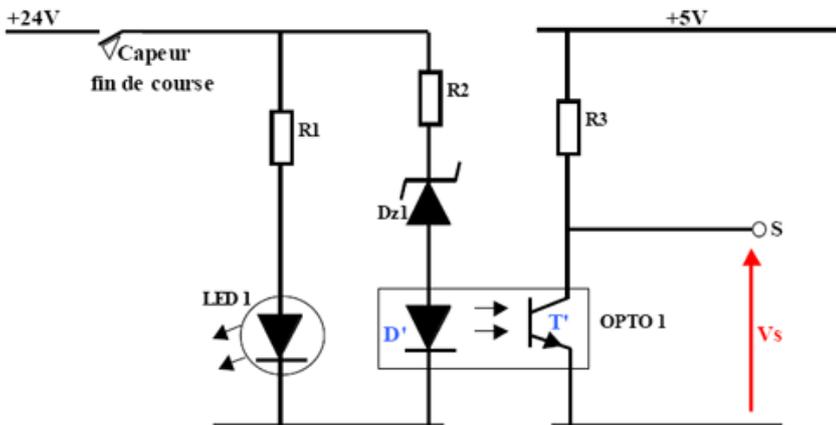
5- Nature des informations traitées par l'automate :

Les informations peuvent être de type :

- **Tout ou rien (T.O.R.)** : l'information ne peut prendre que deux états (vrai/faux, 0 ou 1) C'est le type d'information délivrée par un détecteur, exemple un bouton poussoir ...
- **Analogique** : l'information est continue et peut prendre une valeur comprise dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (pression, température ...)
- **Numérique** : l'information est contenue dans des mots codés sous forme binaire ou bien hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent, un contrôle panel, clavier etc...

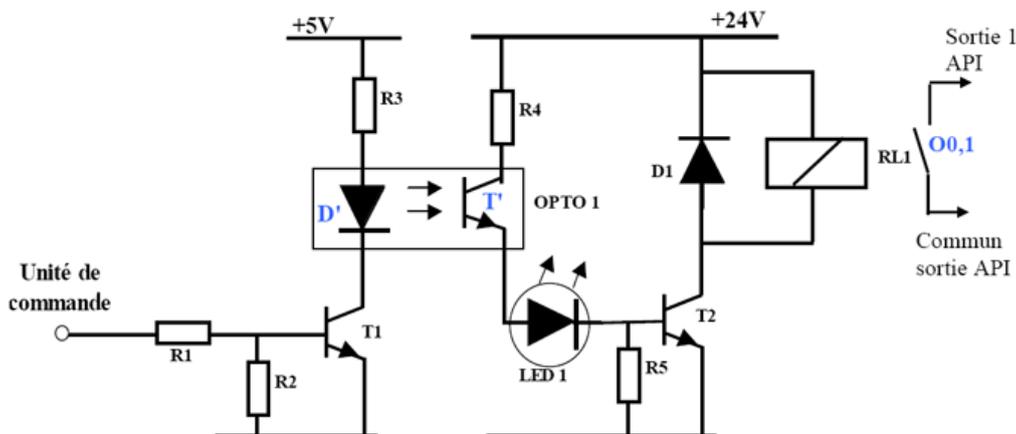
Cartes d'entrées : TOUT OU RIEN (TOR 0 ou1) et Analogique (4-20mA ou 0 10v)

Elles sont destinées à recevoir l'information en provenance des capteurs et adapter le signal en le mettant en forme, en éliminant les parasites et en isolant électriquement l'unité de commande de la partie opérative.



Cartes de sorties : TOUT OU RIEN (TOR 0 ou1) et Analogique (4-20mA ou 0 10v)

Elles sont destinées à commander les Préactionneurs et éléments des signalisations du système

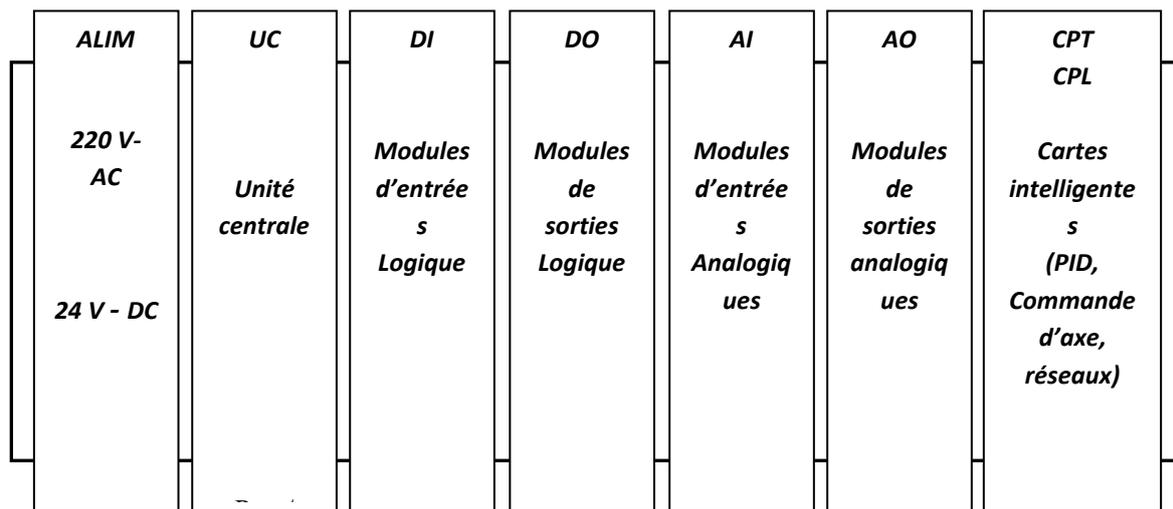


6- Architecture matérielle des API

Un Automate Programmable se présente sous la forme d'un ou plusieurs racks dans lesquels viennent s'enficher les différents modules fonctionnels :

- L'alimentation 110/220 VCA ou 24/48 VCC,
- L'unité centrale de traitement à base de microprocesseurs,
- Des cartes d'entrées/sorties logiques (TOR),
- Des cartes d'entrées/sorties analogiques (ANA),
- Des cartes de comptage rapide (CPT),
- Des cartes de communication (CPL),
- Des cartes dites " intelligentes " ou " métier " (réseaux, asservissement, régulation, ...).

La figure suivante, donne un exemple d'architecture matérielle d'un API.



Chaque module d'entrées/sorties comporte un bornier de raccordement et un ensemble de visualisation de l'état logique de chaque voie. Cette organisation modulaire permet une grande souplesse de configuration adaptée aux besoins de l'utilisateur ainsi qu'un diagnostic et une maintenance aisés.

Câblage d'un automate programmable

Alimentation de l'automate :

L'automate est alimenté généralement par le réseau monophasé 230V ; 50 Hz, de préférence à partir d'un onduleur mais d'autres alimentations sont possibles (110V etc...).

La protection sera de type magnétothermique (voir les caractéristiques de l'automate du constructeur).

Il est souhaitable d'asservir l'alimentation de l'automate par un circuit de commande spécifique (contacteur KM1).

De même, les sorties seront asservies au circuit de commande et alimentées après validation du chien de garde.

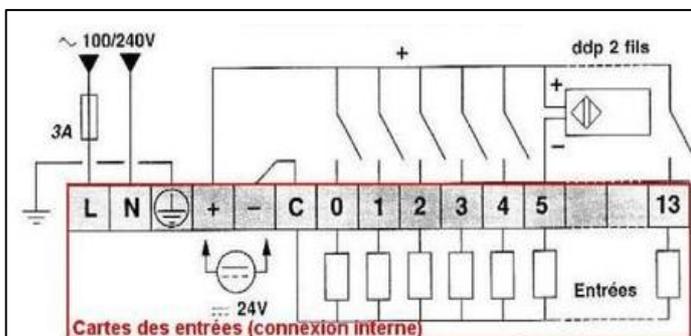
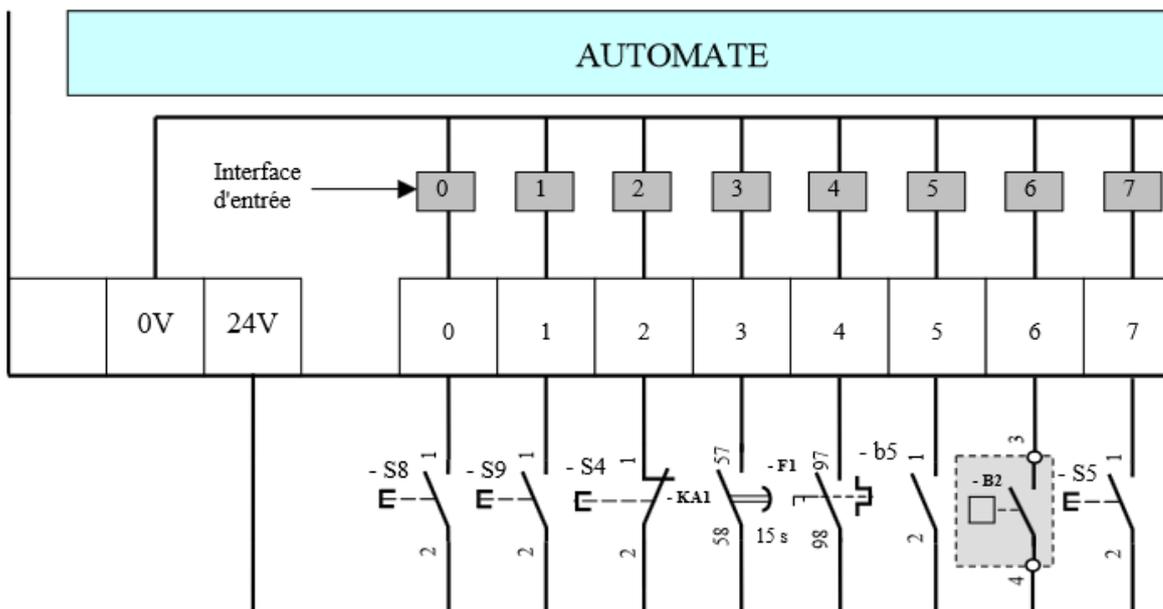
Le temps d'exécution de chaque cycle est vérifié par un temporisateur appelé Watchdog (chien de garde) qui enclenche une procédure d'alarme en cas de dépassement de celui-ci (réglé par l'utilisateur).

Alimentation des entrées en 24 V :

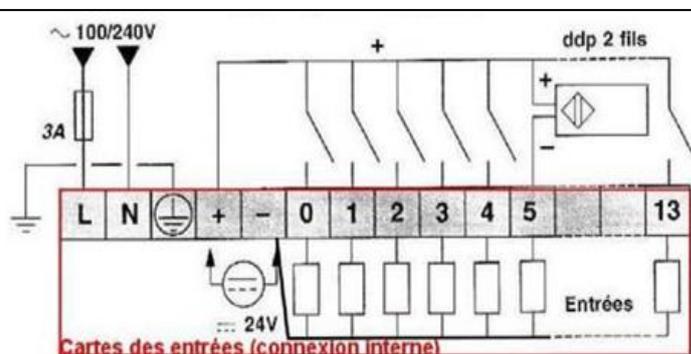
L'automate est pourvu généralement d'une alimentation pour les capteurs/détecteurs (attention au type de logique utilisée : logique positive ou négative).

Les entrées sont connectées au 0 V (commun) de cette alimentation.

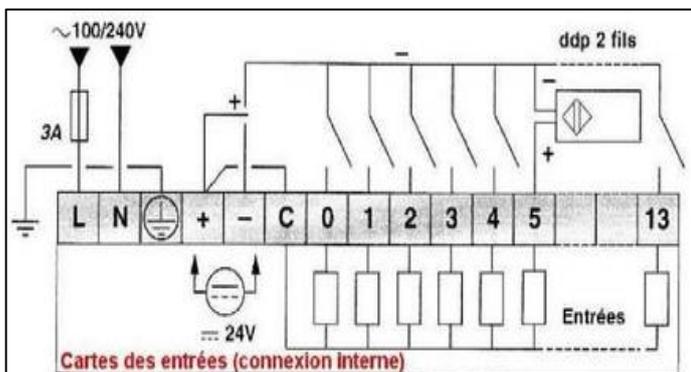
Les informations des capteurs/détecteurs sont traitées par les interfaces d'entrées.



