

## SEMAINE 3

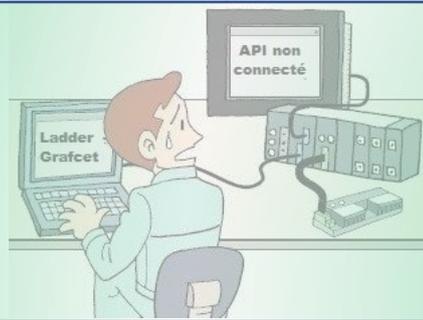
### LES AUTOMATES PROGRAMMABLES INDUSTRIELS

#### FICHE 10 : CONSTITUTION ET FONCTIONNEMENT DES AUTOMATES PROGRAMMABLES

SIEMENS LOGO	CROUZET MILLENIUM	SCHNEIDER ZELIO	SCHNEIDER TWIDO	MOELLER PS4
				

SIEMENS S7-300	SCHNEIDER TSX 37	MOELLER	SCHNEIDER TSX 57
			

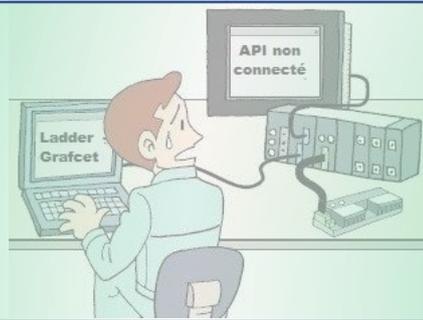
# Automation & Sense



### **Objectifs :**

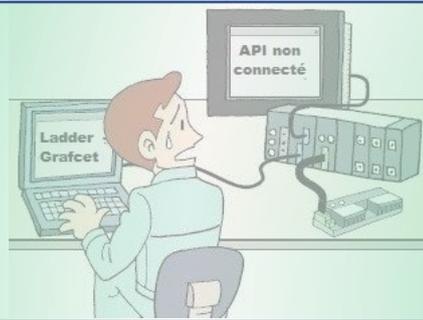
**Cette fiche a pour but de vous initier aux automates programmables industriels**

**Après l'étude de cette fiche vous pourrez identifier les différents constituants d'un automate et comprendre leur fonctionnement**



## SOMMAIRE

- I) Généralités
- II) Différence entre automate et PC
- III) Les types d'automates et leurs critères de choix
- IV) Constitution et principe de fonctionnement d'un automate
- V) Exemple de cahier des charges
- VI) Avantage d'un automate face aux systèmes automatisés à base de relais



## I) Généralités

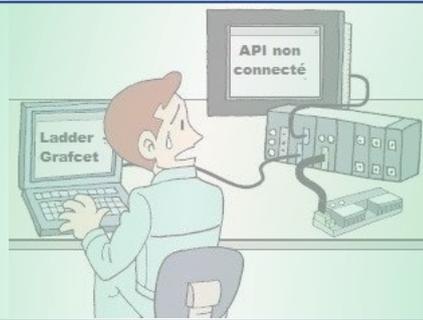
Comme on n'a pu le voir dans la précédente fiche, les automates programmables sont aujourd'hui très utilisés dans le monde industriel. Ils présentent plusieurs avantages parmi lesquels on peut citer : facilitation des opérations de câblage et de dépannage, temps de cycle plus rapide, fiabilité etc... Contrairement à un ordinateur traditionnel, l'automate a été conçu pour les environnements industriels sévères (température élevée, vibration etc...)

Les automates programmables offrent plusieurs avantages comparés aux systèmes à relais conventionnels. En effet dans certaines situations, il n'était pas économiquement viable de recâbler entièrement une armoire électrique lorsque l'on effectue une petite modification sur le système. Avec un automate programmable, on a pas nécessairement besoin de recâbler le système, cela permet d'éliminer les câblages fastidieux. Aujourd'hui, les systèmes de contrôle modernes comprennent toujours des relais, mais ceux-ci sont rarement utilisés pour la logique.

