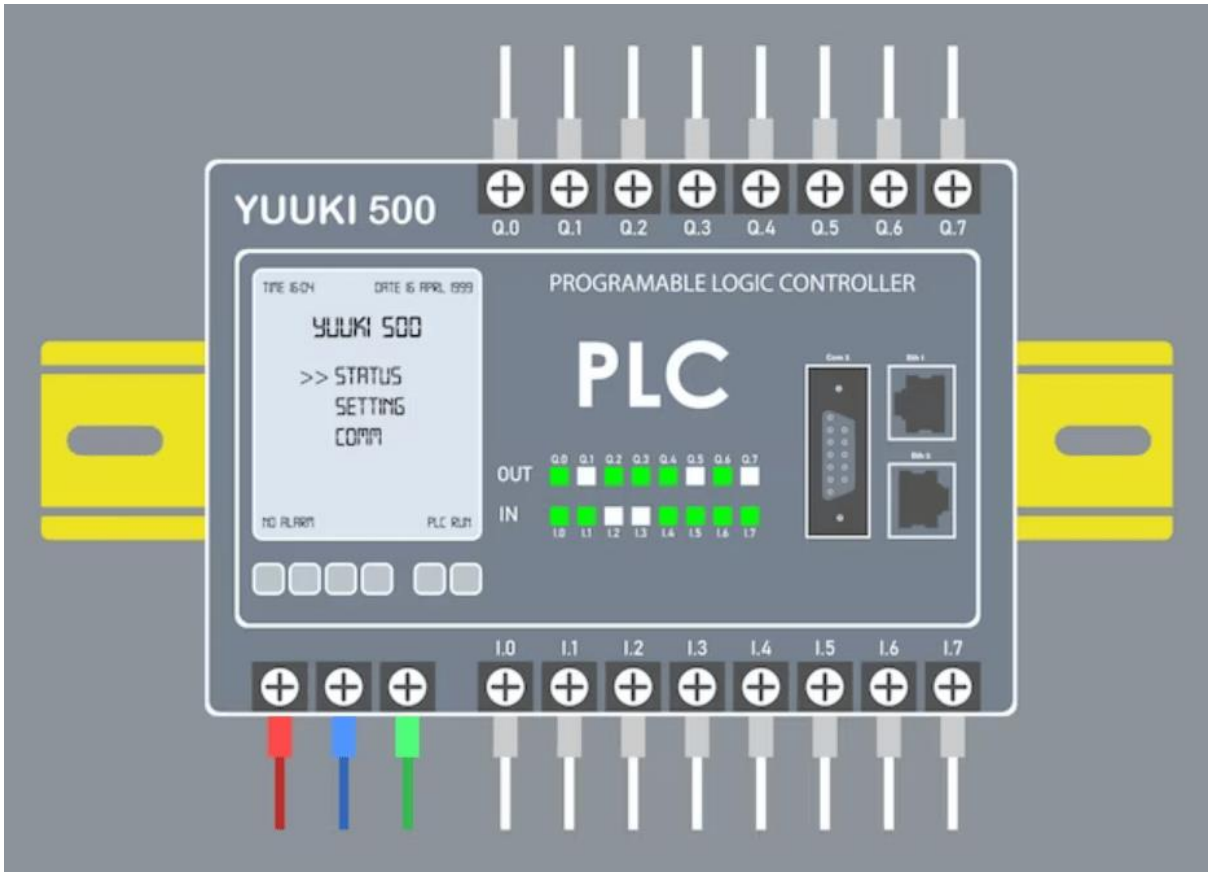
UNIVERSITÉ DES MASCAREIGNES

SAVOIR, C'EST POUVOIR

Université de Limoges

LES AUTOMATES PROGRAMMABLE (PLC)



Departement : **Genie Electronique et informatique Industriel**

Destiné à: Madame Bhamini Sreekeesoon

DATE : 20 Septembre 2024

SOMMAIRE

INTRODUCTION

Les Automates Programmables Industriels, ou PLC (Programmable Logic Controller), sont des dispositifs électroniques utilisés pour le contrôle et l'automatisation des systèmes industriels. Ils jouent un rôle central dans la modernisation de l'industrie en permettant le remplacement des systèmes de relais câblés traditionnels par une solution plus flexible, programmable et fiable. Grâce aux PLC, il est possible de surveiller, contrôler et automatiser des processus complexes dans des environnements variés tels que les lignes de production, les systèmes de transport, et même les centrales électriques Composants de base des PLC. Un PLC est composé de plusieurs éléments clés qui lui permettent de fonctionner de manière optimale. Le cœur du système est l'unité centrale de traitement (CPU), qui exécute les programmes stockés dans la mémoire et traite les signaux reçus des dispositifs d'entrée pour produire des commandes aux dispositifs de sortie. Les modules d'entrée/sortie (I/O) permettent de connecter le PLC aux capteurs et actionneurs externes, assurant ainsi une interaction directe avec l'environnement physique. Ces composants sont essentiels au bon fonctionnement des PLC et garantissent leur efficacité dans les systèmes automatisés. Avantages des PLC Les PLC présentent de nombreux avantages par rapport aux systèmes de contrôle traditionnels. Tout d'abord, ils offrent une grande flexibilité grâce à leur capacité à être reprogrammés facilement, ce qui les rend adaptables à différents types de processus et d'applications. De plus, ils sont compacts, robustes, et plus simples à entretenir, ce qui réduit les coûts d'exploitation et de maintenance. Leur fiabilité, combinée à leur rapidité d'exécution, permet d'améliorer la productivité et d'assurer un contrôle précis des processus Système d'exploitation et Simulation .Le système d'exploitation des PLC est conçu pour gérer les tâches en temps réel, en exécutant les instructions programmées de manière séquentielle tout en surveillant les performances du système. Ce processus est crucial pour assurer une automatisation fiable et efficace. Par ailleurs, la simulation des PLC est un outil puissant qui permet de tester et valider les programmes avant leur implémentation dans des environnements réels. Cela réduit les risques d'erreurs et améliore la sécurité lors de la mise en service.

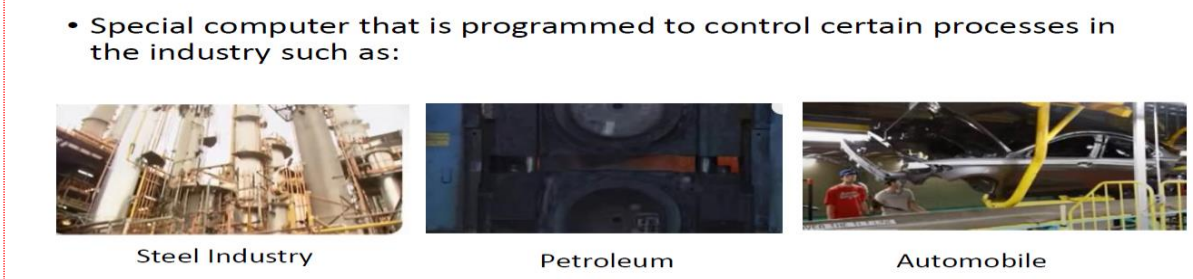
En somme, les PLC sont des piliers de l'automatisation industrielle moderne, offrant des solutions flexibles et évolutives adaptées aux besoins des industries contemporaines.

1 QU’est ce qu’un unités centrale ?

L'unité centrale de traitement (CPU) est le cœur et le cerveau d’un automate programmable industriel (PLC). Son rôle est essentiel, car c'est elle qui assure la gestion de toutes les tâches de contrôle automatisé. En effet, la CPU est responsable de l'exécution des instructions programmées, de la gestion des entrées et sorties, ainsi que de la communication avec les autres composants du système.