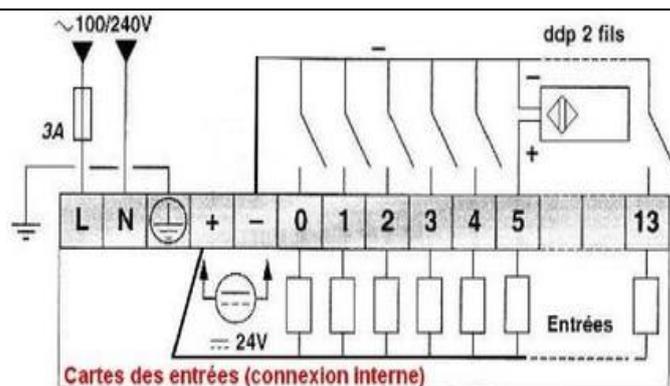
Exemple avec bornes communes des entrées à relier au potentiel négatif (0V)



Exemple avec entrées préconnectés en interne au potentiel négatif (0V)



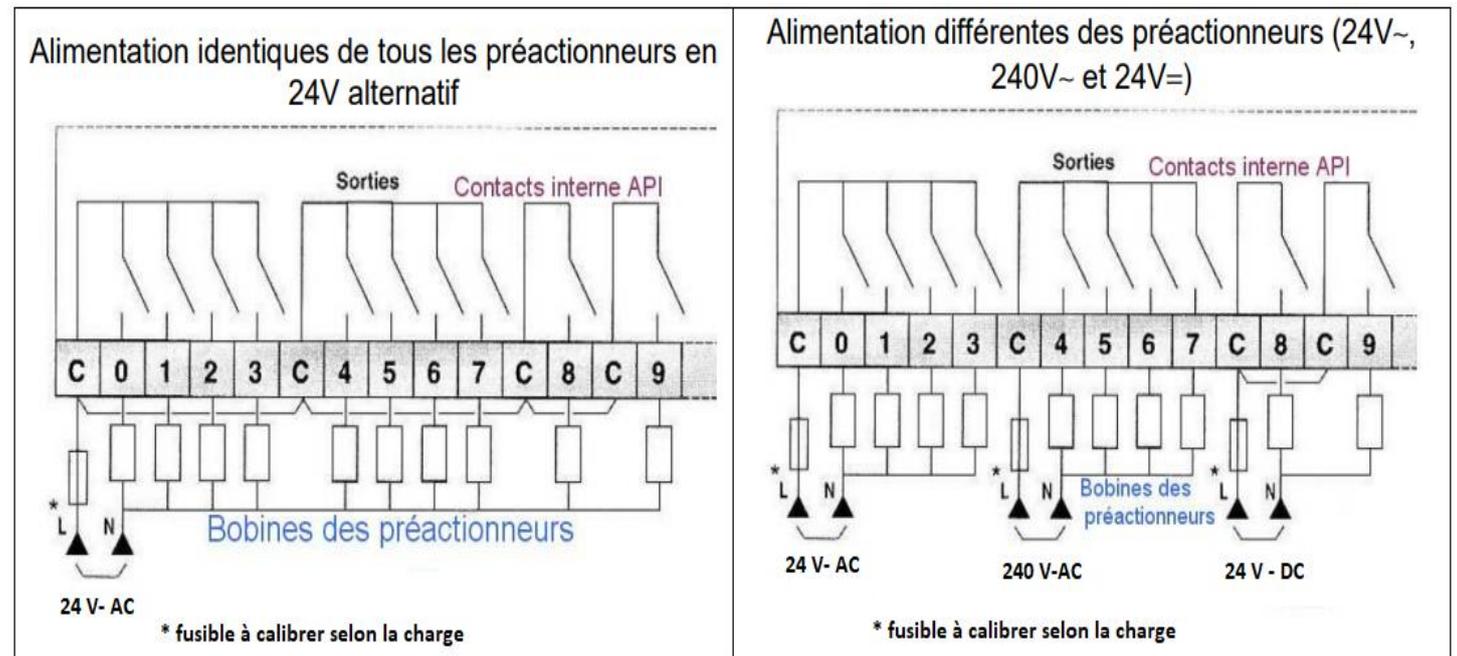
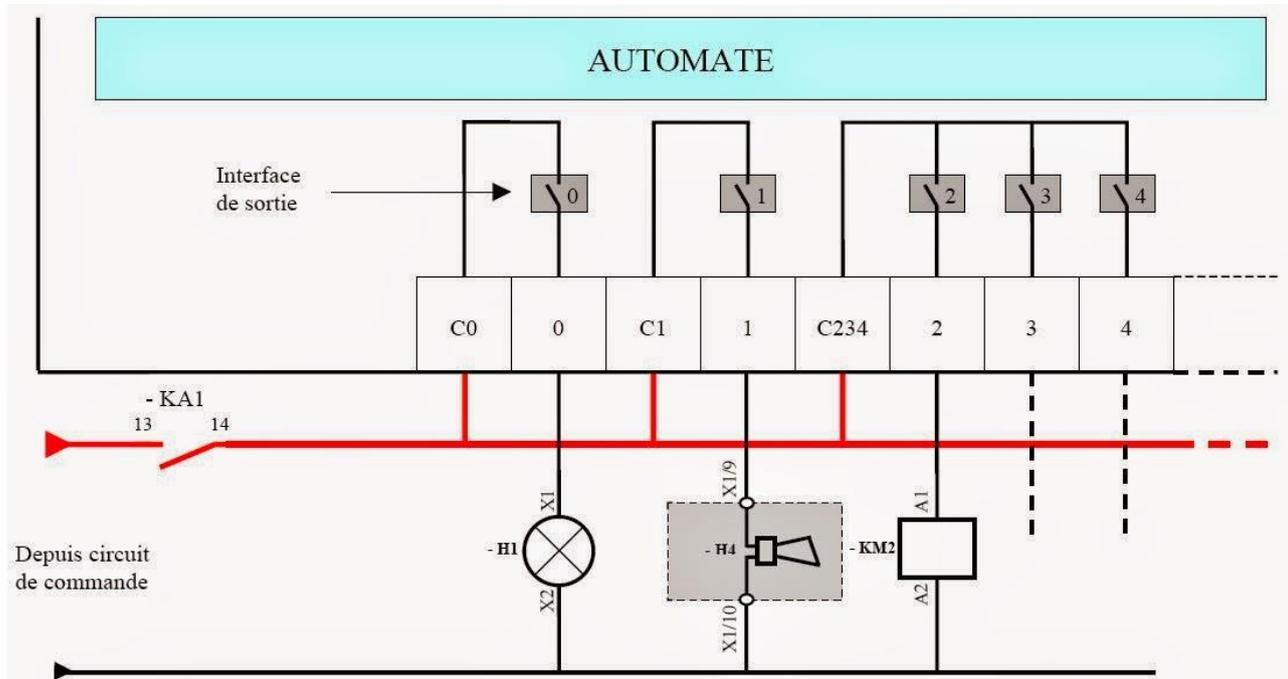
Exemple avec bornes communes des entrées à relier au potentiel positif (24 V)



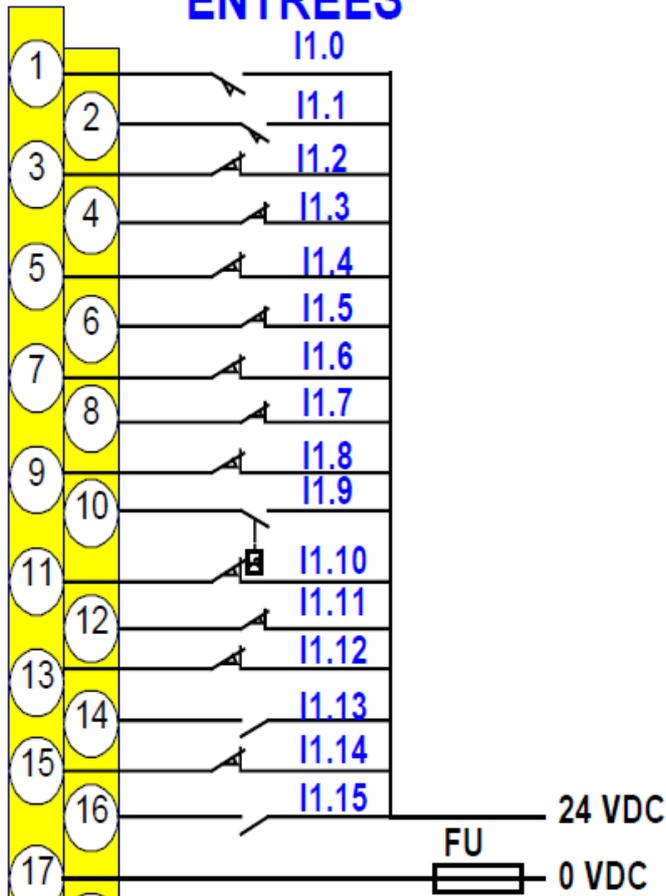
Exemple avec entrées préconnectés en interne au potentiel positif (24 V)

Alimentation des sorties de l'automate :

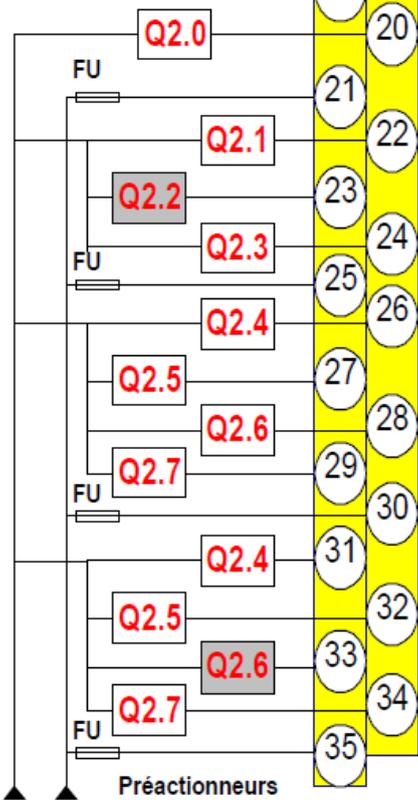
Les interfaces de sorties permettent d'alimenter les divers Préactionneurs.



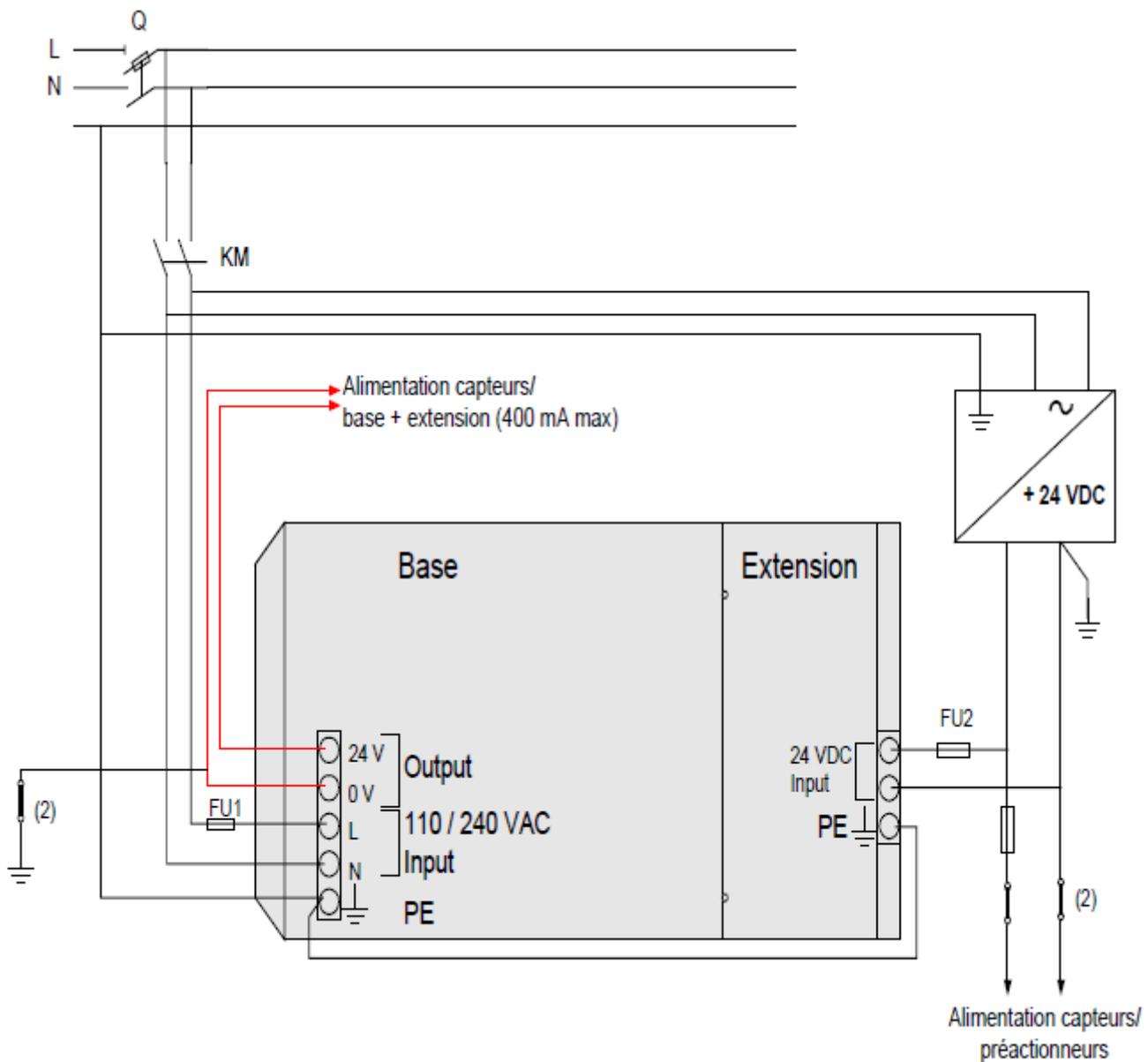
ENTREES



SORTIES



FU: fusible à fusion rapide calibre selon la charge



Programmation de l'automate

Pour programmer un API on utilise soit :

- Une console dont la marque est la même que celle de l'API
- Un PC bureau ou portable avec logiciel de programmation approprié à l'API
- Le transfert du programme vers l'automate se fait grâce à un câble de communication RS232, USB, ou Ethernet Rj45

Exemple d'API avec logiciel approprié

LES AUTOMATES SIEMENS : Step 5, STEP7 ou TIA Portal

LES AUTOMATES ALLEN BRADLEY : RSLOGIX 5, RSLogix 500

LES AUTOMATES TELEMECANIQUE : PL7 Pro, Unity Pro

LES AUTOMATES OMRON : CX PROGRAMMER ou CX-One

Langage de programmation :

Les langages de programmation des API sont de nature diverse, le choix à priori, en fonction des utilisateurs visés et du type de traitement à effectuer.

