

République Algérienne Démocratique et Populaire Ministère de l'Enseignement
Supérieur et de la Recherche Scientifique
Université Mohamed BOUDIAF-M'sila



Faculté de Technologie
Département de Génie Electrique

Polycopie de TP

Spécialité : Licence en Electrotechnique

Intitulé :

TP Automatismes Industriels

Par le Docteur :
Brahim LADGHEM CHIKOUCHE

M'sila 2024

1. AUTOMATE PROGRAMMABLE INDUSTRIEL API

Automate Programmable Industriel est une machine électronique programmable par un personnel non informaticien et destiné à piloter en ambiance industrielle et en temps réel des procédés industriels.

Le développement de l'industrie à entraîner une augmentation constante des fonctions électroniques présentes dans un automatisme c'est pour ça que l'API s'est substituée aux armoires à relais en raison de sa souplesse dans la mise en œuvre, mais aussi parce que dans les coûts de câblage et de maintenance devenaient trop élevés.

2. ROLE D'UN AUTOMAT

Le rôle de l'automate est de réagir aux changements d'état de ses entrées en modifiant l'état de ses sorties selon une loi de contrôle déterminée a priori par le concepteur du système. Cette loi est dite combinatoire si, à chaque instant, l'état des sorties peut être directement déduit de l'état des entrées. Elle est de type séquentiel, s'il faut en plus tenir compte de l'évolution antérieure du système. Cette dernière peut en général être complètement décrite par l'état d'un nombre fini de variables logiques mémorisées au sein de l'automate.

L'UTILISATIONS D'AUTOMATISME

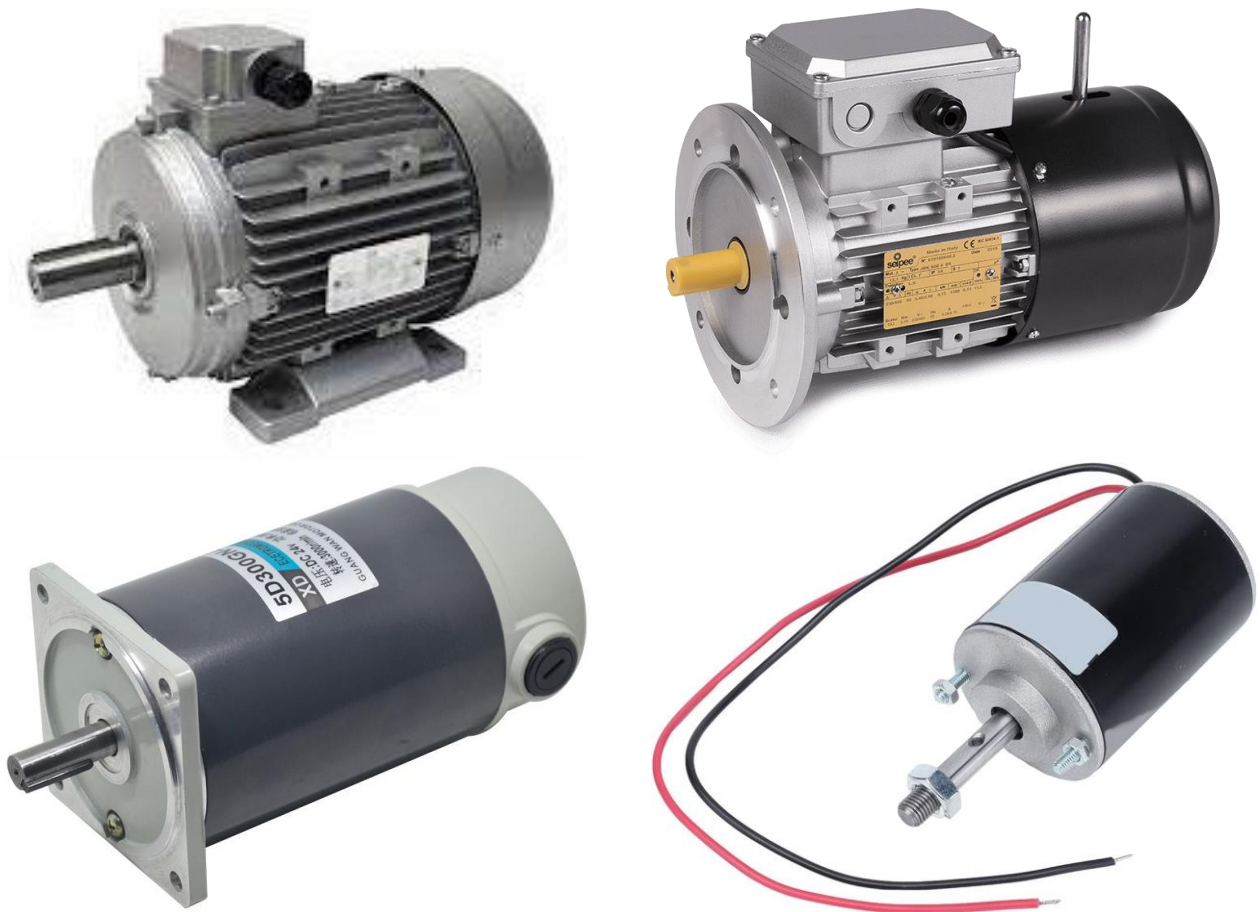
- Accroître la productivité (rentabilité, compétitivité) du système
- Améliorer la flexibilité de production ;
- Améliorer la qualité du produit
- Adaptation à des contextes particuliers tel que les environnements hostiles pour l'homme (milieu toxique, dangereux. Nucléaire...).