1.1 Rôle Fondamental de la CPU dans un PLC

La CPU d'un PLC fonctionne de manière similaire à celle d’un ordinateur, bien que son objectif soit plus spécialisé : elle est conçue pour contrôler des processus industriels en temps réel.



a. Figure montrons : Les diffèrent automatisme dans plusieurs industrialisation

Son rôle est d'exécuter le programme que l'opérateur a défini à l'aide d'un langage de programmation (tel que le Ladder, le FBD ou le ST), en prenant des décisions en fonction des entrées reçues des capteurs et en envoyant des commandes aux actionneurs.

b. Figure montrons : L’exécution de Ladder logic Programming

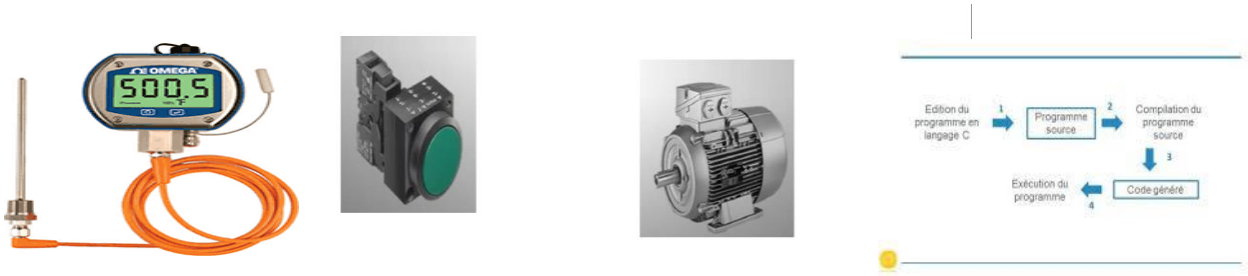
Par exemple, dans une usine, un PLC équipé d’une CPU pourrait contrôler un tapis roulant. Lorsque des capteurs détectent un objet sur le tapis, la CPU traite cette information et décide de démarrer ou d'arrêter le moteur du tapis roulant selon la logique programmée.

1.1.1 Processus d'Exécution de la CPU

Le fonctionnement de la CPU est basé sur un cycle d'exécution en trois étapes principales, appelé "cycle d'interrogation" ou "scan cycle" :

Lecture des entrées* : La CPU commence par lire l'état des dispositifs d'entrée, tels que les capteurs, les boutons ou les interrupteurs. Ces signaux d'entrée sont stockés en mémoire pour être utilisés lors de l'exécution du programme.

Exemple* : Si un capteur de température envoie une valeur à la CPU, cette dernière la stocke en mémoire pour la comparer ensuite avec les seuils programmés.



a. Figure Montrons : Capteur de Températures ;les boutons ou les interrupteurs

1.1.2 Traitement du programme :

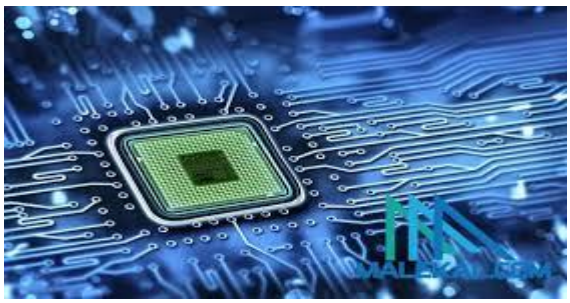
Ensuite, la CPU exécute les instructions du programme dans un ordre séquentiel. Elle utilise les données des entrées pour prendre des décisions et déterminer les sorties appropriées.

- *Exemple* : Si la température détectée est supérieure à la limite programmée, la CPU peut activer une sortie pour allumer un ventilateur ou un système de refroidissement

b. Figure montrons : Capteur de Refroidissements

Mise à jour des sorties* : Enfin, la CPU met à jour les dispositifs de sortie en fonction des décisions prises. Elle commande des éléments comme des moteurs, des lampes ou des vannes, en fonction de la logique programmée.

- *Exemple* : Après avoir traité le programme, la CPU envoie un signal à un actionneur pour ouvrir une valve si la pression dans une canalisation est trop élevée.

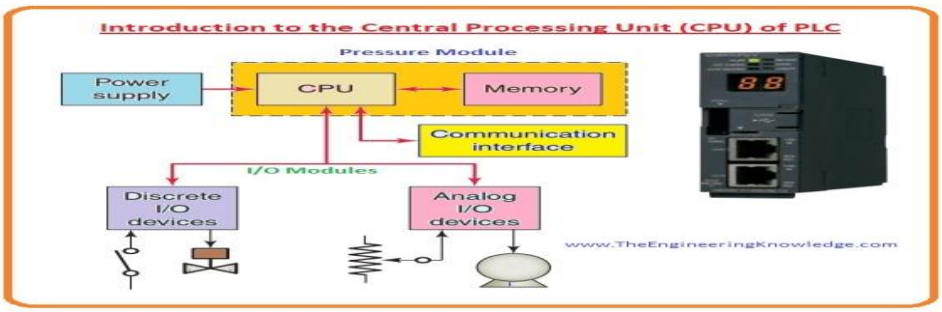


a. **Figure montrons :**TRAITEMENT DES DONNER POUR ENVOYER UN SIGNAL

Ce processus de cycle d'interrogation se répète continuellement à une vitesse extrêmement rapide, permettant au PLC de réagir en temps réel aux changements dans le système. La rapidité de ce cycle est un aspect critique de la performance de la CPU, car elle garantit que le PLC peut surveiller et contrôler des processus complexes sans délai.

1.1.3 Importance de la CPU dans un PLC

La CPU est l'élément central qui détermine la performance et l'efficacité d'un PLC. Plusieurs aspects mettent en lumière son importance ..



b. **Figure montrons :Le PROCESSING FOR PLC**

1.1.4 Unite centrale de Traitement (CPU)

L’unité centrale de traitement (UCT) appelée aussi processeur central (Central Processing

Unit = CPU) est l’élément moteur de l’ordinateur qui interprète et exécute les instructions

du programme. Il est intimement associé à la mémoire centrale ou sont stockées ces instructions avec leurs données à traiter elle est composée de deux unités fonctionnelles.

- L’unité arithmétique et logique (UAL) qui effectue les opérations arithmétiques et logiques.
- L’unité de contrôle et de commande (UCC) qui commande l’exécution de toutes les opérations à tous les niveaux (UAL, MC, E/S) ainsi que le contrôle de leur déroulement

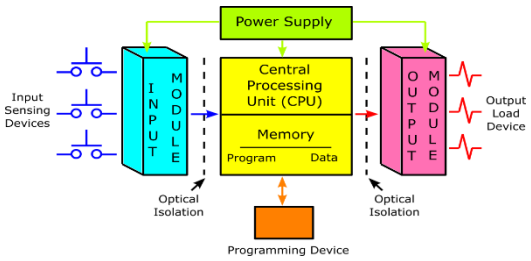
2 A QUOI SERT UN PLC ?

En termes électriques, PLC est l'acronyme de Programmable Logic Controller. Il s'agit d'un ordinateur compact et spécial conçu pour effectuer des opérations logiques pour commander différents systèmes électromécaniques. Initialement, les PLC ont été développés pour effectuer les fonctionnalités des relais câblés dans l'industrie automobile.

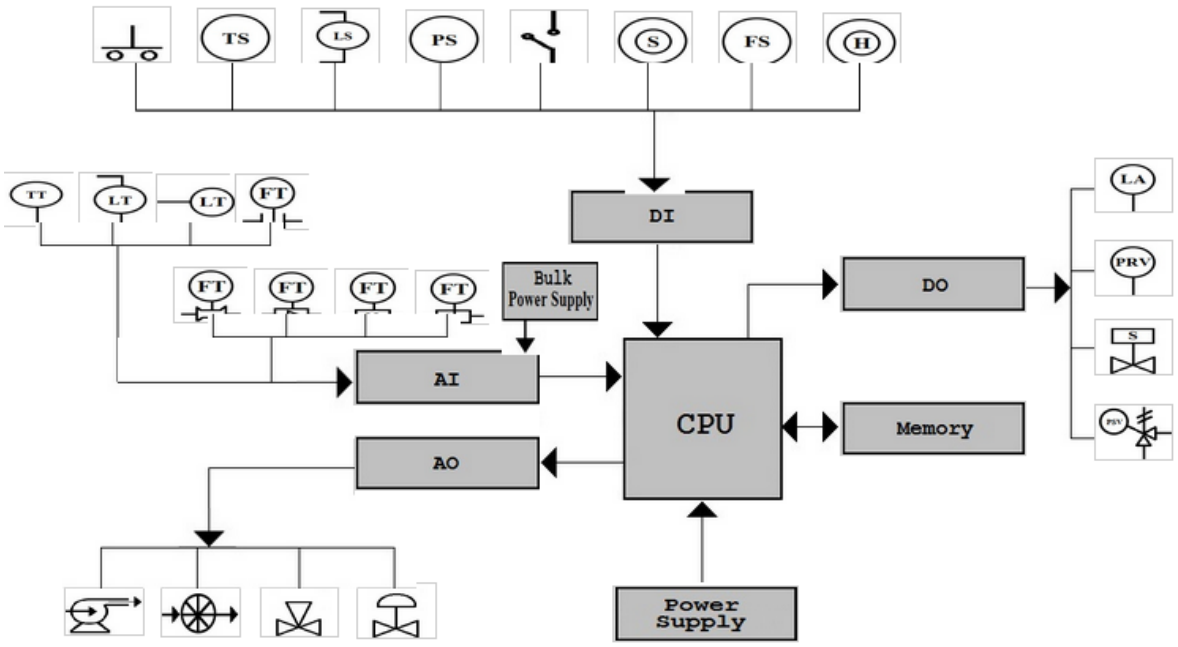
Aujourd'hui, les contrôleurs logiques programmables (PLC) sont largement utilisés pour automatiser les processus de fabrication, les chaînes de montage et différents types de machines industrielles qui nécessitent une programmation facile, des niveaux élevés de fiabilité et un diagnostic avancé des défauts. Ils sont construits comme des systèmes de contrôle robustes qui peuvent fonctionner de manière fiable dans des environnements industriels extrêmes.