Langage IL : Liste d'instructions

Ce langage utilise des codes symboliques et peut être représenté sous forme de lignes ou sous forme d'une suite d'instructions.

Un exemple de programme écrit en langage LIST.

$$A4.0 = E0.0.\overline{E0.1} + \overline{E0.0}.E0.1 = E0.0 \oplus E0.1$$

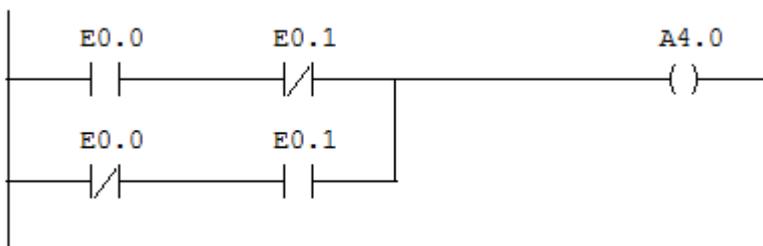
U	E	0.0
UN	E	0.1
O		
UN	E	0.0
U	E	0.1
=	A	4.0

Langage LD : langage à contacts (Ladder)

Le langage Ladder utilise le symbolisme classique des schémas à relais accompagné de blocs graphiques préprogrammés pour réaliser des fonctions d'automatisme (calculs, temporisations, compteurs, ...).

Exemple de programme écrit en langage Ladder.

$$A4.0 = E0.0.\overline{E0.1} + \overline{E0.0}.E0.1 = E0.0 \oplus E0.1$$

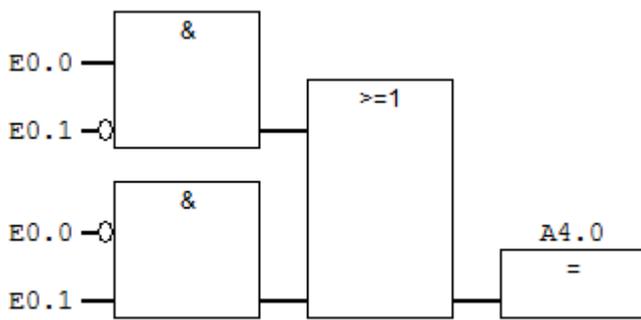


Langage FBD : diagramme fonctionnel — logigramme

Sur certains automates, entre autre SiémenS, on peut également programmer en logigramme graphique.

Un exemple de programme écrit en langage logigramme.

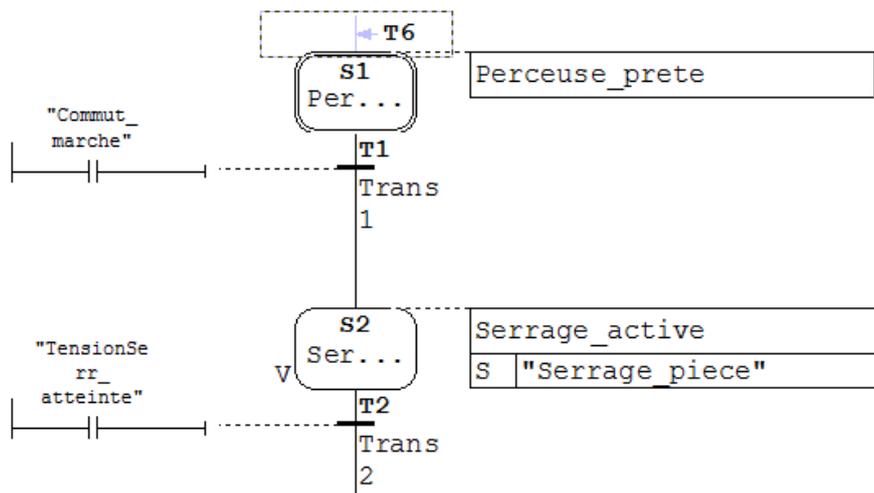
$$A4.0 = E0.0.\overline{E0.1} + \overline{E0.0}.E0.1 = E0.0 \oplus E0.1$$



Langage SFC- diagramme fonctionnel en séquence

Le langage SFC (GRAFCET), est utilisé par certains constructeurs d'automates (Schneider, Siemens,...). Il permet une programmation aisée des systèmes séquentiels

Un exemple de programme écrit en langage GRAFCET sur un API S7-300.



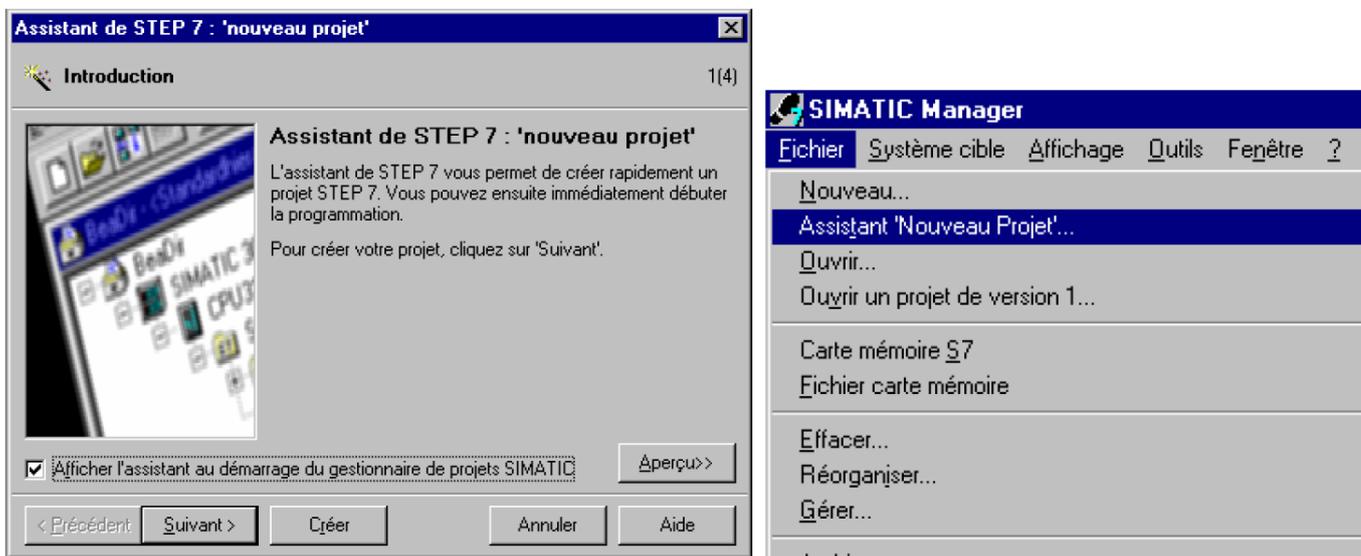
Configuration matériel (exemple API siemens S7-300)

Avant la création d'un projet on doit procéder à la configuration matériel soit manuellement ou à l'aide de l'assistant

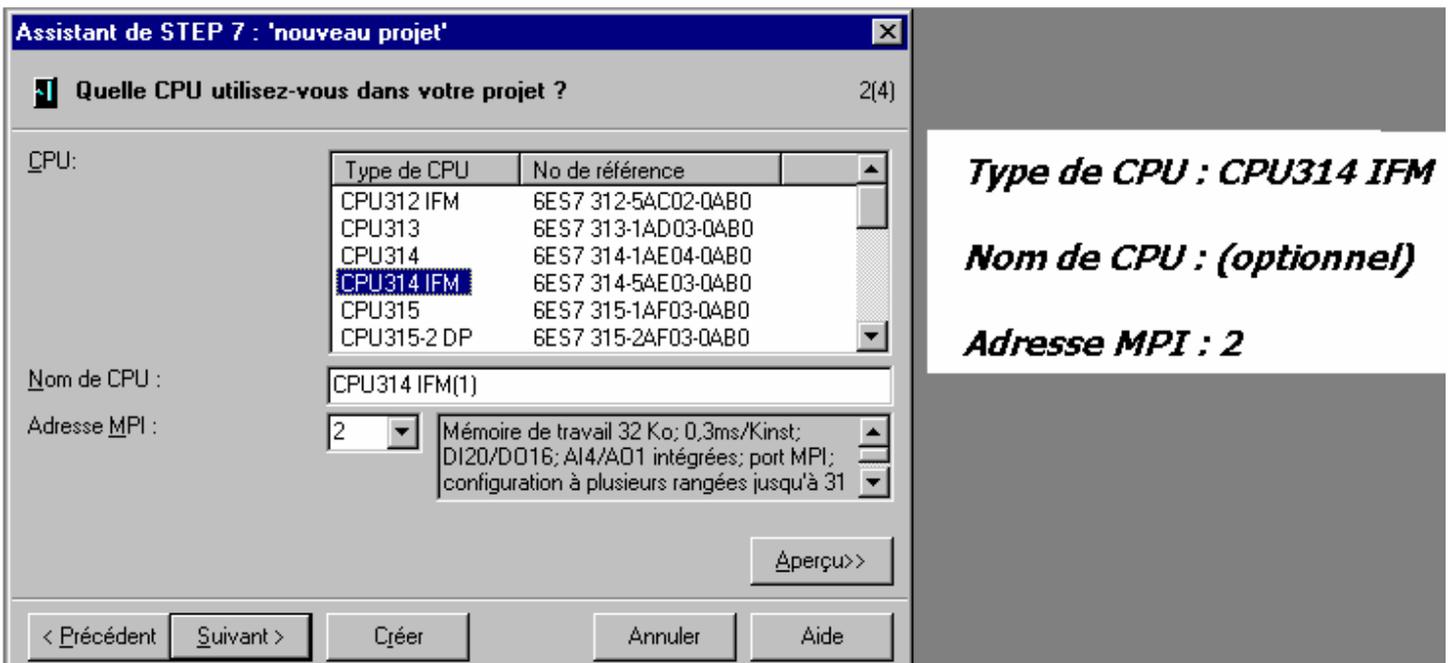
Etape 1 : sur votre PC, on cherche l'icône du logo Step 7 et on clique 2 fois



Etape 2 : l'assistant se lancera tout seul, si non on l'ouvre à partir du menu **Fichier**



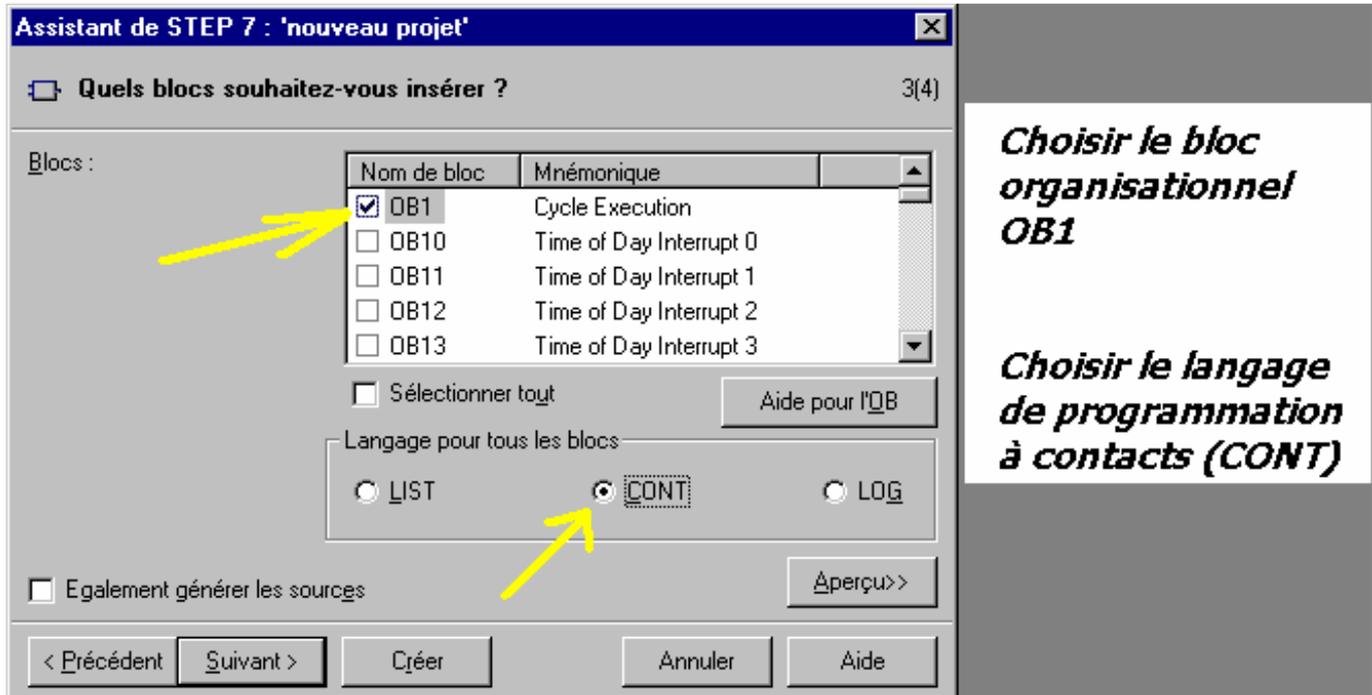
Etape 3 : Cliquez sur **Suivant** ; Cette seconde fenêtre prend la place de la fenêtre précédente. C'est la première des trois fenêtres de configuration du projet.