En plus des économies de coûts, les automates fournissent de nombreux autres avantages y compris la flexibilité : une fois qu'un programme a été écrit et testé, il peut être facilement téléchargé sur d'autres automates. Puisque toute la logique est contenue dans la mémoire de l'automate, on ne peut pas faire une erreur de câblage vu que les câblages sont fortement simplifiés.



Généralement, si un système comporte plus d'une demi-douzaine de relais, il sera probablement moins cher d'installer un automate. Grâce à leurs interfaces



de communication, les automates peuvent communiquer entre eux tout en permettant de contrôler et de collecter les données de process.

## II) Différence entre automate et PC

L'architecture matérielle d'un automate est sensiblement la même que celle d'un ordinateur personnel. En effet, un ordinateur personnel (PC) peut fonctionner comme un automate si on le connecte à des dispositifs comme des capteurs et boutons poussoirs. Par exemple, le Raspberry Pi qui est un mini-ordinateur fonctionnant sur Linux et disposant d'un port d'E/S peut être utilisé comme un contrôleur logique.

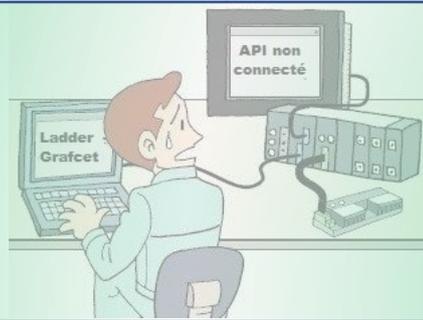


Raspberry Pi

Cependant, certaines caractéristiques importantes permettent de différencier un automate d'un ordinateur personnel. Contrairement aux PC, les automates sont conçus pour fonctionner dans des environnements industriels difficiles avec de larges plages de température ambiante et d'humidité. Aussi, une installation à base d'automate bien conçue n'est généralement pas affectée par les bruits électriques.

Contrairement à l'ordinateur personnel, l'automate est programmé en langage ladder et n'a pas de clavier ni de souris. A la place, les automates sont équipés de cartes d'E/S pour connecter des capteurs et actionneurs.

Quant aux ordinateurs, ce sont des machines informatiques complexes capables d'exécuter plusieurs programmes ou tâches simultanément et dans n'importe quel ordre. En revanche, la plupart des automates, ne sont capables d'exécuter qu'un programme unique et d'une manière ordonnée et séquentielle de la première à la dernière instruction.



Automate Proface

Aujourd'hui, on rencontre certains automates disposant d'un écran tactile permettant d'effectuer des fonctions de contrôle mais aussi des fonctions de supervision.

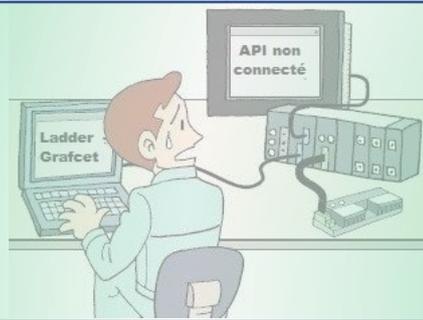
On a par exemple les automates Proface ou encore Unitronics.

### III) Les types d'automates et leurs critères de choix

Les critères utilisés pour catégoriser un automate sont : leur fonctionnalité, leur nombre d'entrées et de sorties, leur coût et leur taille. Parmi ces critères, le nombre d'E/S est le plus important. En général, les automates de type nano sont les plus petits avec moins de 15 E/S. Après les automates nano, on a les micros (avec 15 à 128 E/S), les medium avec 128 à 512 E/S et enfin



les grands avec plus de 512 E/S.



Lors du choix d'un automate, il faudra faire en sorte de choisir la bonne taille d'automate en fonction de l'ampleur du projet à réaliser. En général, il n'est pas conseillé d'acheter un automate pour l'intégrer dans un système dont les exigences sont moindres.

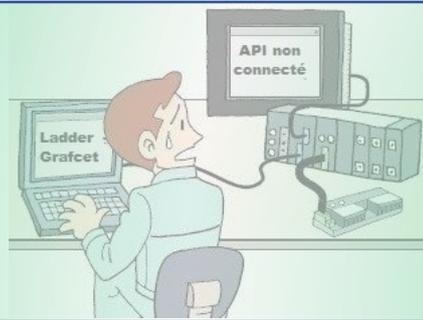
La taille et la sophistication du processus contrôlé sont des facteurs évidents pour déterminer quel automate choisir.