2.1 Composants d'un système PLC

Les éléments de base d'un système PLC typique sont les suivants:



c. **Figure montrons :** Éléments d'un système PLC



d. Figure montrons : en détails le type de données d'entrées que le CPU recoïte et la manière dont il les traite pour les transmettre a ses sorties.

2.2 Avantages des PLC, Fonctionnement et Objectif

Les automates programmables industriels (PLC) sont des systèmes de contrôle extrêmement polyvalents et essentiels à l'automatisation des processus dans de nombreux secteurs industriels. Leur conception repose sur des technologies robustes et flexibles qui leur permettent de remplacer les systèmes de commande traditionnels, souvent basés sur des relais électromécaniques. Voici une vue d'ensemble de leurs avantages, de leur fonctionnement global et du rôle de la CPU.

2.2.1 Avantages des PLC

Les PLC offrent une multitude d'avantages dans les environnements industriels modernes, qui incluent :

- **Fiabilité** : Conçus pour des environnements industriels difficiles, les PLC peuvent résister à des conditions extrêmes telles que des températures élevées, de fortes vibrations ou des environnements poussiéreux.

- Flexibilité : Un des plus grands avantages des PLC est leur capacité à être reprogrammés facilement. Contrairement aux systèmes à relais câblés, la logique d'un PLC peut être changée rapidement en modifiant le programme sans changer le câblage physique.

- Maintenance réduite : Les PLC nécessitent peu de maintenance, car ils n'ont pas de pièces mobiles comme les relais électromécaniques, ce qui réduit les pannes dues à l'usure.

- **Rapidité et efficacité**: Les PLC traitent les informations en temps réel avec des temps de réponse très rapides, ce qui les rend particulièrement efficaces dans les systèmes automatisés où la précision et la rapidité sont essentielles (par exemple, les chaînes de production).

- Extensibilité : De nouveaux modules d'entrées/sorties ou des fonctionnalités supplémentaires peuvent être facilement ajoutés à un PLC existant, ce qui permet de suivre l'évolution des besoins industriels.

2.2.2 Objectif des PLC

L'objectif principal d'un PLC est de surveiller et contrôler des machines ou des processus automatisés. Les PLC reçoivent des données en temps réel des capteurs et autres dispositifs d'entrée, exécutent des instructions logiques programmées (généralement sous forme de programmes Ladder ou texte structuré), puis commandent des actionneurs, des moteurs, des vannes ou d'autres dispositifs de sortie.

Exemple d'objectif dans l'industrie :

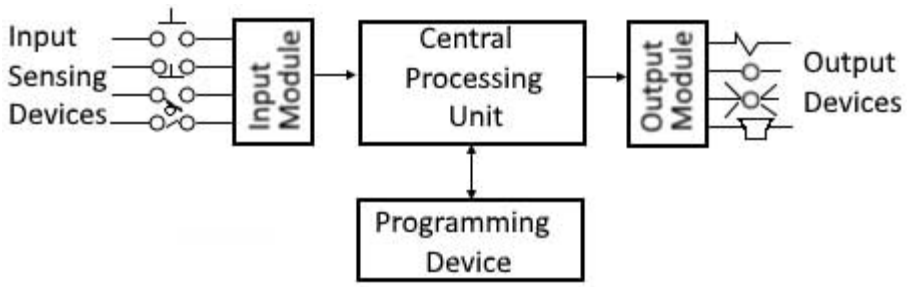
- Automatisation d'une chaîne de production: Le PLC contrôle le mouvement des produits sur la chaîne, vérifie leur qualité avec des capteurs, et ajuste la vitesse ou la direction en fonction des besoins du processus.

e. **Figure montrons :** Equipements d'automatisation

2.2.3 Mode de Fonctionnement d'un PLC

Le mode de fonctionnement d'un PLC repose sur un cycle d'exécution en temps réel. Ce cycle, appelé "cycle d'interrogation" (scan cycle), comprend plusieurs étapes répétées à grande vitesse.

a. **Lecture des entrées :** Le PLC commence par lire l'état des capteurs, boutons, interrupteurs, ou autres dispositifs d'entrée connectés. Ces entrées peuvent inclure des signaux numériques (comme un interrupteur ouvert ou fermé) ou analogiques (comme un capteur de température mesurant une valeur variable).



f. Figure montrons : Schéma fonctionnel du module PLC

b. **Exécution du programme** :

Une fois les données d'entrée collectées, le PLC exécute le programme de contrôle qui a été programmé dans la CPU. Ce programme détermine les actions à entreprendre en fonction des conditions d'entrée. Par exemple, si un capteur détecte une température trop élevée dans un système de chauffage, le programme peut décider de couper l'alimentation du chauffage.

c. **Mise à jour des sorties:**

Après avoir traité les instructions programmées, le PLC met à jour les dispositifs de sortie (comme des moteurs, des vannes ou des alarmes) en fonction des résultats du traitement. Cela peut inclure l'ouverture d'une vanne, l'activation d'un moteur ou le déclenchement d'une alarme.

d. **Communication avec d'autres systèmes**:

Certains PLC intègrent des fonctionnalités de communication pour interagir avec d'autres dispositifs, tels que des ordinateurs, des réseaux SCADA ou d'autres PLC. Cela permet d'échanger des données pour un contrôle coordonné et centralisé.

Ce cycle d'exécution est extrêmement rapide, se déroulant plusieurs fois par seconde, garantissant ainsi que le PLC réagit rapidement à tout changement dans le système.

e. **Communication avec d'autres systèmes :** Certains PLC intègrent des fonctionnalités de communication pour interagir avec d'autres dispositifs, tels que des ordinateurs, des réseaux SCADA ou d'autres PLC. Cela permet d'échanger des données pour un contrôle coordonné et centralisé.

Ce cycle d'exécution est extrêmement rapide, se déroulant plusieurs fois par seconde, garantissant ainsi que le PLC réagit rapidement à tout changement dans le système.

f. Le Rôle de la CPU dans le Fonctionnement du PLC

La CPU (unité centrale de traitement) est l'élément clé du fonctionnement d'un PLC. Elle gère l'exécution du programme, le traitement des données des entrées et des sorties, et la communication avec les autres composants. Le rôle de la CPU peut être divisé en plusieurs tâches :

- Traitement des données : La CPU lit en continu les données des dispositifs d'entrée (capteurs, boutons) et les compare aux conditions définies dans le programme. Cela permet au PLC de "comprendre" l'état actuel du système à tout moment.

- Exécution des instructions* : Le programme stocké dans la mémoire de la CPU est exécuté étape par étape. Chaque instruction, basée sur une logique prédéfinie (par exemple, des portes logiques OR, NAND, NOR), est traitée pour prendre des décisions sur les actions à entreprendre.

- Gestion des sorties: Après avoir traité les entrées et exécuté le programme, la CPU détermine quels dispositifs de sortie doivent être activés ou désactivés. Par exemple, elle peut commander l'ouverture d'une porte automatique si un capteur de proximité détecte une personne.