- Vous devez choisir le modèle du CPU selon l'automate à programmer.
- Le champ « nom de la CPU » peut être modifié pour identifier la fonction ou l'emplacement de ce CPU dans l'usine (exemple Station de pompage).
- Enfin, l'adresse de communication MPI doit être fixée à 2.

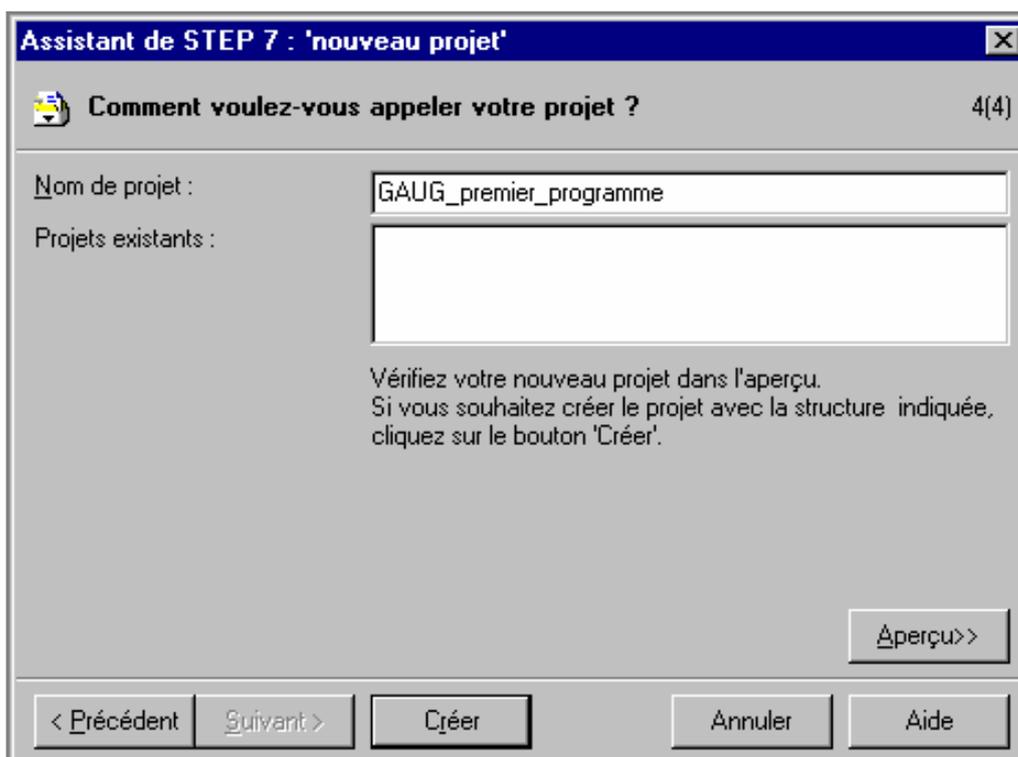
Une fois ces opérations faites, cliquez sur **Suivant**.

Etape 4 : Cette fenêtre apparaît alors. C'est la deuxième fenêtre de configuration du projet.



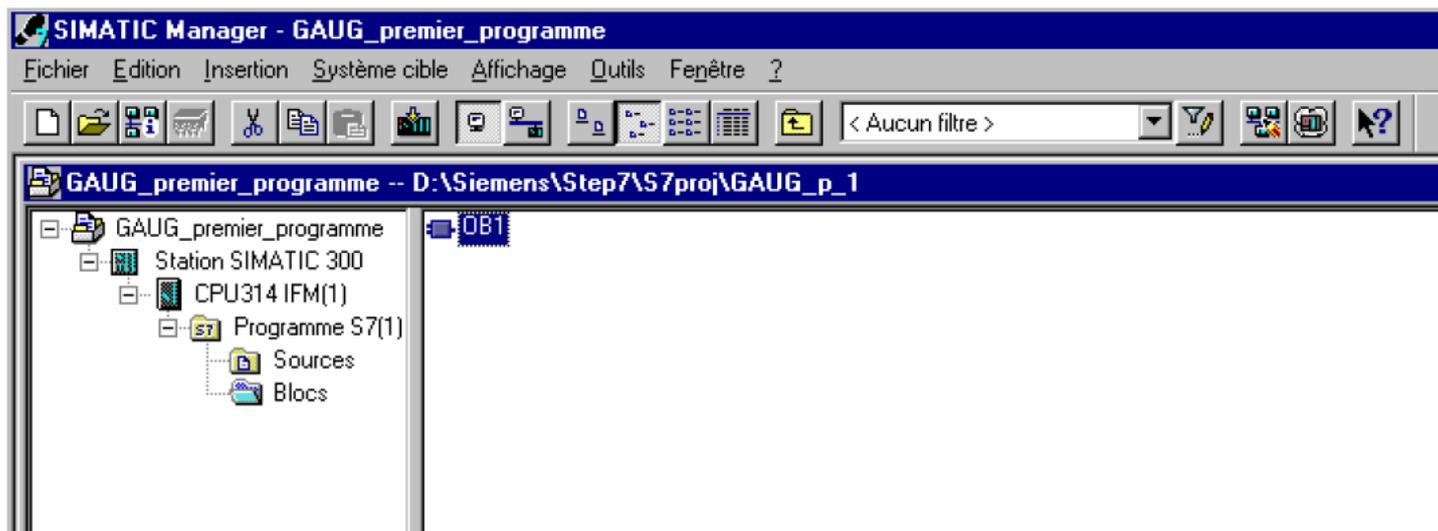
Choisir le Bloc d'organisation **OB1** et le langage de programmation à contacts (**CONT**), puis cliquez sur **Suivant**.

Etape 5 : Cette dernière fenêtre apparaît alors :



Il vous suffit alors de choisir un nom pour le projet, puis de cliquer sur **Créer**.

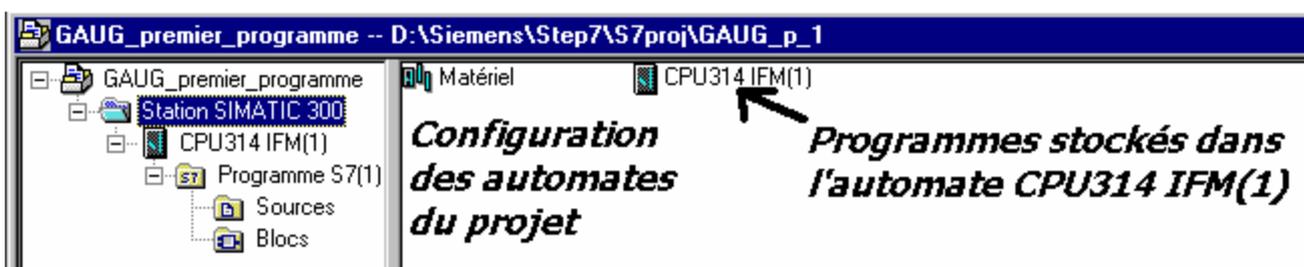
L'écran du « Simatic Manager » devrait alors avoir cette apparence :



En cliquant sur le répertoire « Premier programme » vous voyez apparaître les items « Station SIMATIC 300 » et « MPI(1) ». Le premier item mène à la définition du matériel utilisé dans ce projet, le second sur la configuration du réseau utilisé dans ce projet.



En cliquant sur le répertoire « Station SIMATIC 300 »

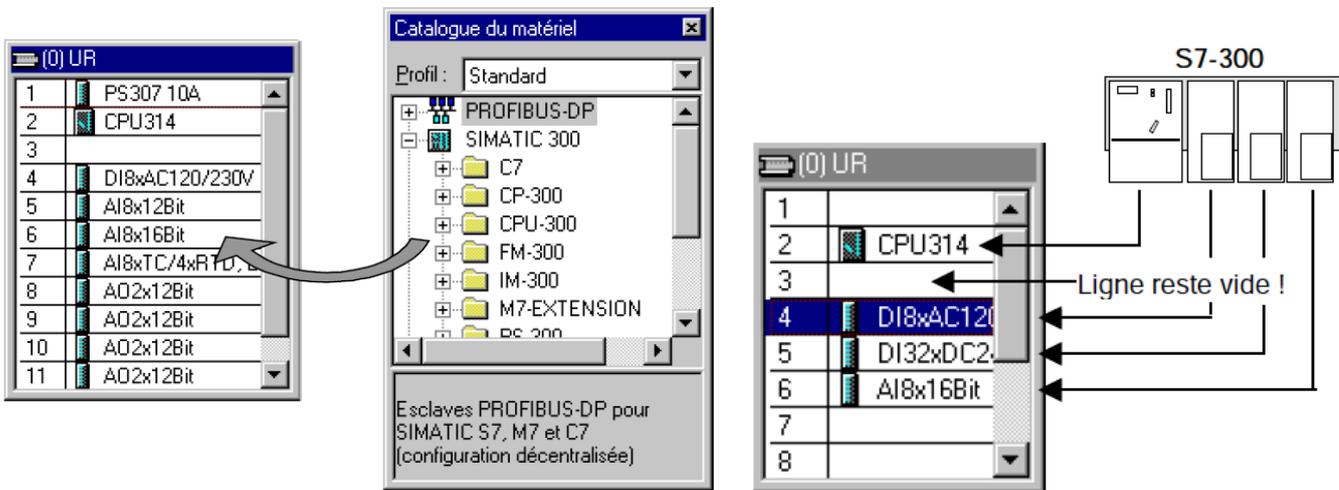
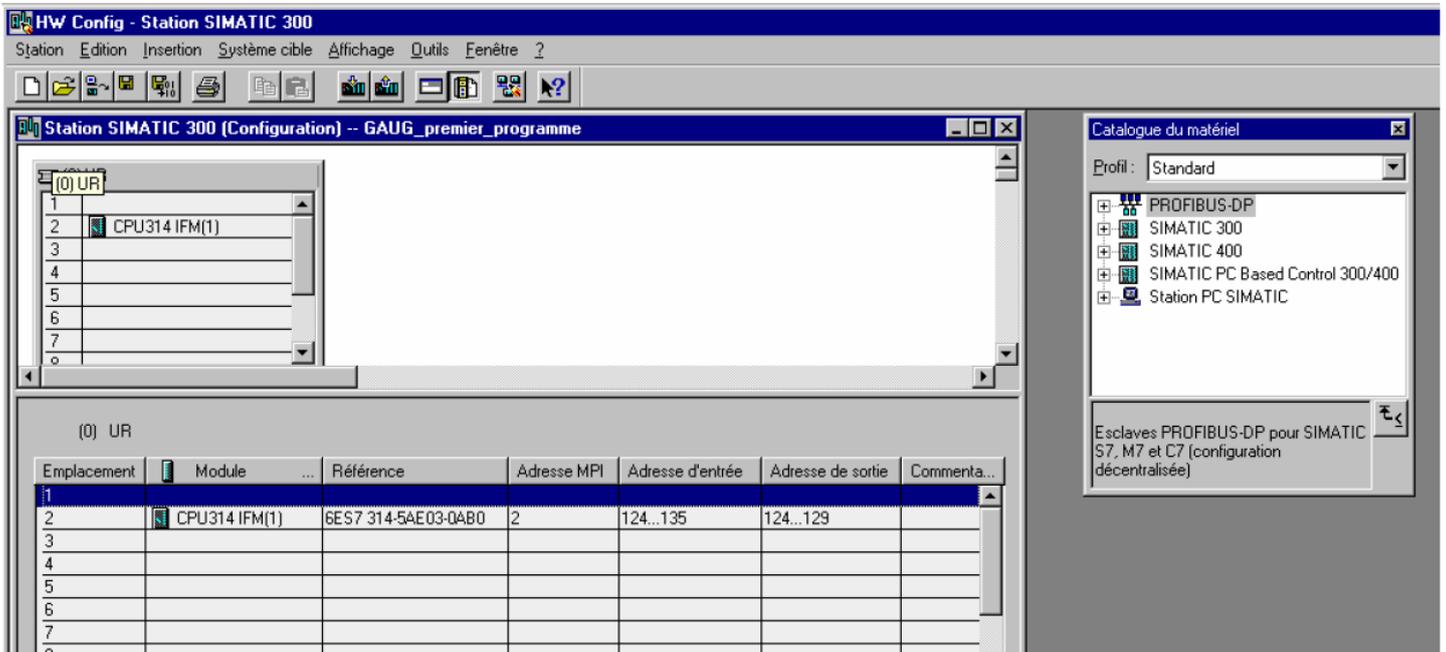


Configuration supplémentaire du matériel de l'automate

En double cliquant sur « Matériel », vous lancez le logiciel de configuration de matériel.