3. TP AUTOMATES PROGRAMMABLES INDUSTRIELS

- **TP01** : Prise en main de l'environnement API :
 - Simulation d'un système automatisé,
 - Revue des différents logiciels.
 - Introduction au logiciel Step7 de Siemens
- **TP02** : Mise en œuvre d'un API :
 - Configuration Hardware.
 - Initiation à la programmation en Ladder.
 - Utilisation des entrées/sorties TOR :
 - Utilisation des relais,

- Contacteur, (faire éventuellement les câblages nécessaires).
- **TP03** : Mise à l'échelle des entrées/sorties analogiques :
 - Mesure du signal à l'entrée d'un capteur (faire éventuellement les câblages nécessaires).
 - Utilisation de quelques blocs usuels :
 - Opérations arithmétiques,
 - Temporisateurs,
- **TP04** : Commande d'un moteur asynchrone dans les deux sens de rotation avec et sans freinage.
- **TP05** : Démarrage Etoile/Triangle d'un moteur asynchrone.
- **TP06** : Contrôle d'un processus basé sur des vérins pneumatiques.

4. Les Actionneurs électriques :



5. Pré-actionneur électriques

- Disjoncteurs triphasés :



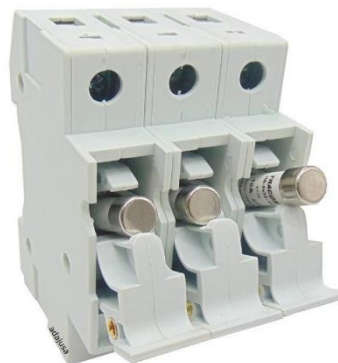
- Contacteurs triphasés :



- Relais thermiques triphasés :



- Sectionneur porte fusible triphasé :



- Capteurs :



Fin de course



Capteur capacitif



Capteur inductif



Capteur photoélectrique

6. Les Actionneurs pneumatiques :



7. Les Pré-Actionneurs pneumatiques :



8. Les automates programmables

Un automate programmable industriel (ou API) est un dispositif électronique programmable destiné à automatiser des processus tels que la commande de machines au sein d'une usine et à piloter des robots industriels par exemple.

L'automate programmable reçoit des données par ses entrées, celles-ci sont ensuite traitées par un programme défini, le résultat obtenu étant délivré par ses sorties. Ce cycle de traitement est toujours le même, quel que soit le programme, néanmoins le temps d'un cycle d'API varie selon la taille du programme et la puissance de l'automate.

Ce dispositif a l'avantage d'être composé d'éléments particulièrement robustes et possède d'énormes capacités d'exploitation. En contrepartie, il est beaucoup plus cher que les solutions informatiques classiques comme les micro-ordinateurs.

- **Avantages multiples**

L'API présente des intérêts multiples :

- Il est réactif et obéit promptement aux informations qui lui livrent les capteurs,
- Il est solide et résiste aux températures, aux vibrations, voire aux coupures d'alimentation.
- Il assure une maintenance facile et son redémarrage est rapide.