Il faudra aussi prendre en compte la taille de la mémoire de l'automate lors du choix de matériel. La mémoire est la partie de l'automate qui stocke les données, les instructions, et le programme de contrôle. La taille de la mémoire est généralement exprimée en Kilo-octets: 1 Ko, 6 Ko, 12 Ko, et ainsi de suite.

Le choix de la taille mémoire d'un automate dépend du système à concevoir et des facteurs suivants :

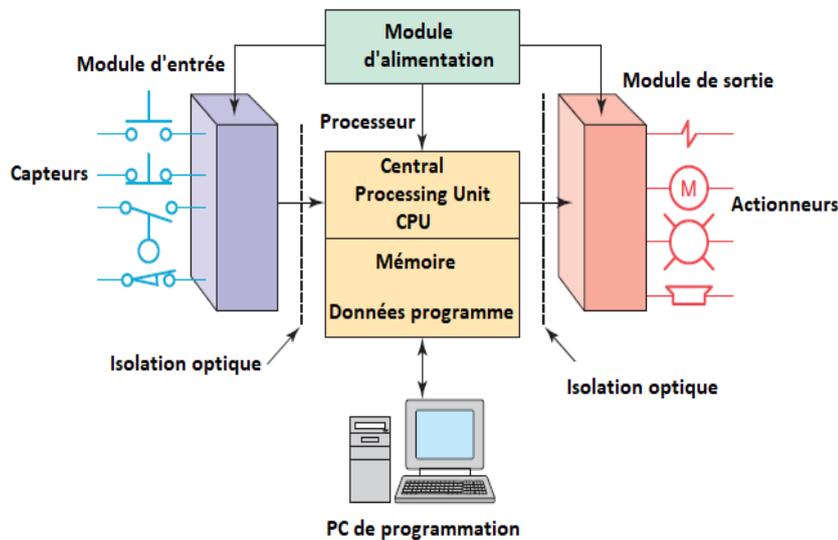
- Nombre d'E/S utilisés
  - Taille du programme de contrôle
  - Projet d'étendre le système dans le futur
- Etc...

Le jeu d'instructions pris en charge par un automate dépend aussi de la taille de l'automate. Ainsi certains automates (généralement les automates de type pico) ont des jeux d'instructions très limités comparés à d'autres qui offrent plus de fonctionnalités.



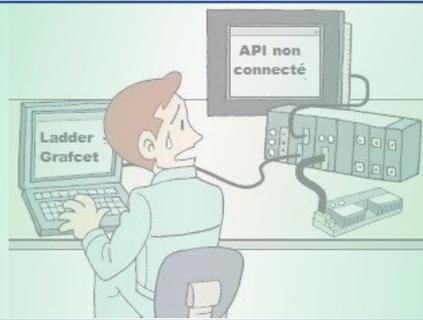
#### IV) Constitution et principe de fonctionnement d'un automate

Un automate peut être divisé en plusieurs parties, comme illustré dans l'image ci-dessous.



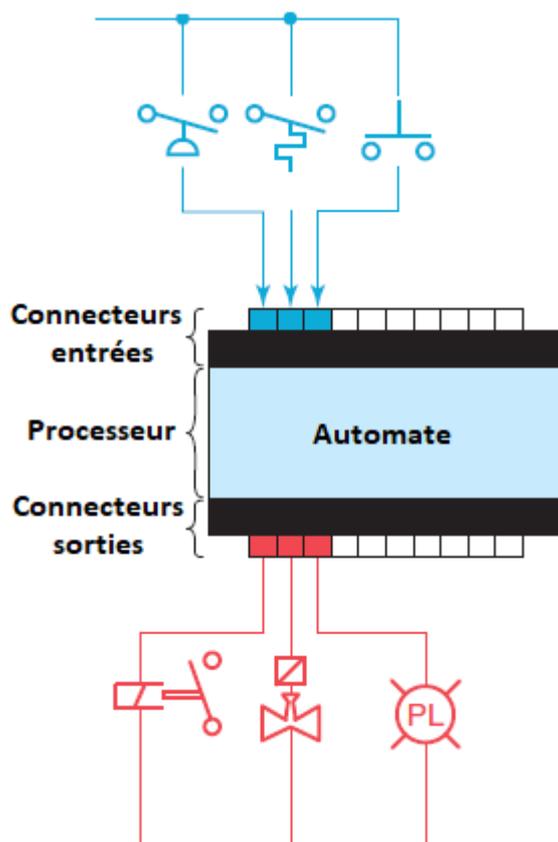
- Le CPU (Central Processing Unit) ou unité de traitement centrale
- La mémoire
- Les modules d'E/S
- Le module d'alimentation
- Le PC ou console de programmation