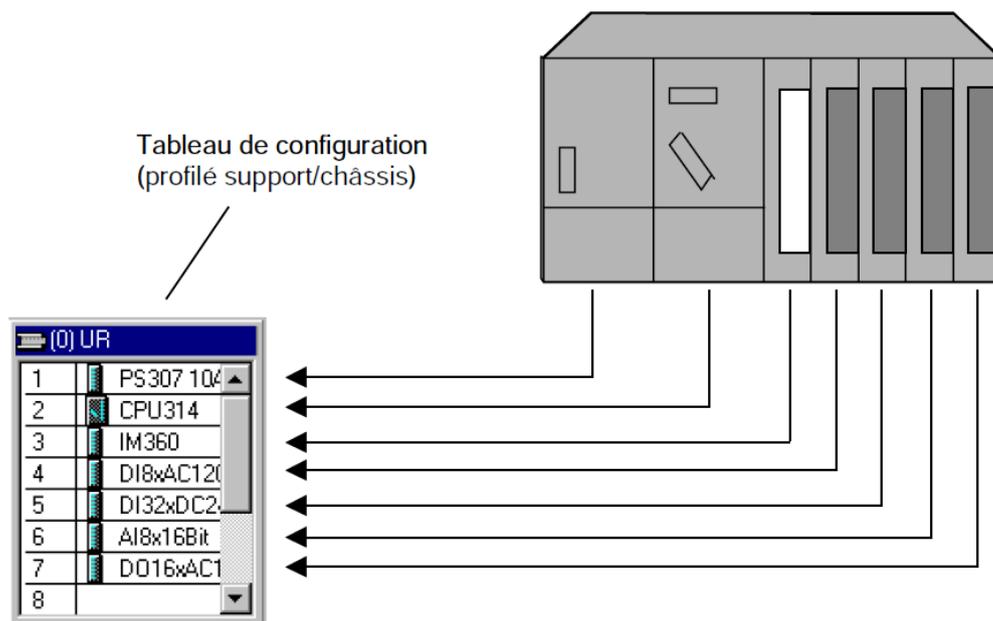
La fenêtre suivante apparaît :

1. Sélectionnez une composante matérielle dans la fenêtre "Catalogue du matériel".
2. Amenez la composante sélectionnée dans la fenêtre de station en utilisant la fonction Glisser lâcher ou double cliquer.



Respectez l'ordre suivant :

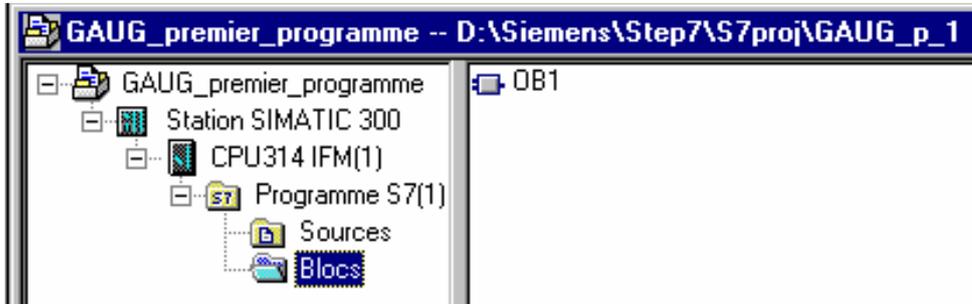
Rack 0 : c'est le profilé support/châssis



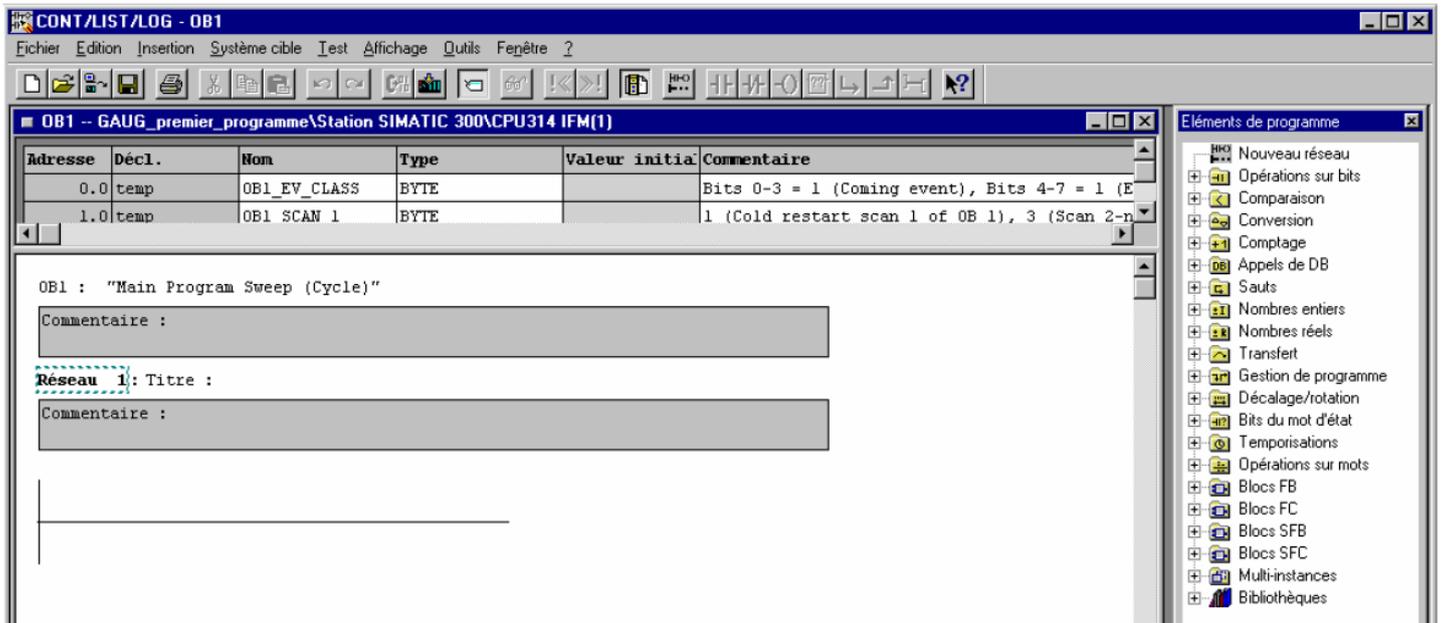
A la fin de configuration, cliquer sur menu **Fichier** puis **enregistrer**, après **quitter**.

Programmation avec le logiciel STEP 7

Après fermeture de la fenêtre configuration matériel, la fenêtre suivante s'affichera.



Double cliquez sur l'item « OB1 » pour lancer le logiciel « STEP 7 ». La fenêtre de l'éditeur apparaît alors.



Maintenant, voyons comment programmer un barreau de langage à contacts suivant :

CONT/LIST/LOG - OB1

Fichier Édition Insertion Système cible Test Affichage Outils Fenêtre ?

OB1 -- GAUG_premier_programme\Station SIMATIC 300\CPU314 IFM(1)

Adresse	Décl.	Nom	Type	Valeur initia	Commentaire
0.0	temp	OB1_EV_CLASS	BYTE		Bits 0-3 = 1 (Coming even
1.0	temp	OB1_SCAN 1	BYTE		1 (Cold restart scan 1 of

OB1 : "Main Program Sweep (Cycle)"

Commentaire :

Réseau 1: Titre :

Commentaire :

???

1) Placer le curseur ici **2) Puis cliquer cet icône**

Cliquez ensuite sur l'icône de contact normalement ouvert, puis inscrire le numéro du contact d'entrée I124.0 ou E124.0 selon API

Pour compléter le réseau, positionner le curseur à la fin de la zone du réseau 1 (voir image ci-dessous), puis cliquez sur l'icône de la bobine de sortie et inscrire le numéro de la bobine, soit Q124.7.ou A124.7 selon API

CONT/LIST/LOG - OB1

Fichier Édition Insertion Système cible Test Affichage Outils Fenêtre ?

OB1 -- GAUG_premier_programme\Station SIMATIC 300\CPU314 IFM(1)

Adresse	Décl.	Nom	Type	Valeur initia	Commentaire
0.0	temp	OB1_EV_CLASS	BYTE		Bits 0-3 = (Coming event), Bits 4-
1.0	temp	OB1_SCAN 1	BYTE		1 (Cold restart scan 1 of OB 1), 3 (

OB1 : "Main Program Sweep (Cycle)"

Commentaire :

Réseau 1: Titre :

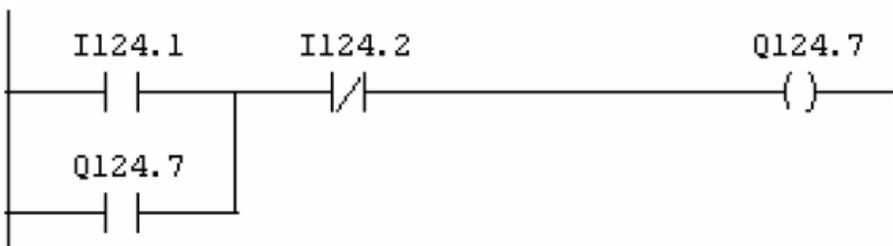
Commentaire :

I124.0

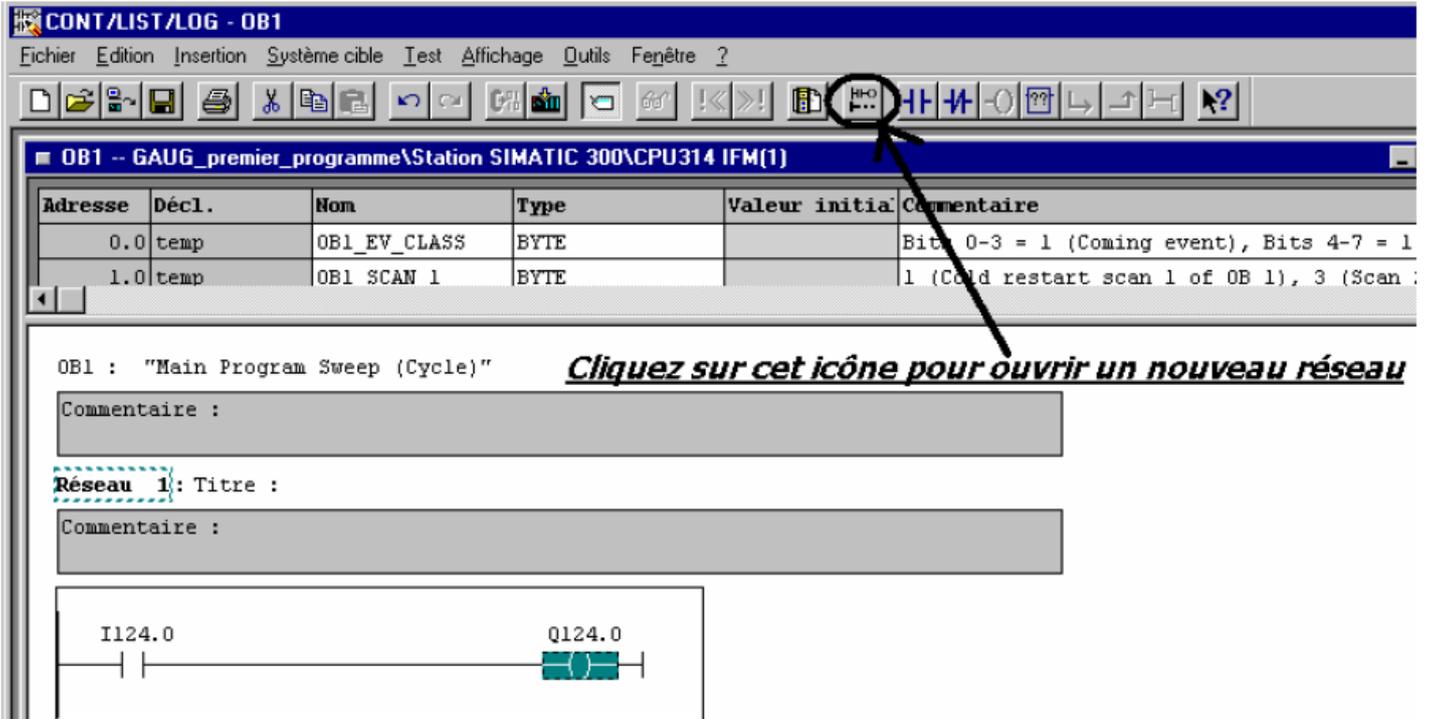
???

1) Placer le curseur ici **2) Puis cliquez cet icône**

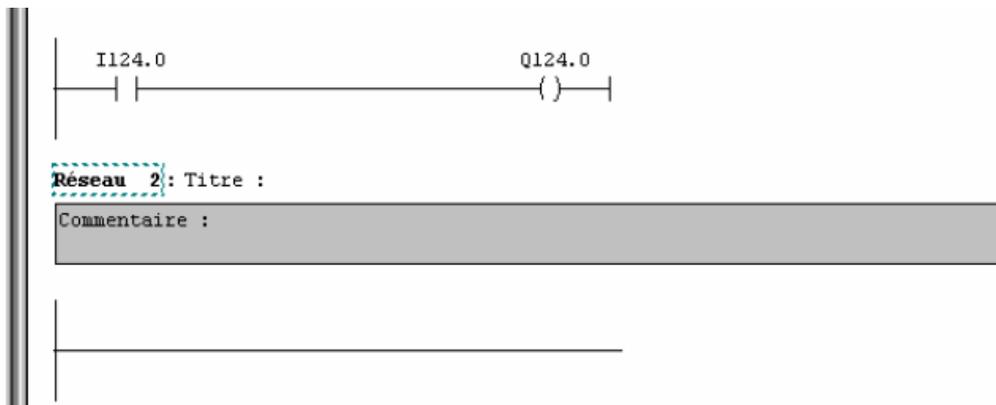
Une fois le premier barreau programmé, programmons ensuite le barreau suivant :



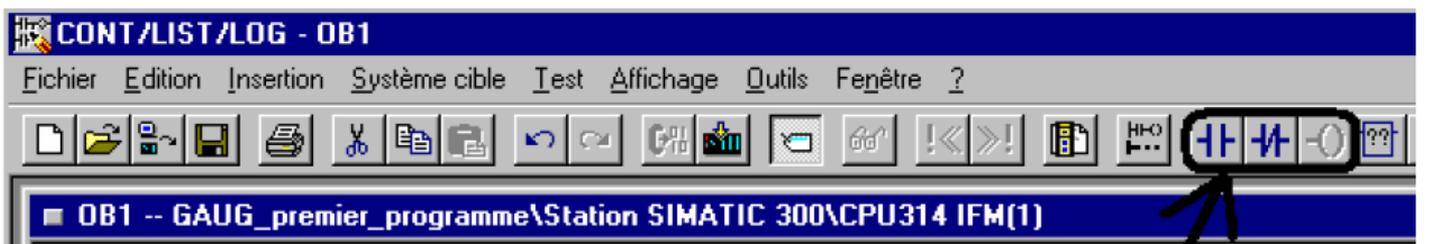
Il faut introduire un nouveau réseau pour programmer ce nouveau barreau.