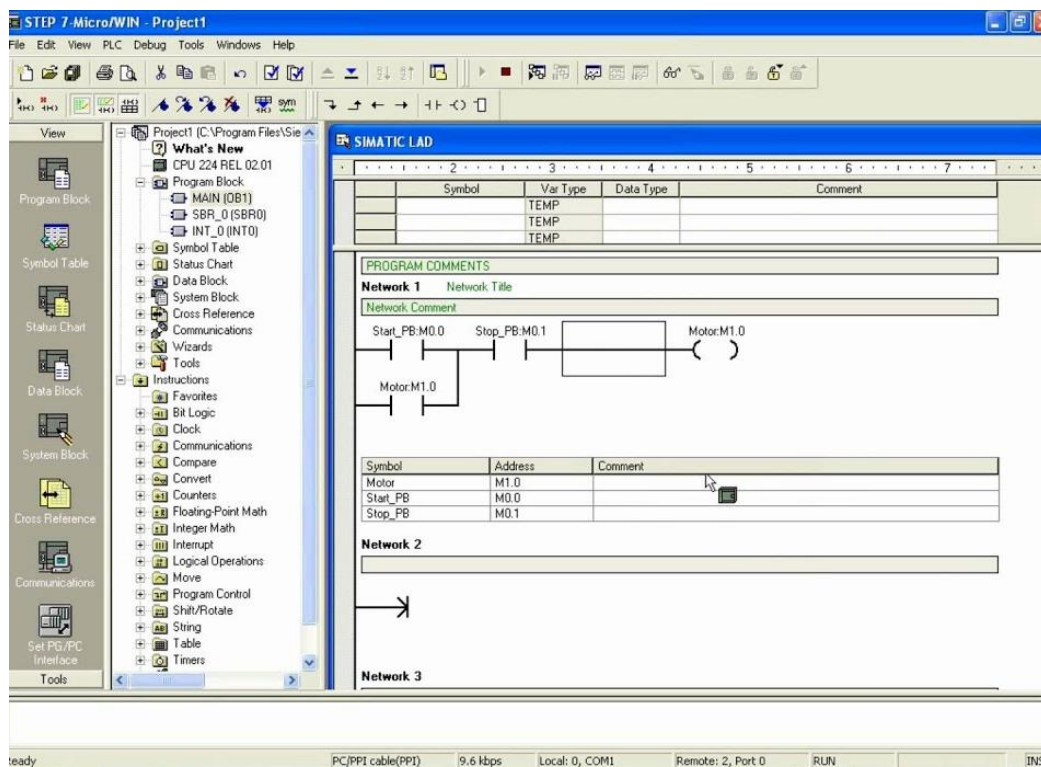
On peut ainsi le faire fonctionner sous des latitudes et des environnements industriels différents, même pollués ou exposés à des parasites électromagnétiques. Il est parfaitement adapté aux exigences de la mondialisation et des firmes multinationales, voire aux contraintes des pandémies.



- **Logiciel SIMATIC Step7 de programmation des automates de marque SIEMENS de type S200**

STEP 7 est le progiciel de base pour la configuration et la programmation de systèmes d'automatisation SIMATIC. Il fait partie de l'industrie logicielle SIMATIC. Le progiciel de base STEP 7 existe en plusieurs versions

- STEP 7-Micro/DOS et STEP 7-Micro/Win pour des applications autonomes simples sur SIMATIC S7-200.
- STEP 7 pour des applications sur SIMATIC S7-300/400, SIMATIC M7-300/400 et SIMATIC C7 présentant des fonctionnalités supplémentaires par rapport à la version précédente ST-200.



9. Langages de programmation

L'éditeur de programme contient la logique du programme et une table de variables locales dans laquelle vous affectez des mnémoniques aux variables locales temporaires. Les sous-programmes et les programmes d'interruption apparaissent sous forme d'onglets au bas de la fenêtre de l'éditeur de programme. Cliquez sur ces onglets pour aller et venir entre sous-programmes, programmes d'interruption et programme principal. STEP 7-Micro/WIN vous propose trois éditeurs pour la création de votre programme : schéma à contacts (CONT-SIMATIC ou LD-CEI), liste d'instructions (LIST) et logigramme (LOG-SIMATIC ou FBD-CEI). Avec quelques restrictions, les programmes