La plupart des automates sont en fait propriétaires, vous devez donc vous assurer que tout matériel ou logiciel générique que vous utiliserez est compatible avec votre automate. De manière générale, tous les automates sont constitués de la même manière, cependant, il pourrait y avoir de légères différences dans l'adressage, l'allocation de la mémoire, la récupération et le traitement des données pour différents modèles d'automates. Par conséquent, les programmes ne peuvent pas être échangés entre automates de différents fabricants.



On rencontre sur le marché des automates compacts dont les E/S sont incorporées dans le boîtier du CPU et des automates modulaires dont les E/S doivent être enfichées dans le rack de l'automate. Ainsi on aura des E/S fixes et des E/S modulaires selon le type de l'automate.

### ○ Les E/S fixes

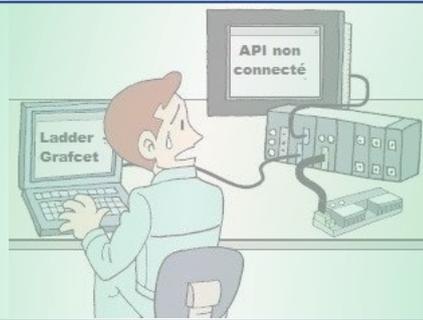


On les retrouve intégrées au niveau des petits automates programmables. Non amovibles, elles ne peuvent pas être retirées : le processeur et les E/S sont montés ensemble dans le même package.