Un nouveau réseau est créé .



Programmer la première ligne du barreau tel que montré à la figure suivante, puis inscrire les adresses.



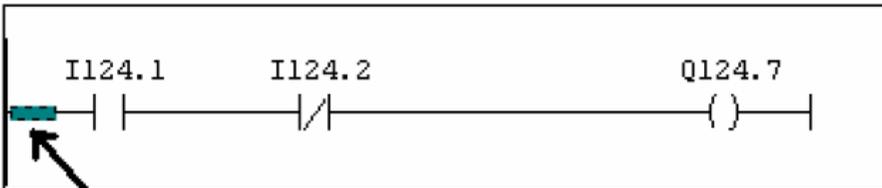
Réseau 2: Titre :

Commentaire :

Insérer les contacts d'entrée et de sortie

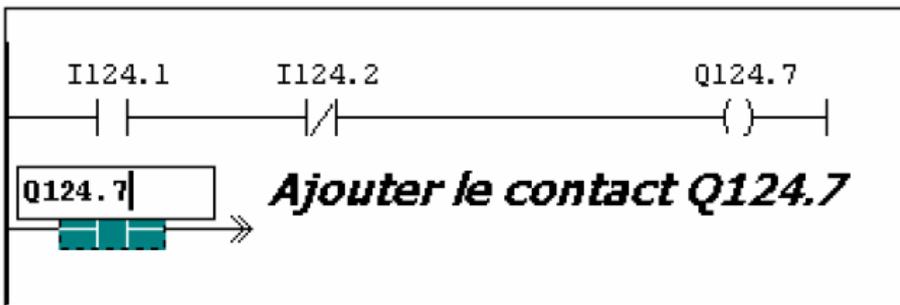


Ouvrir la branche parallèle



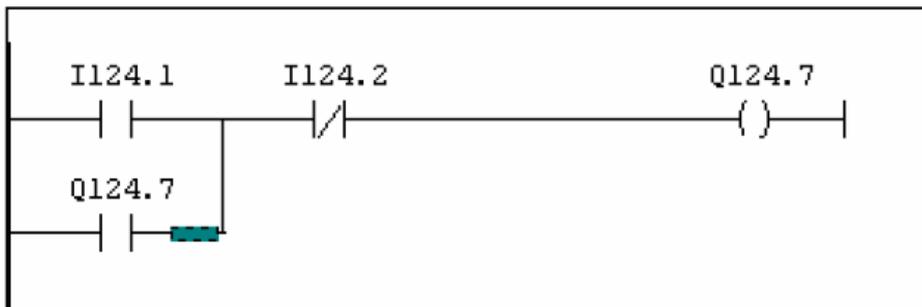
Placer le curseur ici et appuyer sur la touche [F8]

Insérer le contact ouvert avec adresse de Q124.7



Il faut maintenant fermer cette branche. On y arrive en appuyant sur la touche F9.

Le réseau 2 aura donc cette apparence.

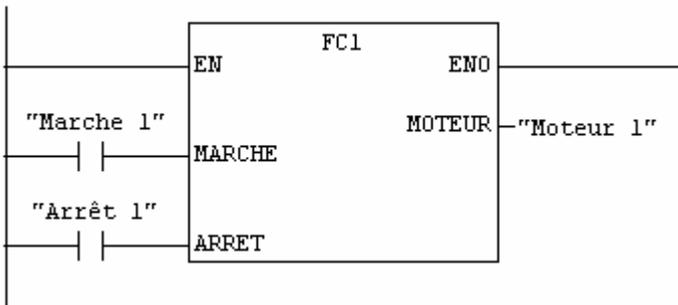


Sauvegarder ce programme en cliquant sur l'option « Sauvegarde » du menu déroulant « Fichier ».

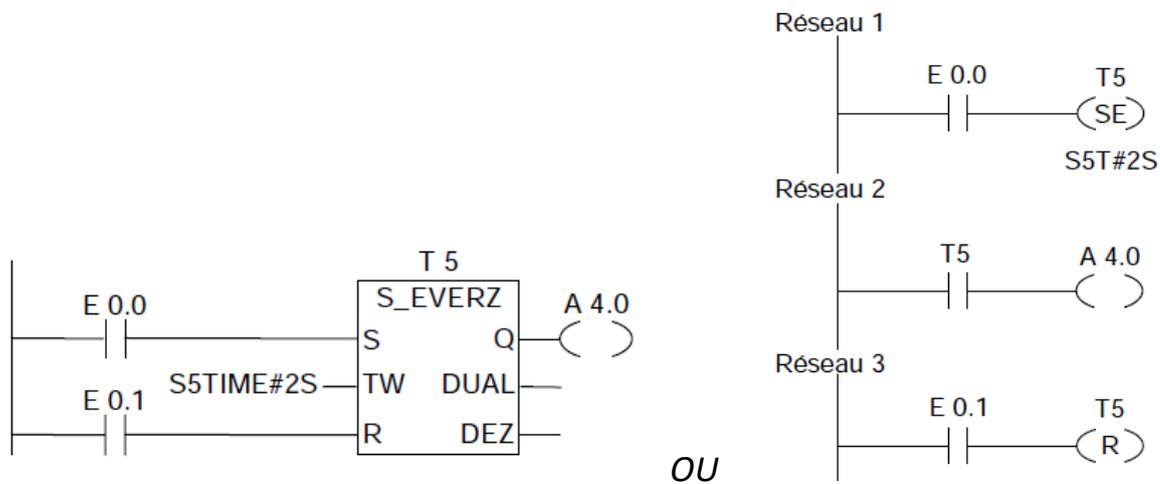
Autres fonctions souvent employées :

✚ Un bloc de fonction FC correspond à une sous-routine appelée par d'autres blocs.

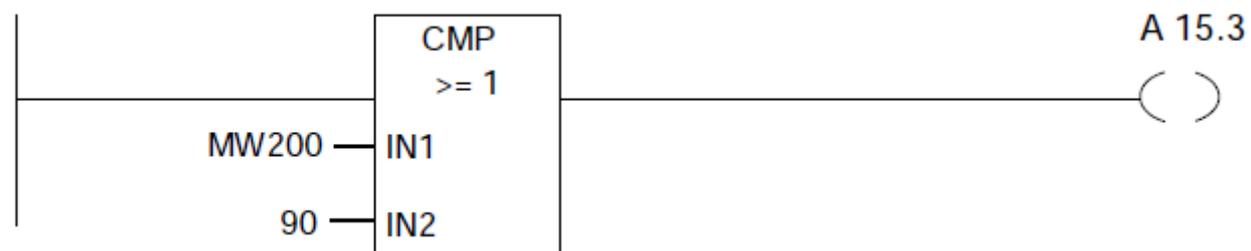
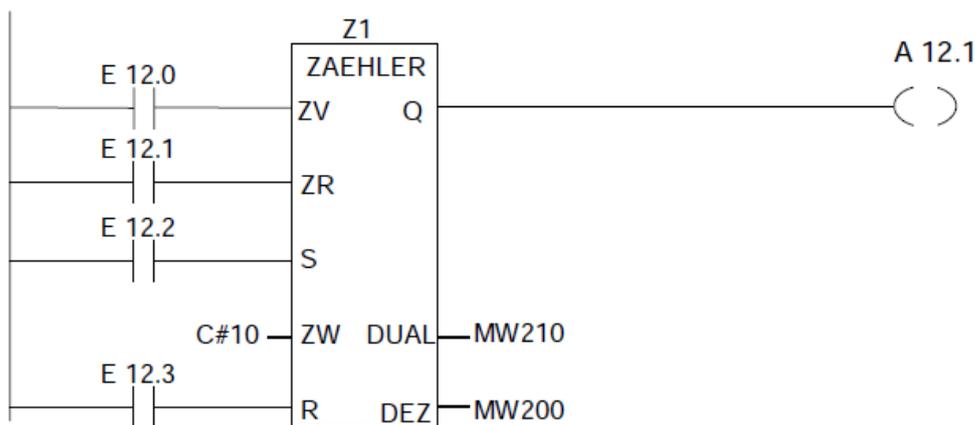
Exemple : démarrage direct



Temporisation :



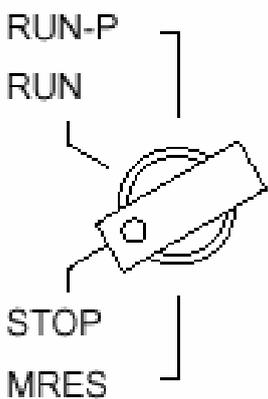
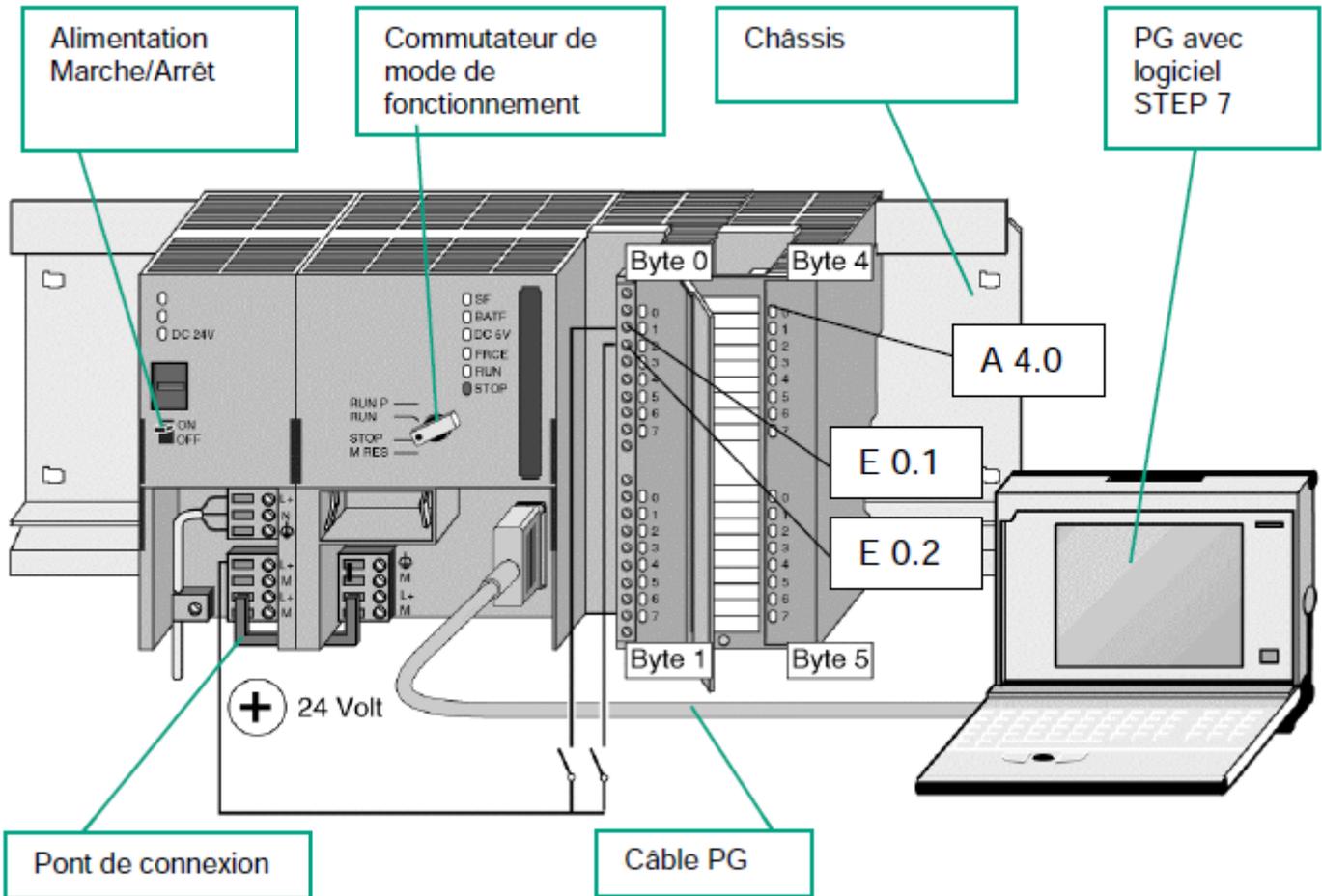
Comptage :



Autres à découvrir par pratique ou expérience

Charger le programme dans l'automate

L'automate que vous utilisez en laboratoire possède un module de source d'alimentation PS307 de 2 Ampères et le module de CPU 314-IFM de la série S7-300



En mode STOP, l'automate n'exécute pas le programme. Dans ce mode il est possible de transférer un programme de l'ordinateur vers l'automate et de transférer un programme de l'automate vers l'ordinateur. Il est aussi possible de modifier le programme dans l'automate.