Exemple d'utilisation :

Dans un système de gestion de température*, la CPU du PLC surveille en permanence la température à l'aide de capteurs. Si la température dépasse un seuil défini, la CPU exécute des instructions qui activent des ventilateurs ou des systèmes de refroidissement.

3 Logic Gates (Portes Logiques)

Les *portes logiques* sont des éléments fondamentaux dans les systèmes PLC. Elles permettent d'effectuer des opérations logiques, qui déterminent le comportement des sorties en fonction des entrées. Parmi les portes logiques les plus couramment utilisées dans les PLC, on trouve :

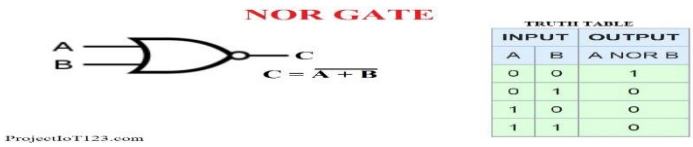
- *Porte OR* : La sortie est active (1) si au moins une des entrées est active (1). Cela permet de déclencher une action lorsqu'une ou plusieurs conditions sont remplies.

- Exemple : Un moteur démarre si l'un des deux boutons de commande est appuyé.

g.Figure Montrons : Une porte d'Entrée a deux Entrée

- Porte NOR* : La sortie est active (1) uniquement si toutes les entrées sont inactives (0). Elle est utilisée pour assurer que l'action se produise seulement en l'absence d'entrée active.

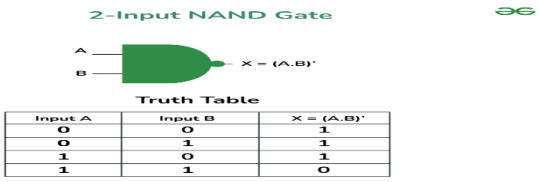
- *Exemple* : Un système de sécurité se déclenche si aucun des capteurs n'est activé.



g.Figure Montrons : La porte de NOR

- Porte NAND* : La sortie est active (1) sauf si toutes les entrées sont actives (1). Cette porte est souvent utilisée dans les systèmes où la sécurité est primordiale.

- *Exemple* : Un arrêt d'urgence est déclenché si un certain seuil est atteint par plusieurs capteurs



h.Figure montrons : NAND Gate Truth Table

Ces portes logiques permettent de structurer les décisions prises par le PLC, et leur utilisation dans le programme détermine les actions en fonction des états des capteurs ou des boutons.

a. ▪ Ladder Diagram (Diagramme Échelle)

Le *Ladder Diagram*, ou diagramme échelle, est un langage de programmation graphique largement utilisé pour les PLC. Il est basé sur les anciens schémas de relais électromécaniques et représente la logique du programme sous forme d'une échelle avec des "rangs" ou "barreaux".

- **Barres verticales** :

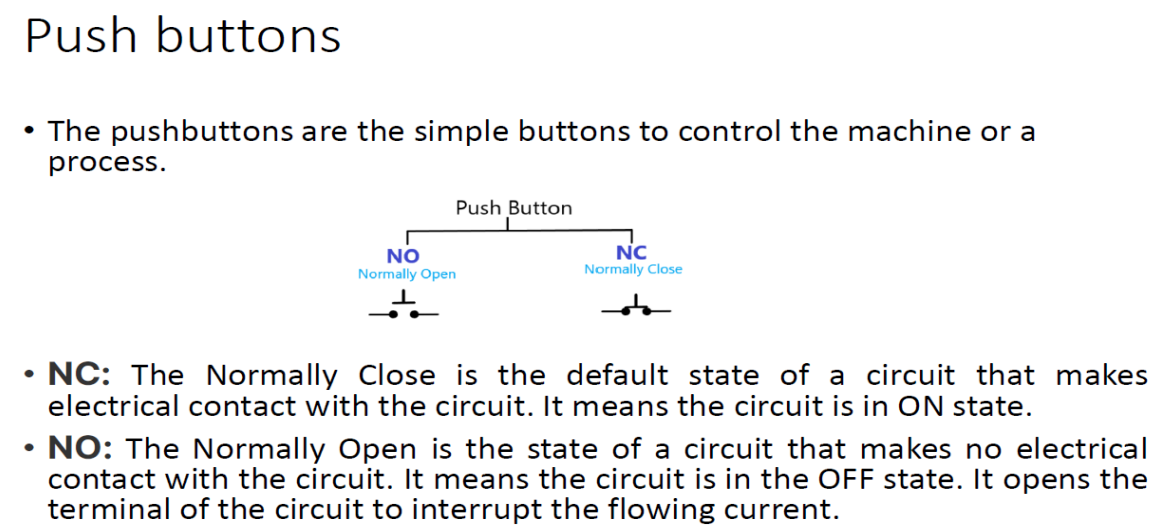
- Ces deux lignes verticales représentent les bornes d'alimentation d'un circuit, une pour l'alimentation positive (ou phase) et l'autre pour le neutre ou la masse.

- Tous les composants logiques (comme les contacts ou les bobines) sont placés entre ces deux lignes pour simuler le flux du courant dans un circuit logique.

- **Rungs (Échelons)** :

- Les rungs sont les lignes horizontales qui représentent les circuits de commande. Chaque rung est une "ligne logique" qui exprime une condition pour activer ou désactiver une sortie. Cela fonctionne comme une déclaration "SI...ALORS" dans la programmation.

- Un rung peut contenir des contacts normalement ouverts (NO), normalement fermés (NC), des temporisateurs, des compteurs, des bobines, et d'autres composants.



3.1 Composants dans un Ladder Diagram :

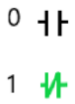
- Contact normalement ouvert (NO) :
- Ce contact est ouvert lorsqu'il n'y a pas de courant. Il se ferme quand une condition est vraie, permettant le passage du courant. Cela est représenté graphiquement par un symbole ressemblant à deux barres verticales parallèles (| |).
- Ex: Si un bouton est pressé, un contact NO peut se fermer pour activer une bobine.

- Contact normalement fermé (NC) :
- Ce contact est fermé lorsqu'il n'y a pas de courant et s'ouvre lorsque la condition devient vraie. Graphiquement, cela est représenté par un symbole ressemblant à deux barres avec une diagonale (|/|).
- Ex: Lorsque l'arrêt d'un moteur est demandé, un contact NC s'ouvre pour couper l'alimentation du moteur.

NO

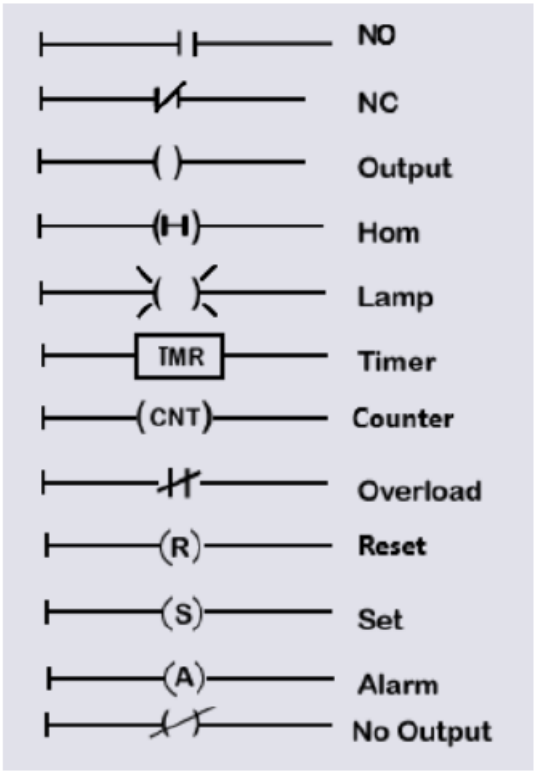
- The NO button turns NC when it is pressed. It means, when the input is 1, NO turns NC. It means the current can pass through.