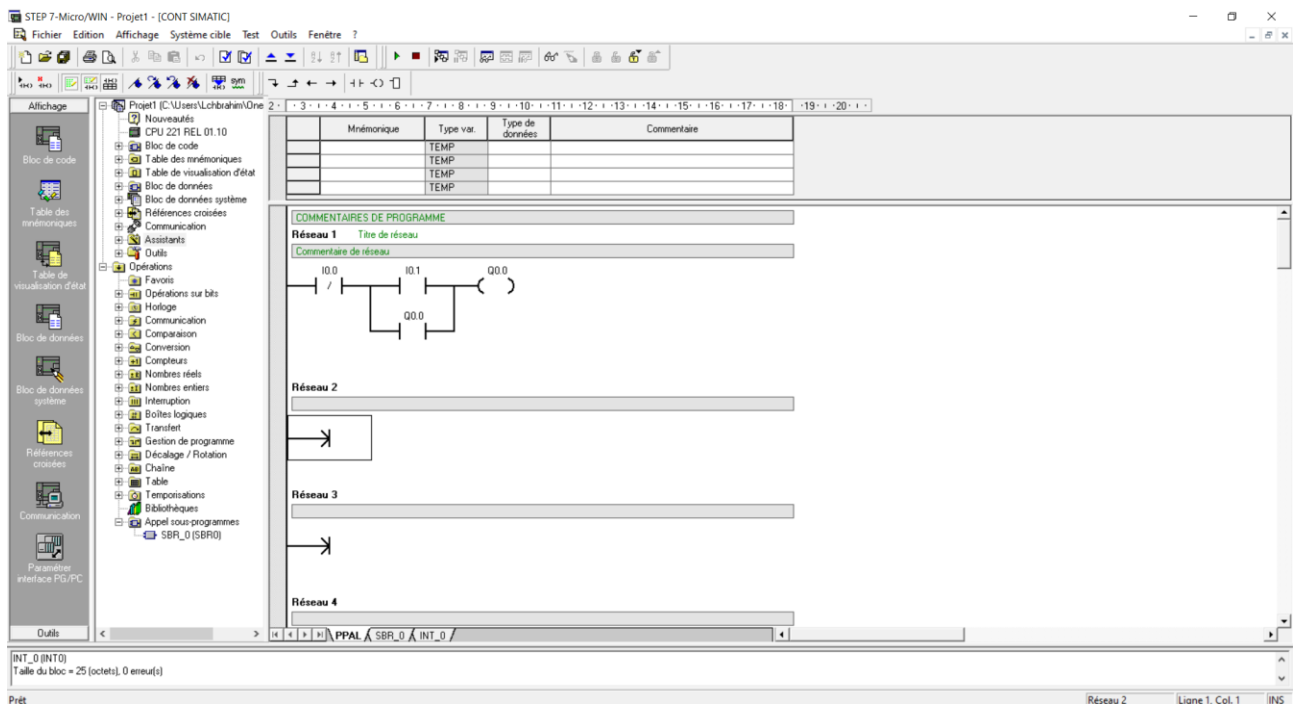
écrits dans l'un quelconque de ces éditeurs de programme peuvent être affichés et édités dans les autres éditeurs.

Les langages de programmation sont : CONT, LIST et LOG

- **Schéma à contacts (CONT)**

L'éditeur CONT (ou LD) affiche le programme sous forme de représentation graphique similaire aux schémas de câblage électriques. Les programmes CONT permettent au programme d'émuler le flux de courant électrique partant d'une source de tension, à travers une série de conditions d'entrée logiques validant, à leur tour, des conditions de sortie logiques. Un programme CONT comprend une barre d'alimentation à gauche qui est alimentée en courant. Les contacts fermés permettent au courant de circuler à travers eux vers l'élément suivant alors que les contacts ouverts bloquent le trajet du courant. La logique est subdivisée en réseaux. Le programme est exécuté réseau par réseau, de la gauche vers la droite et de haut en bas comme indiqué par le programme. L'exemple ci-dessous montre un exemple de programme CONT. Les différentes opérations sont représentées par des symboles graphiques de trois types fondamentaux.