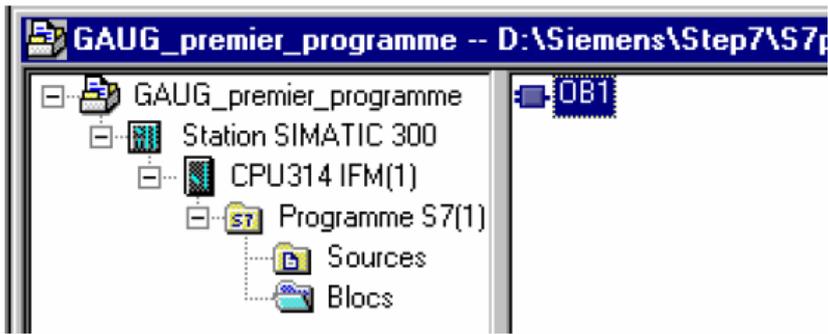
En mode RUN, l'automate exécute le programme. Dans ce mode, mode il est possible de transférer un programme de l'automate vers l'ordinateur.

En mode RUN-P (RUN-PROGRAM), l'automate exécute le programme. Dans ce mode il est possible de transférer un programme de l'ordinateur vers l'automate et de transférer un programme de l'automate vers l'ordinateur. Il est aussi possible de modifier le programme dans l'automate.

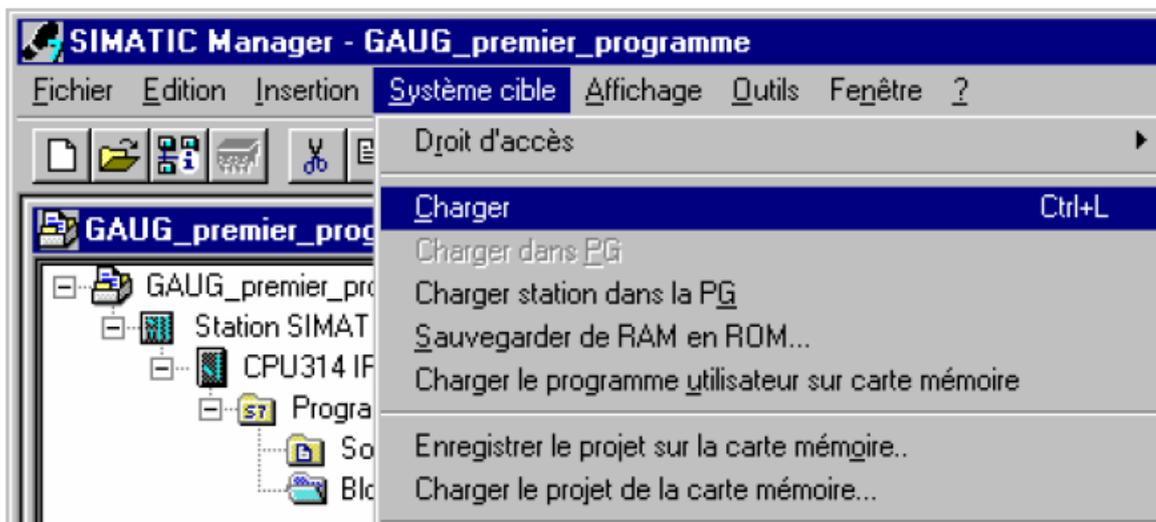
En mode MRES, avec appuis quelques seconde le programme en mémoire est effacé ou autrement on fait RESET de l'API (RAZ)

Transfert du programme vers l'API :

Pour ce faire, retourner au SIMATIC manager et sélectionner l'item Blocs. La fenêtre du SIMATIC manager devrait ressembler à :



Sélectionner alors l'item OB1



La fenêtre suivante apparaît :



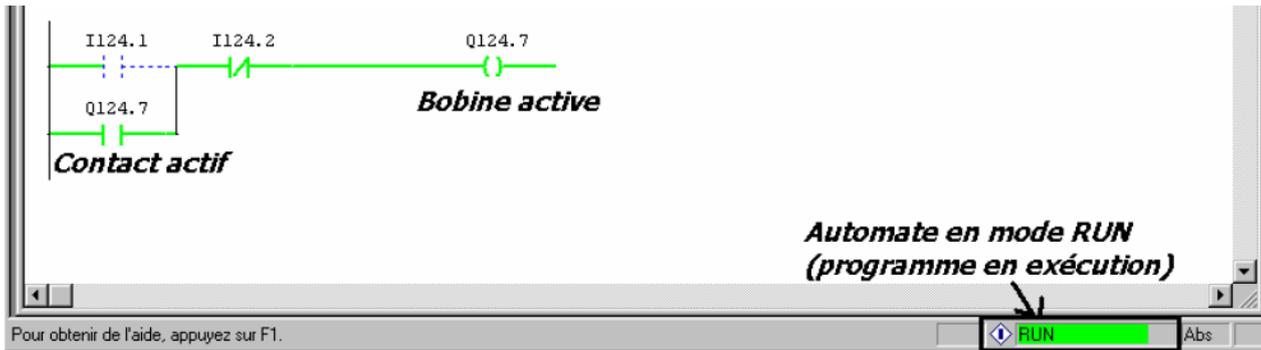
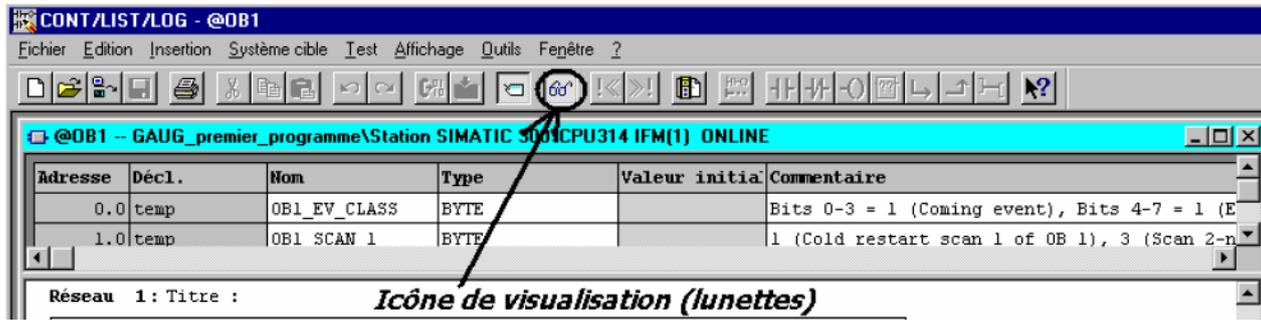
Cliquer alors sur OUI pour écraser l'ancien programme.

Visualisation du programme dans l'automate :

L'état de fonctionnement du programme dans l'automate peut être visualisé de diverse façon. Bien sûr, l'observation des voyants des sorties permet de vérifier l'état des sorties en fonction de l'état des entrées.

Une autre façon de visualiser, c'est via l'éditeur de programme « STEP7 ».

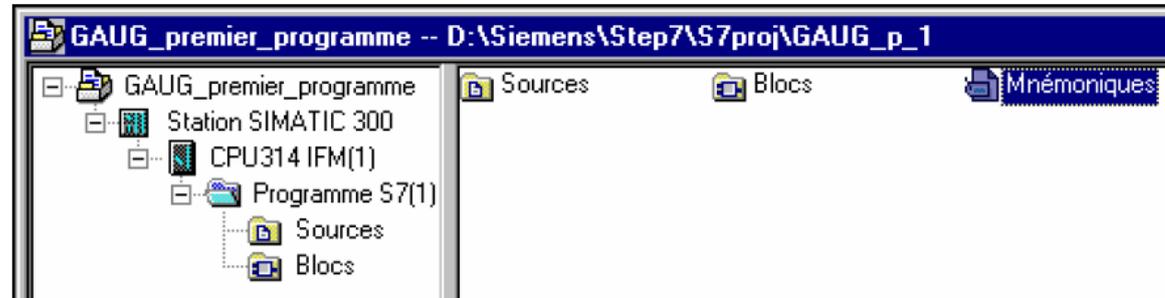
Pour visualiser l'état des variables dans l'automate, cliquer sur icône de la paire de lunettes.



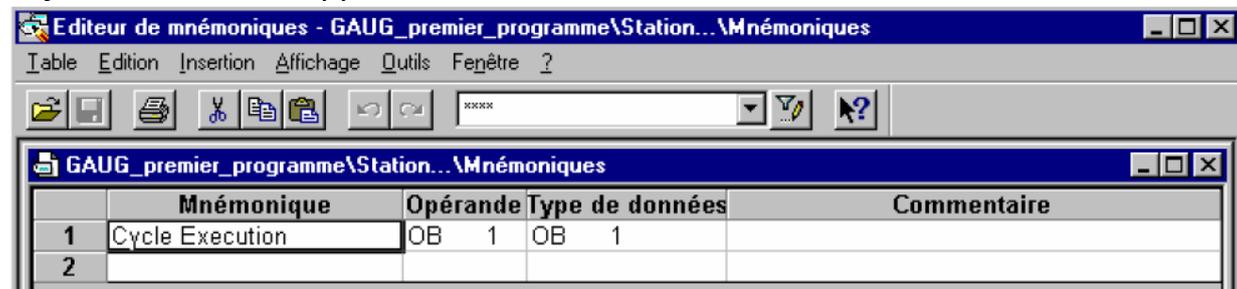
L'utilisation d'étiquettes :

Les étiquettes (appelées aussi mnémoniques) permettent de donner aux variables d'entrées, de sorties ou de mémoire interne un nom plus révélateur que I124.0 par exemple.

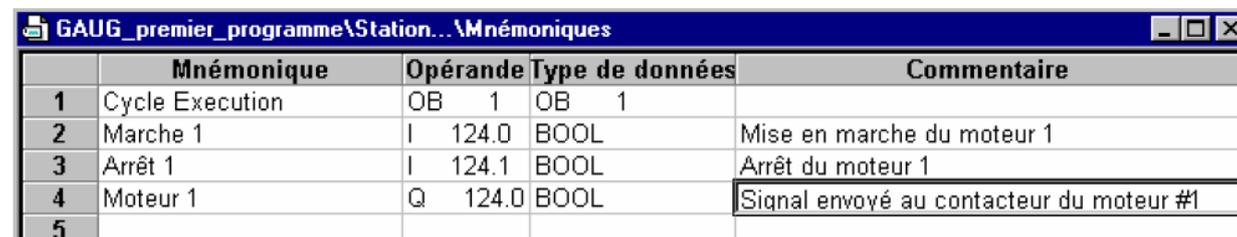
Choisir l'item « **Programme S7** » du menu de gauche.



La fenêtre suivante apparaît alors :



On affecte pour chaque variable entrée ou sortie un nom et un commentaire si nécessaire.



EXEMPLES D'APPLICATION

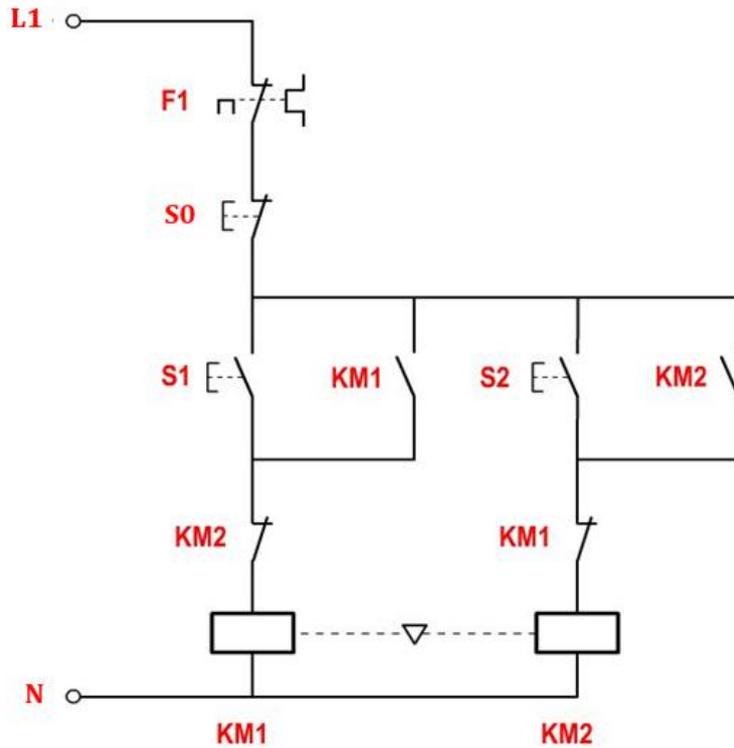
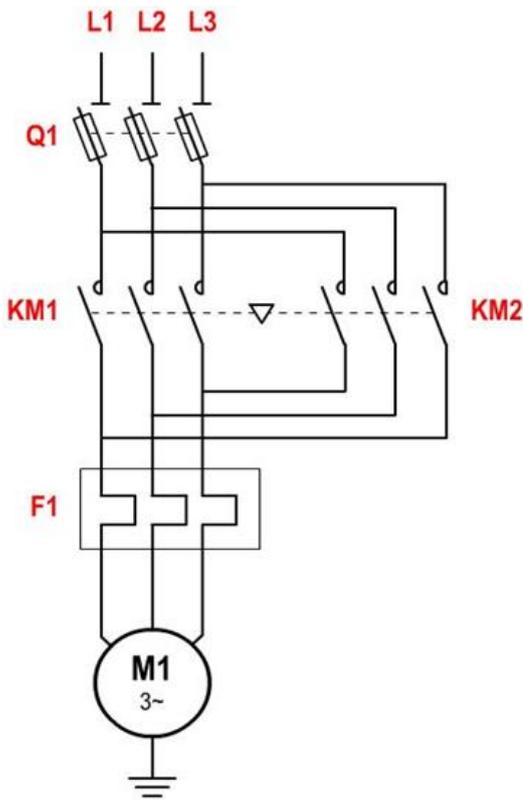
Exemple : On demande de réaliser le montage : démarrage direct 2 sens de rotations