- Consider the below image:

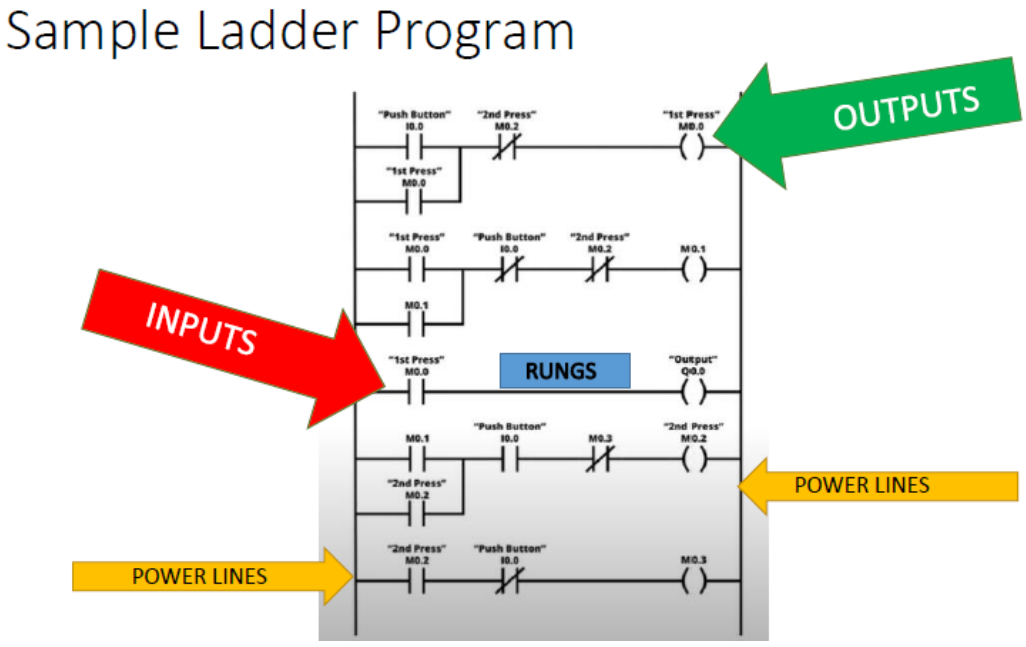


It clearly shows that when the input is 0, NO remains NO. It turns NC when the input is 1.

a. Symbols utiliser dans PLC pour les Programmation



b. Exemple de Programme à contacts :



- **Bobine (Coil) :**

- Une bobine est un élément de sortie qui représente un relais ou un actionneur à activer ou désactiver. Graphiquement, elle est représentée par un symbole en forme de parenthèses (-( )-). Si les conditions précédentes dans le rung sont satisfaites, la bobine s'active (ou se désactive).

- Ex: Une bobine peut représenter l'activation d'un moteur ou d'une lampe.

- **Temporisateurs (Timers) :**

- Un temporisateur introduit un délai avant qu'une action ne se produise. Il peut être utilisé pour retarder l'activation ou la désactivation d'une sortie.

- Ex: Un temporisateur peut attendre 5 secondes avant d'activer une bobine.

- **Compteurs (Counters):**

- Un compteur compte les événements (comme les impulsions) et déclenche une sortie après qu'un nombre prédéfini d'événements se soit produit.

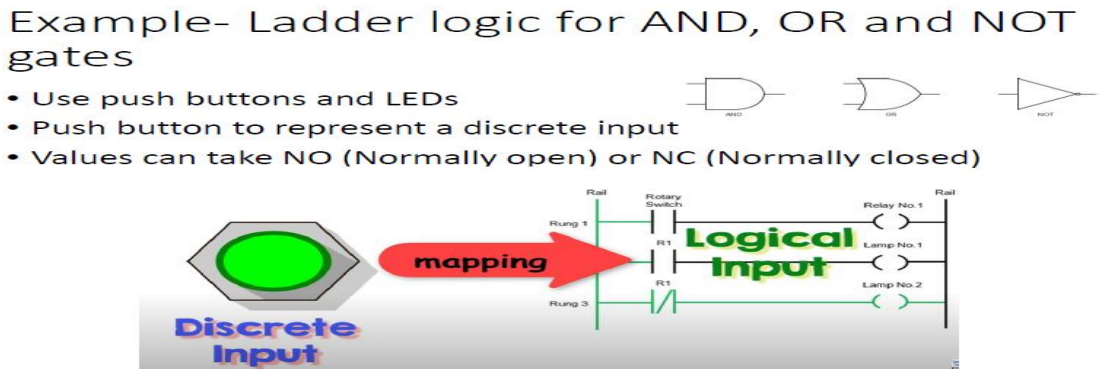
- Ex: Après 10 pressions d'un bouton, un compteur peut activer une bobine.

b. Logique combinatoire :

- Les contacts (NO et NC) peuvent être combinés en série ou en parallèle pour créer une logique complexe.

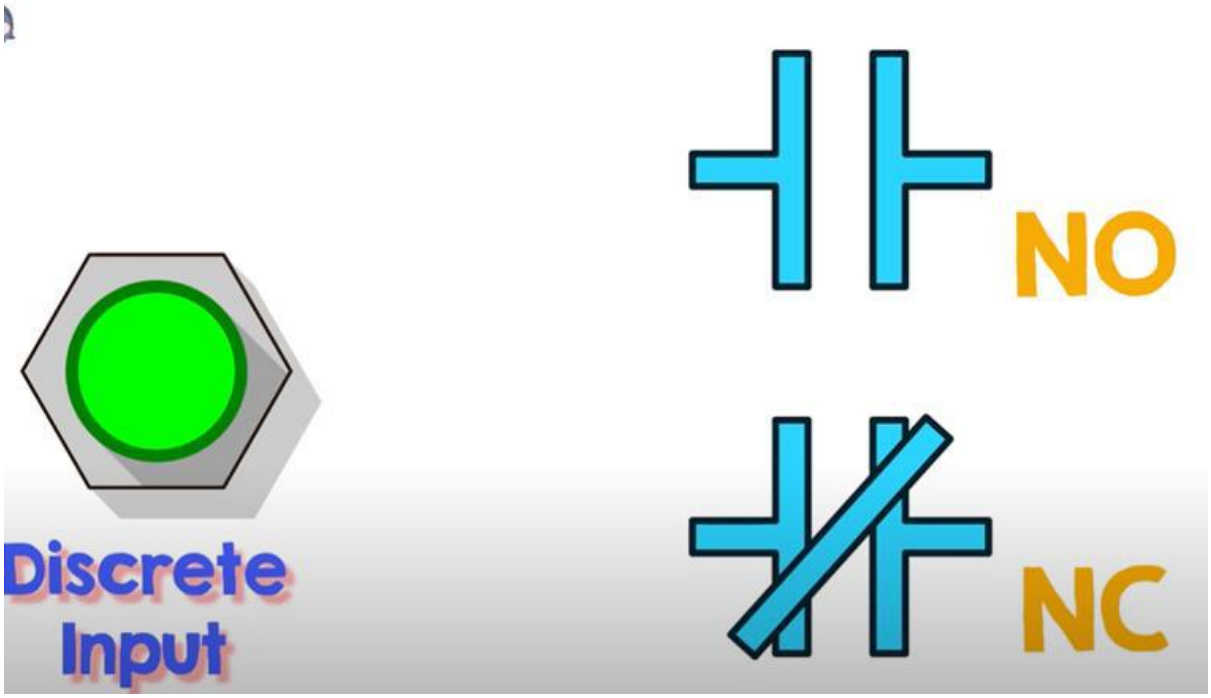
- En série* : Si plusieurs contacts sont placés en série dans un rung, alors tous doivent être fermés pour que le courant passe. Cela fonctionne comme une porte logique AND.