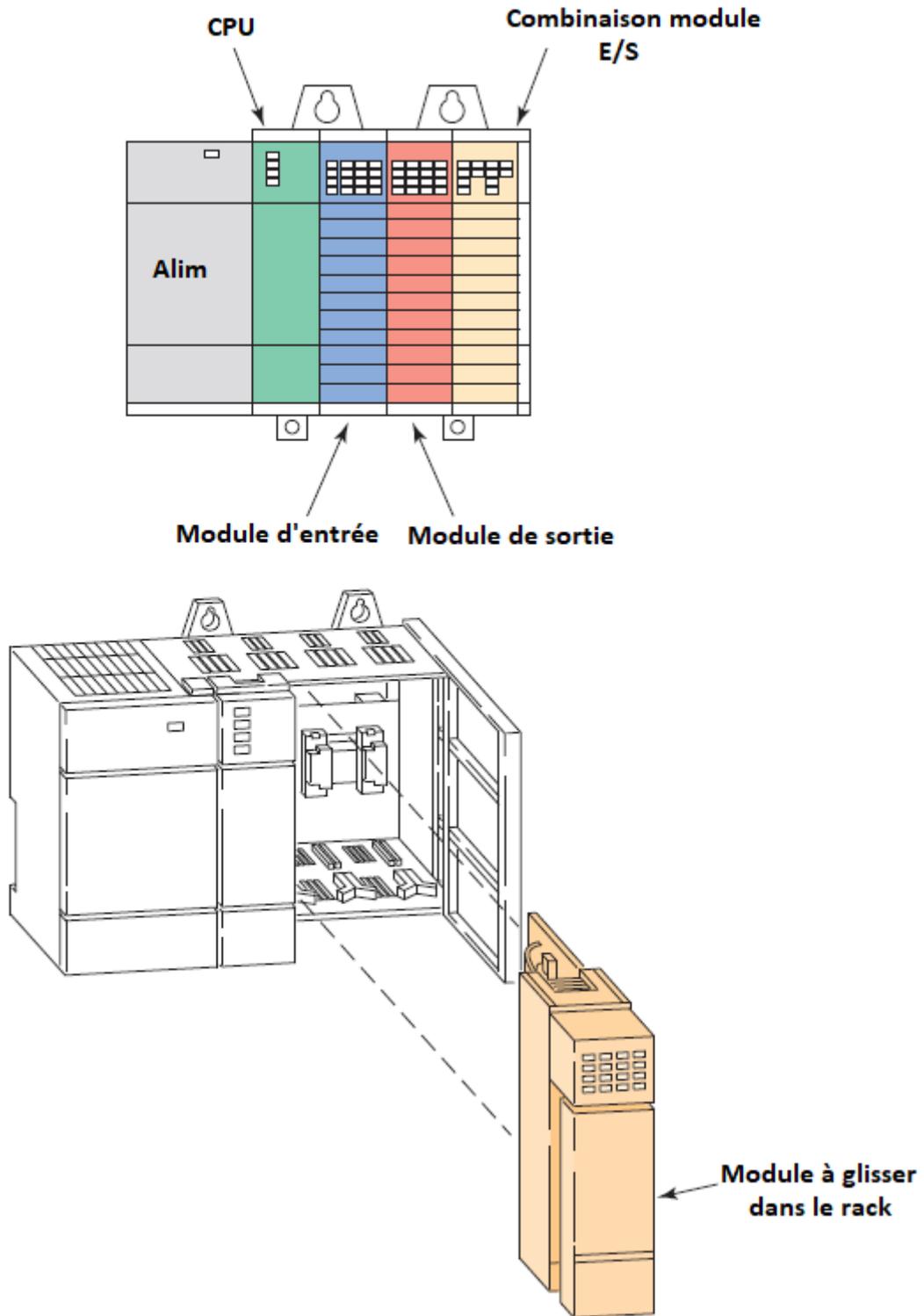
Les automates à E/S fixes présentent l'avantage d'être moins chers. Si on veut augmenter le nombre d'E/S de ces automates, on peut utiliser des modules d'extensions.

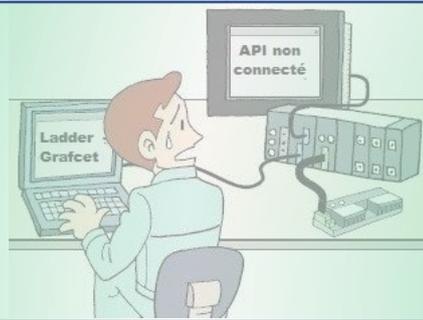
Par contre, un inconvénient des E/S fixes est leur manque de flexibilité on n'est limité en ce qui concerne le nombre de modules d'extensions pouvant être ajouté. Aussi, pour certains modèles d'automates à E/S fixes, si une partie de l'unité tombe

en panne, toute l'unité devra être remplacée.



- Les E/S modulaires





Avec ce type d'architecture, l'automate est divisé en compartiments dans lesquels des modules séparés peuvent être enfichés. Cela augmente considérablement la flexibilité de l'automate. On pourra par exemple choisir différents modules disponibles dans le catalogue du fabricant et les agencer à notre guise.

Les automates modulaires sont composés :

- D'un rack
- D'une alimentation
- D'un CPU
- D'un module d'E/S

Une interface de communication permettra de se connecter à d'autres équipements tels que des variateurs de vitesse, interface homme-machine etc...

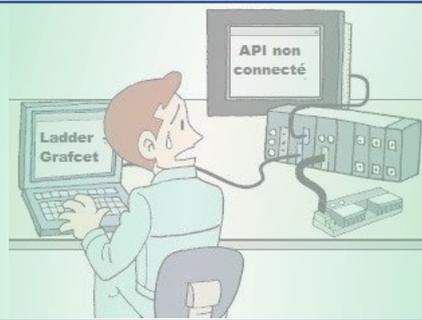
Au niveau des automates modulaires, les modules s'enfichent dans un rack.



Quand un module est glissé dans le rack, il établit une connexion électrique avec une série de contacts appelé fond de panier, situé à l'arrière du rack.