Circuit de puissance :

circuit de commande :

Reste le même

la logique câblé sera remplacée par logique programmée



Affectations des entrées et des sorties :

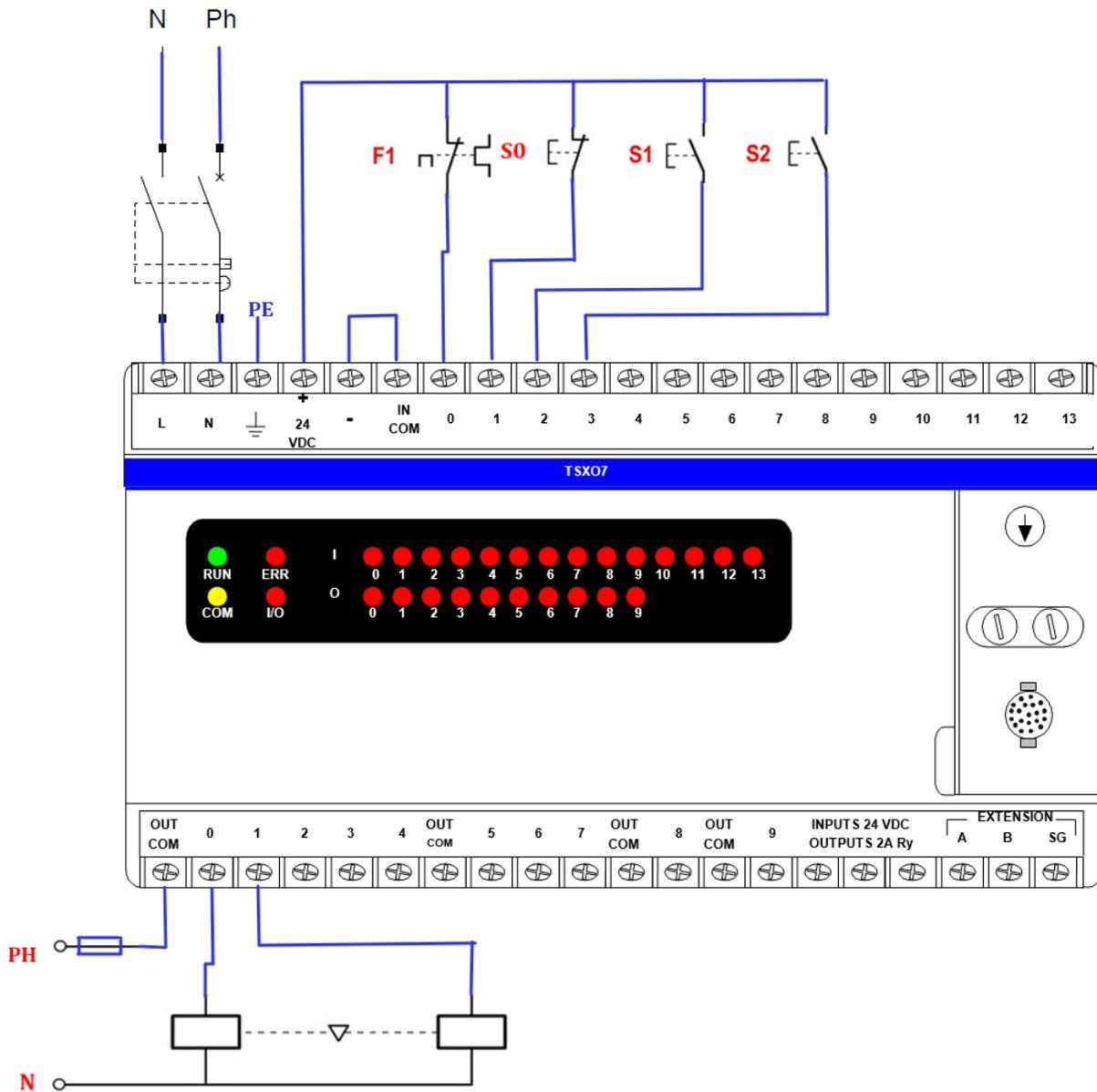
Les sorties :

Fonction	Mnémonique	Entrée
Contacteur sens 1	KM1	% Q0.0
Contacteur sens 2	KM2	% Q0.1

Les entrées :

Fonction	Mnémonique	Entrée
Contact relais thermique	F1	% I0.0
BP Arrêt	S0	% I0.1
BP marche sens 1	S1	% I0.2
BP marche sens 2	S2	% I0.3

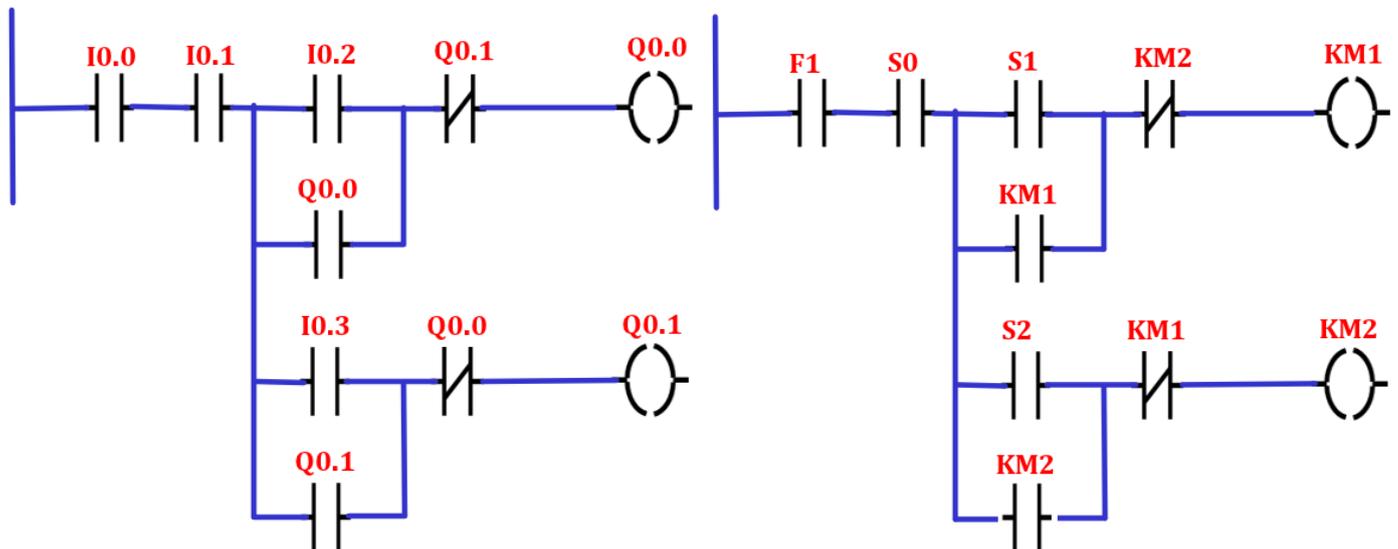
Schéma de branchement :



Programme :

Avec Opérandes

Avec Mnémoniques

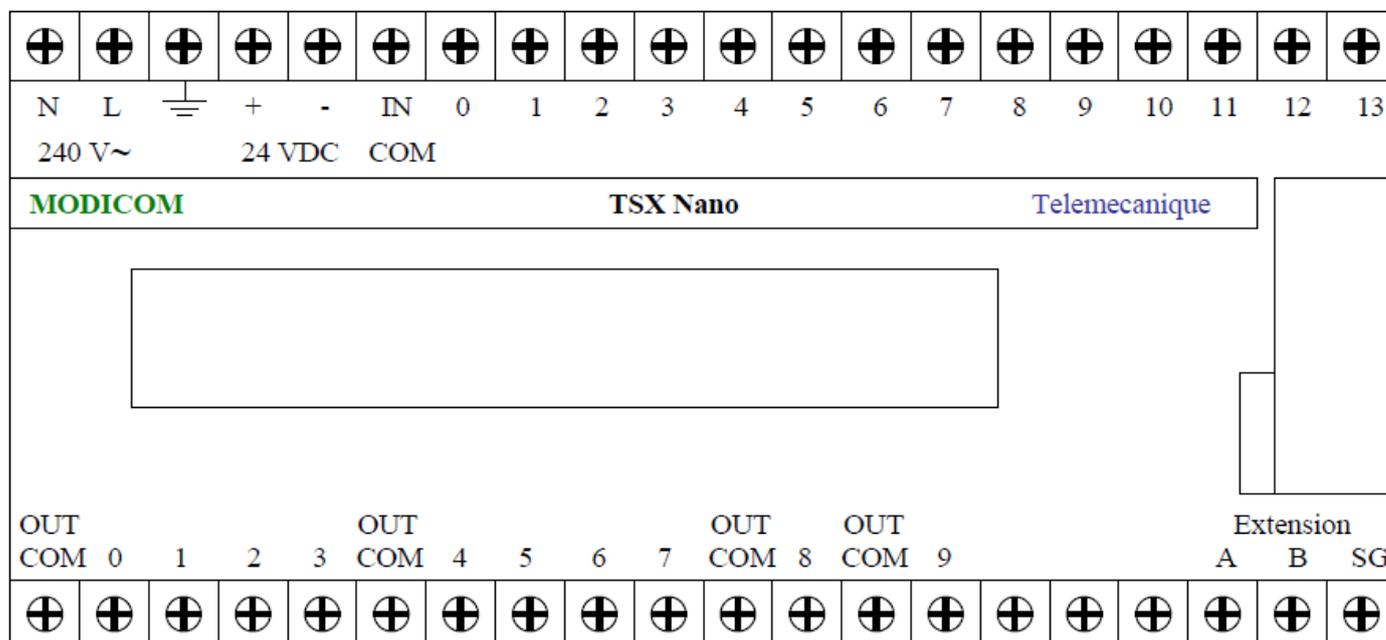


Exercice 1 : On demande de réaliser le montage : démarrage direct 2 sens de rotations commandé de 2 endroits différents avec signalisation H1 défaut, H2 Arrêt, et H3 marche

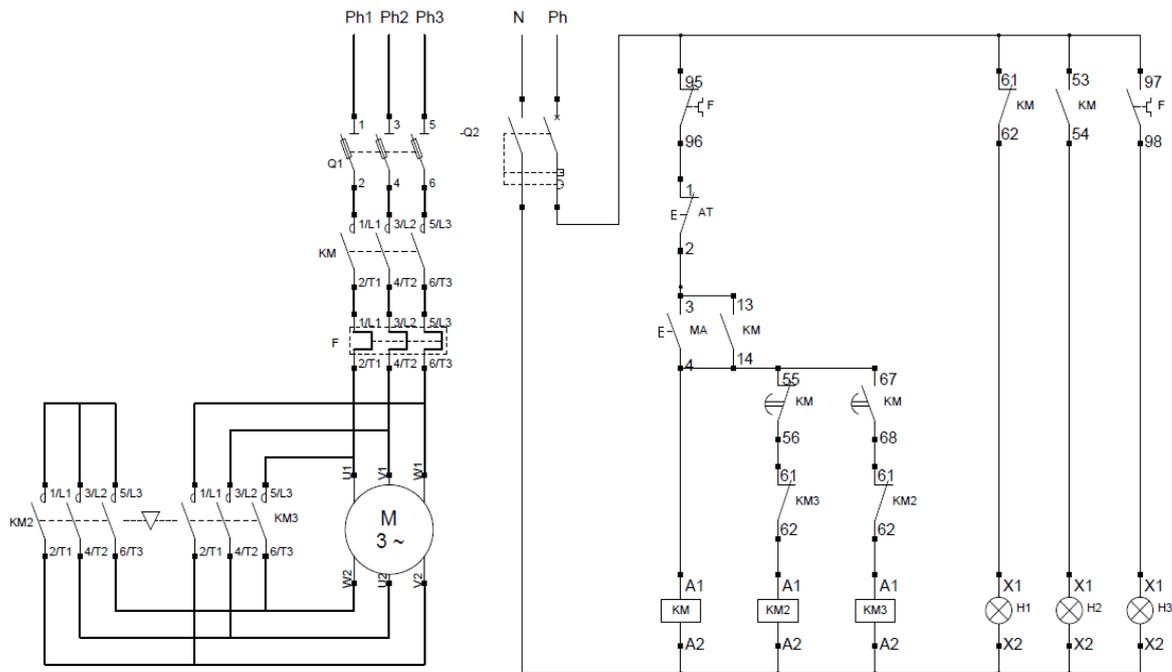
BP Arrêt : S0, S0' BP marche sens1 :S1, S1' BP marche sens2 :S2, S2'

Travail demandé :

- Circuits de commande logique câblée
- Circuits de commande logique programmée
- Affectation des entrées :
- Affectation des sorties
- Câblage de l'API
- Programme



Exercice 2 : On demande de réaliser le montage étoile triangle commandé par API



Travail demandé :

- Circuits de commande logique câblée
- Circuits de commande logique programmée
- Affectation des entrées :
- Affectation des sorties
- Câblage de l'API
- Programme