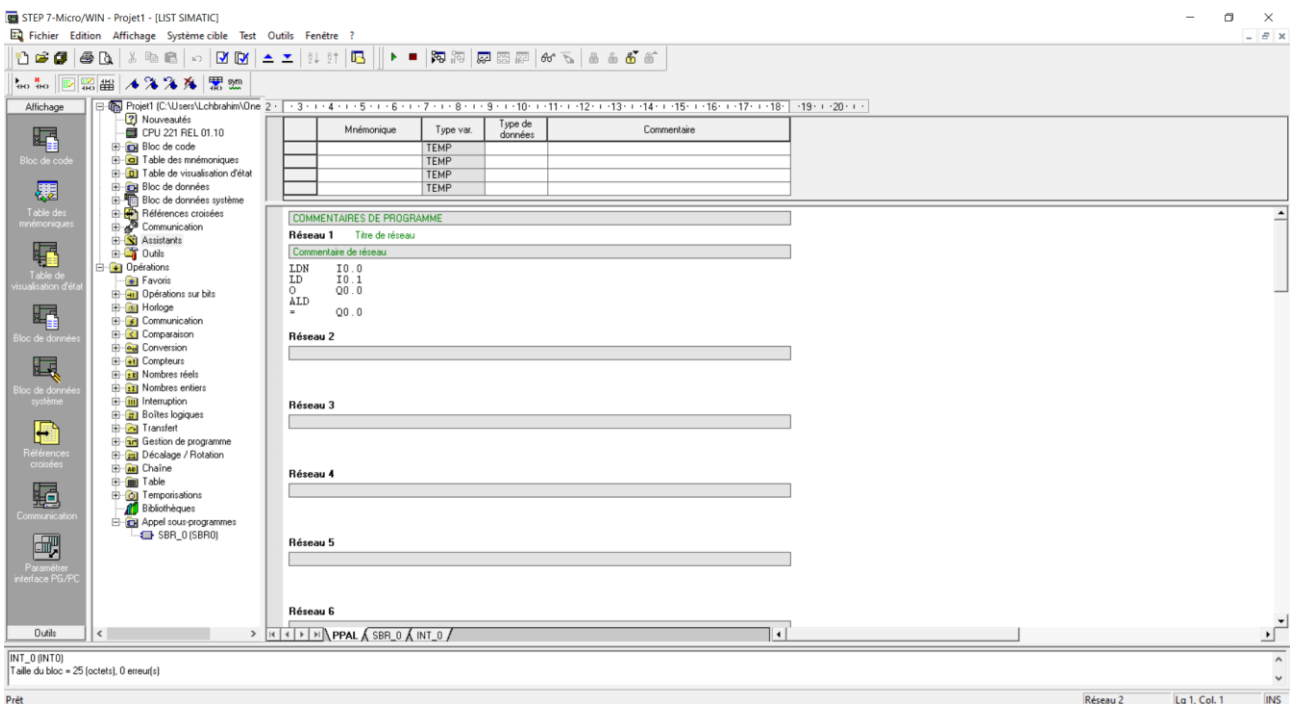
Exemple de programme en CONT/LD :



- **Liste d'instructions (LIST)**

L'éditeur LIST affiche le programme sous forme textuelle. Avec l'éditeur LIST, vous entrez des abréviations d'opérations quand vous créez des programmes de commande. L'éditeur LIST permet également de créer des programmes qu'il n'est pas possible de représenter avec les éditeurs CONT ou LOG. En effet, avec cet éditeur, vous programmez dans un langage natif du S7-200 et non dans un éditeur graphique où des restrictions sont nécessaires pour que les schémas soient tracés correctement. Comme vous le constatez dans l'exemple ci-dessous, la programmation avec ce langage littéral est très similaire à la programmation en langage assembleur.

Exemple de programme en LIST :



- **Logigramme (LOG)**

C'est un langage de programmation graphique qui utilise les boîtes de l'algèbre de Boole pour représenter les opérations logiques. Les fonctions complexes, comme par exemple les fonctions mathématiques, peuvent être représentées directement combinées avec les boîtes logiques.

Le S7-200 lit les entrées physiques au début de chaque cycle et écrit ces valeurs dans la mémoire image des entrées. Vous pouvez accéder à la mémoire image des entrées par bits, octets, mots ou doubles mots.

A la fin du cycle, le S7-200 copie dans les sorties physiques les valeurs contenues dans la mémoire image des sorties. Vous pouvez accéder à la mémoire image des sorties par bits, octets, mots ou doubles mots.

Exemple de programme en LOG/FBD :

The screenshot displays the SIMATIC Manager environment for a STEP 7-Micro/WIN project. The main workspace shows the following elements:

- Variable Declaration Table:**

Mnémonique	Type var.	Type de données	Commentaire
	TEMP		
	TEMP		
	TEMP		
- Network Comments:**
 - COMMENTAIRES DE PROGRAMME**
 - Réseau 1** Titre de réseau
 - Commentaire de réseau
- Logic Diagram (FBD):**

```

graph LR
    I01[I0.1] --- OR[OR]
    Q00[Q0.0] --- OR
    OR --- AND[AND]
    I00[I0.0] --- AND
    AND --- Q00
    
```
- Other Networks:** Réseau 2, Réseau 3, and Réseau 4 are shown as empty horizontal bars.

At the bottom of the window, the status bar indicates: "Active/désactive les signets à l'emplacement en cours." and "Réseau 3 Ligne 1, Col. 1 INS".