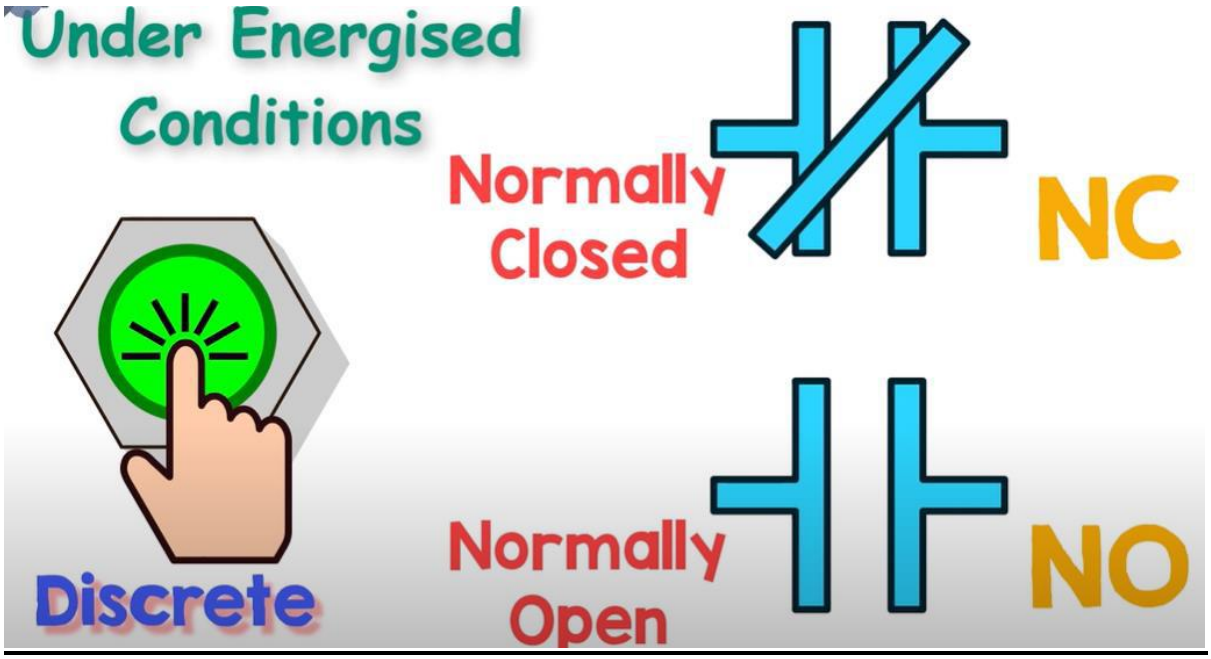
- En parallèle* : Si plusieurs contacts sont placés en parallèle, alors le courant peut passer si au moins un contact est fermé. Cela fonctionne comme une porte logique OR.



3.2 . Conditions :

- Le moteur doit démarrer quand le bouton de démarrage est pressé (contact NO).
- Le moteur doit s'arrêter quand le bouton d'arrêt est pressé (contact NC).
- Une lampe doit s'allumer quand le moteur fonctionne.





a. Ladder Diagram :

- Rung 1 : Le bouton de démarrage (NO) est en série avec un bouton d'arrêt (NC) et une bobine du moteur. Si le bouton de démarrage est pressé et que le bouton d'arrêt est relâché, la bobine du moteur s'active.

- Rung 2: La lampe est activée par une bobine en série avec la bobine du moteur. Donc, quand le moteur est activé, la lampe s'allume également.

Graphiquement :

- (| | -- |/| -- ( )) → pour le moteur.
- (| | -- ( )) → pour la lampe.

Avantages du Ladder Diagram

b. Facilité de lecture et d'utilisation :

- Le Ladder Diagram est visuellement simple et facile à comprendre pour ceux ayant une expérience dans les circuits électriques traditionnels.

c. Débogage aisé :

- En raison de sa nature graphique, il est facile d'identifier où une condition peut échouer et pourquoi une sortie n'est pas activée.

d. Similitude avec les circuits électriques :

- Les schémas Ladder s'inspirent directement des anciens schémas de relais câblés utilisés dans les systèmes électriques, ce qui facilite la transition vers les systèmes programmés.

e. Programmation directe pour les PLC :

- Le Ladder Diagram est directement utilisable sur la plupart des API/PLC modernes, sans besoin de beaucoup de formation supplémentaire pour les électriciens.

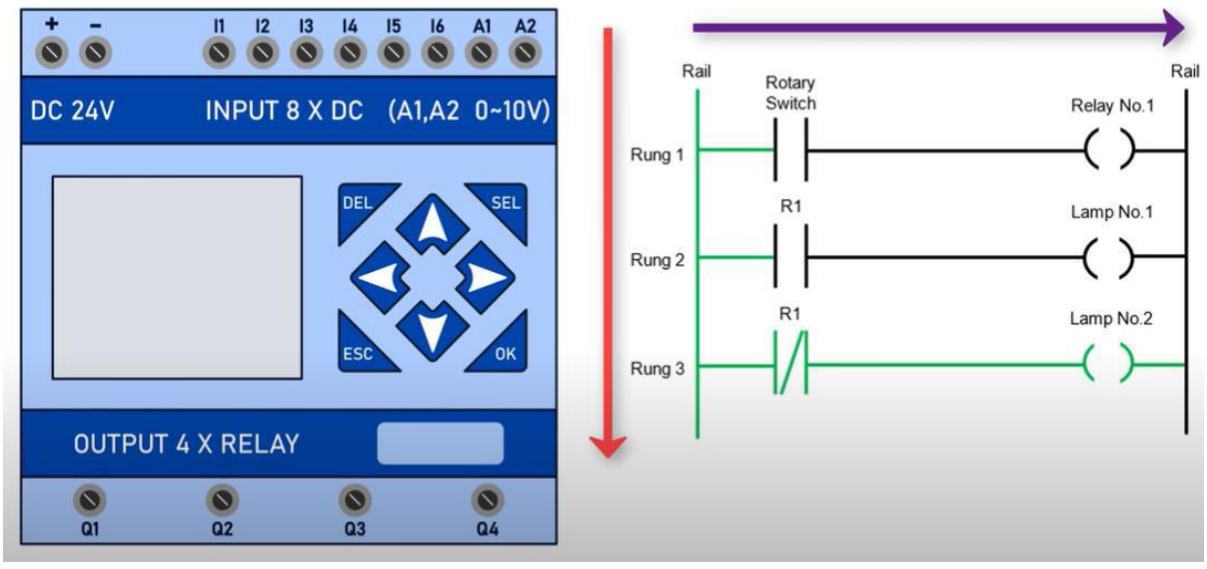


Figure montrons : Traitement des code sur PLC de Gauche a droite et de haut en bas

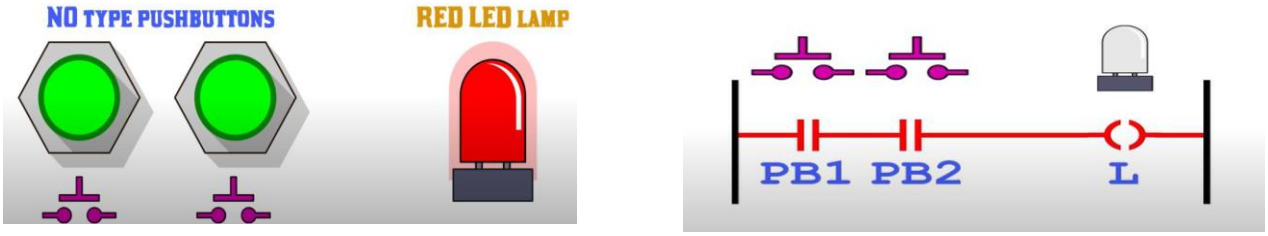
Exemple 1 : Echelle PLC pour logique AND

Utilisez 2 boutons poussoirs et une LED

Les entrées sont connectées sur le cote gauche et peuvent être en série ou en parallèles

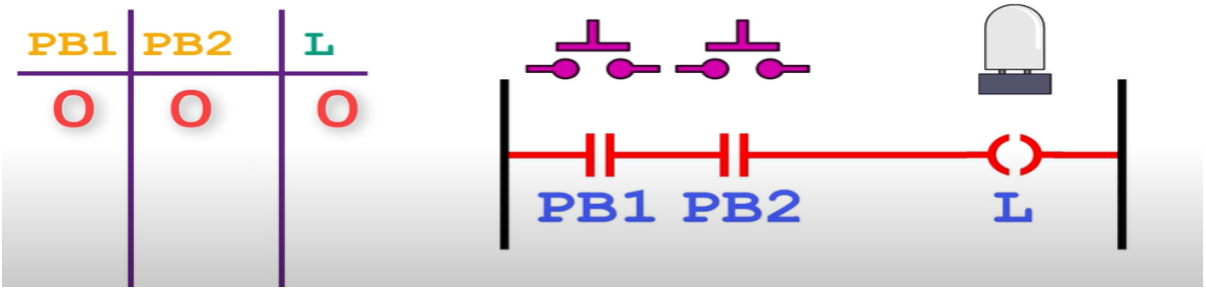
Les sorties ne peuvent être connectées en parallèles que sur le cote droit

Le courant circule de gauche a droite.



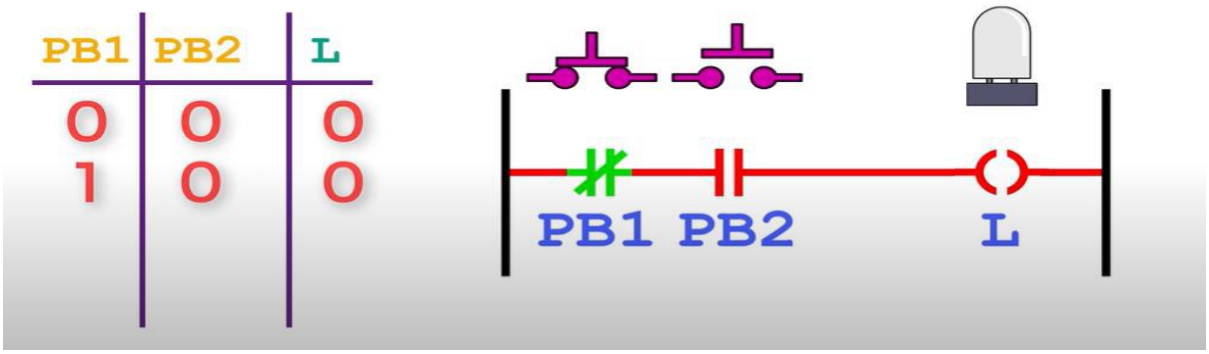
g. **Figure montrons** : L'expériences

AND gate logic :

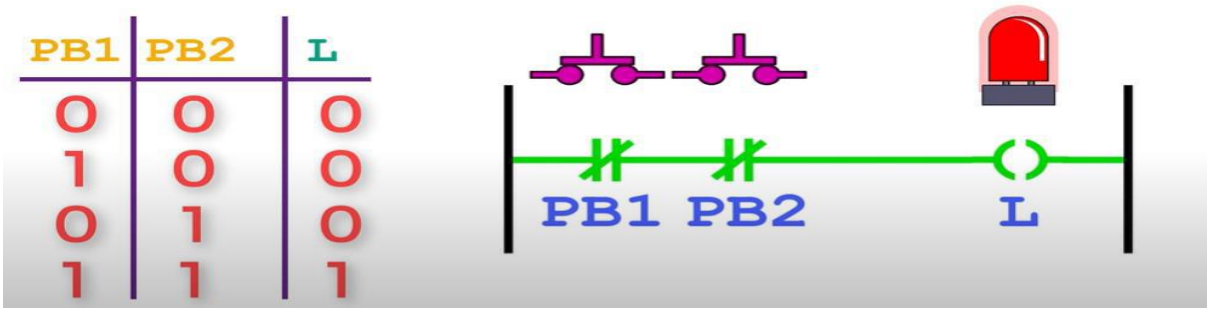


h. **Figure montron** : Les deux interrupteurs son tous ouvert alors le courant est bloquée par conséquences il n'y a pas de passage de courant d'ou la lampe ne s'allume pas.

AND gate logic :



i. **Figure montrons**: L'interrupteurs **PB1** est fermées ce qui veut dire le courant est passant, mais pour l'interrupteur **PB2** l'interrupteur est ouvert cella signifie que le courant est bloquée,par conséquence le courant de l'interrupteurs PB1 ne passe pas donc la lamps reste éteint .



j. **Figure montrons:** L'interrupteurs **PB1** et **PB2** sont tous les deux fermées cella signifie que le courant est passant par conséquences la lampes est Allumer.

4 Le sujet examiné : un système de transport

Dans le système étudié ici, les bouteilles sont transportées depuis une première machine jusqu'à la seconde. L'objectif est de garantir un transfert fluide de ces bouteilles, en toute sécurité, tout en garantissant une coordination optimale entre les deux machines, comme illustré dans la figure ci-dessous :

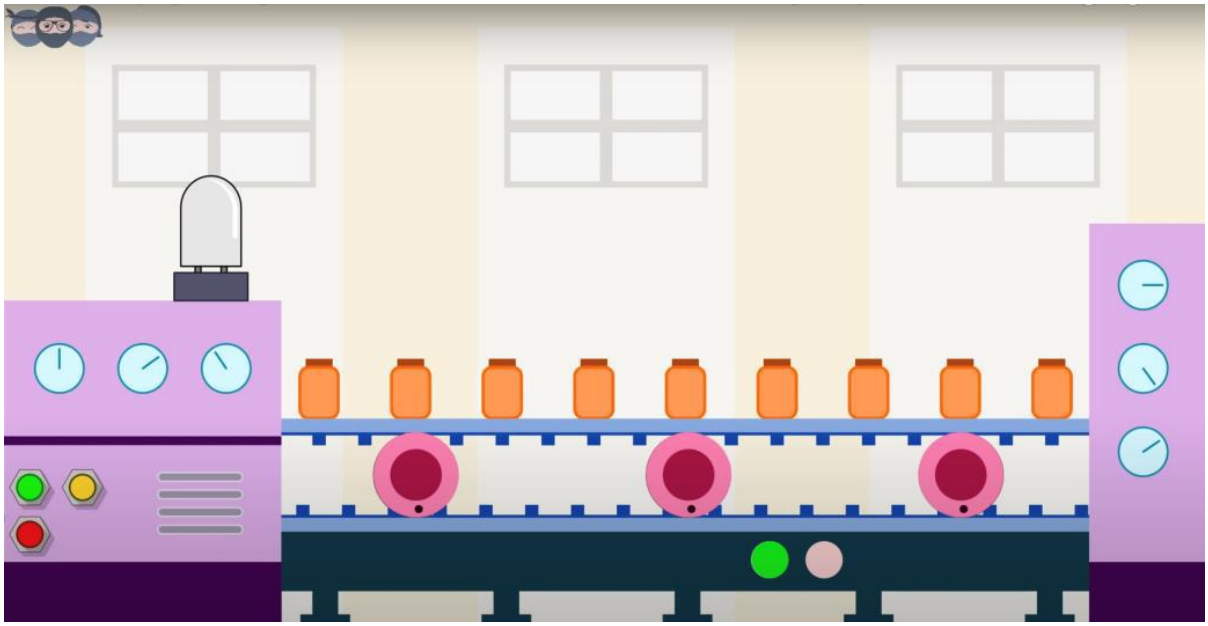


Figure 1 : Système convoyeur de bouteilles

L'objectif de cette automatisation est de permettre au système de déclencher une alarme et de mettre fin à la bande transporteuse en cas de problème ou d'arrêt inattendu de l'une des machines.

Afin d'accomplir cela, j'ai employé « Virtual Labs », qui offre la possibilité d'analyser des portes logiques en utilisant le contrôleur logique programmable de l'outil de simulation en ligne, qui s'affiche comme suit :

Figure 2.

Figure 2 : Interface du simulateur Virtual Lab

4.1 Déclencher l'Alarme

La première difficulté consistait à mettre en marche la sonnette d'alarme en cas de problème ou d'arrêt inattendu de l'une des deux machines.

Dans ce système, nous avons deux machines, ce qui nous permet d'avoir deux entrées qui nous permettent d'obtenir en temps réel l'état des deux machines (Marche/Arrêt). Ainsi, il existe deux options de sortie : l'une simule l'alarme et l'autre affiche l'état de fonctionnement des moteurs des machines. L'alarme doit être activée si l'une des machines s'arrête.

Il est possible de représenter cela à l'aide d'une logique booléenne OU : l'état d'entrée sera l'état de sortie. Afin de simuler une porte logique OU, les deux entrées sont reliées en parallèle sur un seul rangé, avec les contacts des entrées étant des contacts NO (Normally Open), et la sortie représentera l'alarme.

La représentation graphique suivante

Figure 3 : Simulation de la première problématique

Pour que le système soit fonctionnel, il faut tout d'abord compiler le programme à l'aide du bouton « Compile », bien sûr les noms des entrées et de la sortie doit être renseigné avant compilation du programme.
Et afin de tester la logique il est nécessaire d'exécuter le programme en cliquant sur le bouton « Run », ce qui change en vert les deux côtés si tout est en ordre comme visualisé ci-dessous :

Figure 4 : Simulation de la première problématique après exécution

Je rappelle que l’alarme ne s’actionne que lorsque l’une des machines est en arrêt.

Pour tester, que ça soit le cas, ci-dessous sont présenté différents tests :

‘Figure 5 : tests du bon fonctionnement du programme

Comme on peut le constater, les contacts NO deviennent NC (Normally Closed) même si une entrée est activée, ce qui entraîne l'activation de l'alarme. Dans les deux premières illustrations, l'une des machines est en panne, tandis que dans la dernière, les deux machines sont en panne, dans tous ces cas, l'alarme est activée.

La situation est donc résolue, car en cas d'arrêt d'une ou plusieurs machines, l'alarme est activée sans interrompre le convoyeur.

2.Stopper le convoyeur 2

Dès maintenant, je vais résoudre la seconde problématique qui consiste à interrompre le système de transport lorsque l'alarme s'active en cas d'arrêt ou de dysfonctionnement d'une ou plusieurs machines.
Afin d'accomplir cela, je reprends le programme déjà utilisé et ajoute une entrée représentant l'alarme et une sortie représentant le moteur du convoyeur. Cette fois-ci, j'utilise une porte logique non en entrée, avec un contact NC. La figure ci-après illustre ce système :

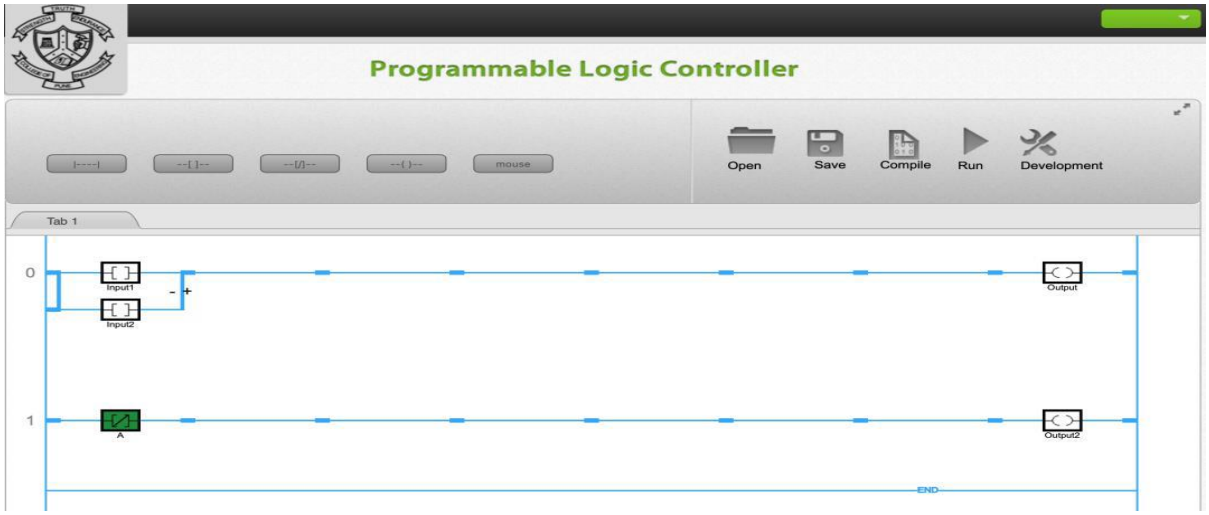


Figure 6 : Programme de déclenchement d'alarme et d'arrêt du convoyeur

Je vais dorénavant tester ce programme en n'oubliant pas de le compiler avant de l'exécuter :

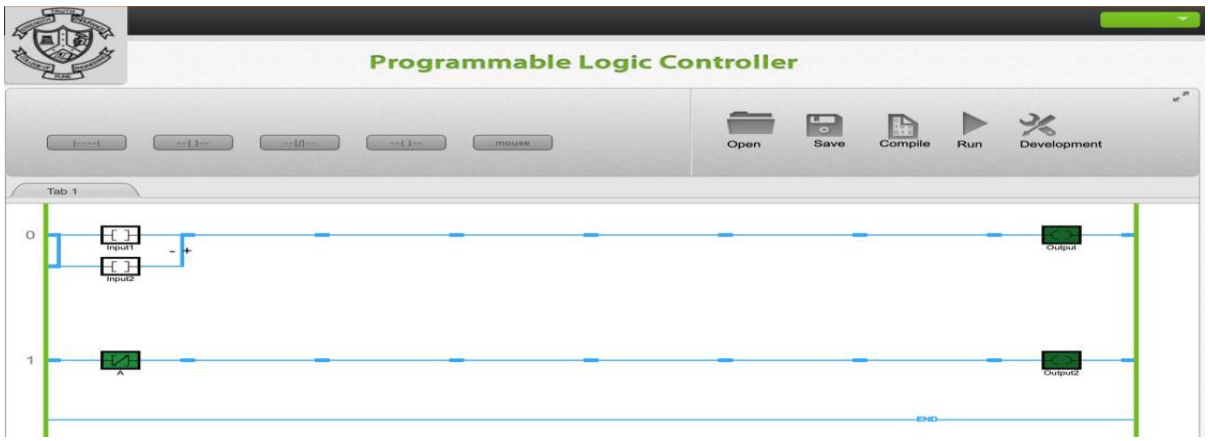


Figure 7 : Programme de déclenchement d'alarme et d'arrêt du convoyeur après exécution

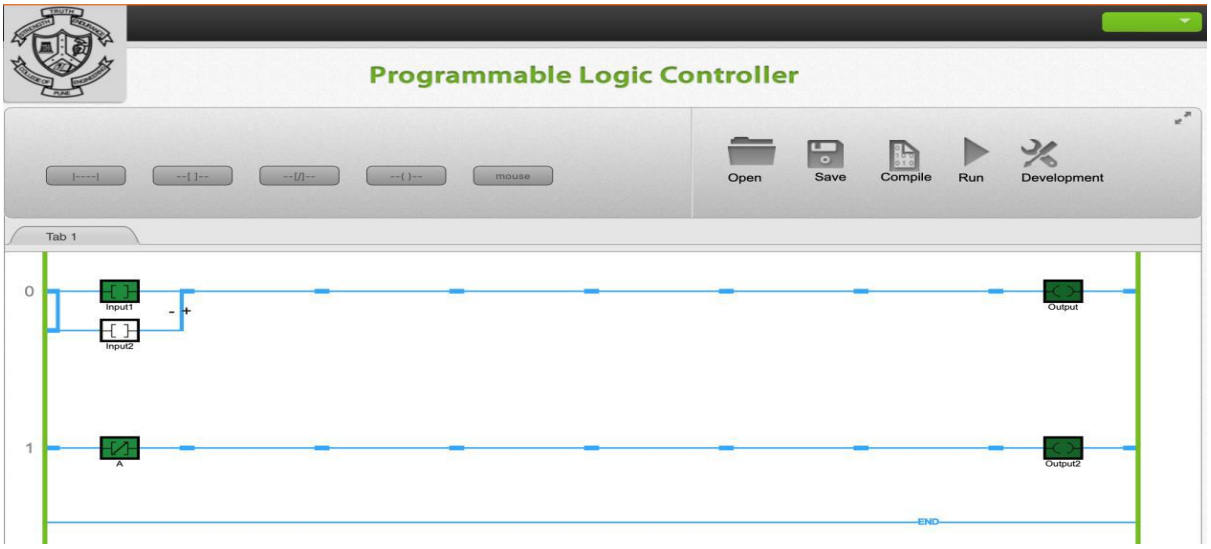


Figure 8 : Test du programme de déclenchement d'alarme et d'arrêt du convoyeur

Comme il est aisé de voir, lorsque l'alarme est actionnée, le contact NC se transforme en contact NO et coupe l'alimentation du moteur, arrêtant ainsi le système de convoyage.
La problématique est ainsi résolue.

En conclusion.

En résumé, le programme élaboré satisfait de manière efficace les exigences du sujet en déclenchant automatiquement l'alarme en cas d'arrêt ou de dysfonctionnement de l'une des

deux machines, et en arrêtant le convoyeur pour éviter toute conséquence de ce dysfonctionnement.

De plus, l'emploi des portes logiques, en particulier les portes OU et NON, a été extrêmement bénéfique pour gérer les conditions d'alarme et d'arrêt du convoyeur. Cela m'a fait réaliser que l'utilisation de logiques simples permet de traduire des situations aussi complexes qu'elles soient.

Grâce à ce programme, j'ai pu acquérir une compréhension approfondie des fondements de l'automatisation industrielle. Grâce à cette expérience, j'ai pu développer des compétences techniques fondamentales en automatisation, tout en améliorant ma compréhension.