Le processeur de l'automate est également connecté au fond de panier et peut

communiquer avec tous les modules du rack. Le module d'alimentation fournit une alimentation en courant continu aux autres modules de l'automate.



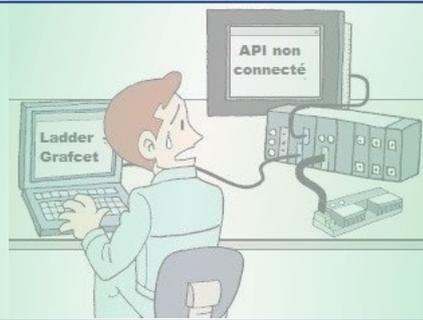
Au niveau des automates de grandes tailles, on utilisera souvent une alimentation externe pour alimenter les dispositifs qui seront connectés à l'automate. Pour les petits automates, l'alimentation intégrée est utilisée pour alimenter les équipements connectés à l'automate (capteurs, actionneurs etc..).

Le CPU est le "cerveau" de l'automate. Un CPU d'automate se compose généralement d'un microprocesseur pour mettre en œuvre la logique et le contrôle de la communication entre les différents modules.



**Modules automate Rockwell**

Il a besoin d'une mémoire pour stocker les résultats des opérations logiques effectuées par le microprocesseur. Ainsi, le CPU utilisera des mémoires de type EPROM,EEPROM ou RAM pour stocker temporairement le programme de l'automate. Celui-ci est exécuté de la première à la dernière instruction et en boucle tant que l'automate est en mode RUN.



On appelle temps de cycle de scrutation ou temps de cycle, le temps requis par l'automate pour lire ses entrées, exécuter son programme et mettre à jour ses sorties. Ce temps est variable en fonction :

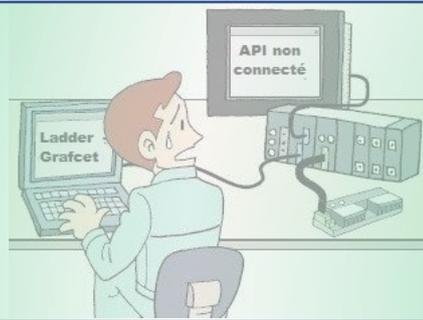
- Du nombre d'entrées et de sorties à gérer
- De la fréquence du microprocesseur de l'automate
- De la taille du programme à lire, sachant que certaines parties du programme peuvent être "sautées" temporairement

Lorsque l'automate lit ses entrées pendant une scrutation, les données lues sont stockées dans une mémoire image au fur et à mesure. Ceci permet à l'automate de disposer pendant le traitement du programme d'une seule valeur d'entrée. L'inconvénient est que si une entrée change d'état plus rapidement que la durée du cycle de scrutation, l'automate ne la prendra pas en compte. Pour éviter ce problème il existe des cartes d'entrées spéciales dont les entrées sont scrutées en permanence, indépendamment du cycle de scrutation (cartes d'interruption).

Pendant le traitement du programme l'API n'accède pas aux cartes d'entrées ou de sortie (sauf commande spéciales, écrites dans le programme). L'état des entrées est lu dans la MIE (mémoire image des entrées), les valeurs de sortie sont positionnées dans la Mémoire Image des Sorties (MIS).

Lors du positionnement (écriture) des sorties, l'API copie l'état des sorties de sa mémoire de donnée (MIS) vers les cartes de sorties. Ce n'est qu'à ce moment-là que les sorties physiques de l'automate vont éventuellement changer d'état.

Sur certains API la lecture de entrées s'effectue juste avant ou après le positionnement des sorties. Sur certains "gros" API, comportant un nombre important de rack d'extension, le cycle de scrutation (lecture des entrées, traitement du programme, positionnement des sorties) s'effectue rack par rack, le programme étant découpé en autant de tranches (segment) que de rack.



## Le chien de garde ou watchdog

C'est une minuterie de valeur (fixe ou réglable) légèrement supérieure au temps de cycle maximum autorisé pour un automate donné.

Cette minuterie est enclenchée à chaque début de cycle.

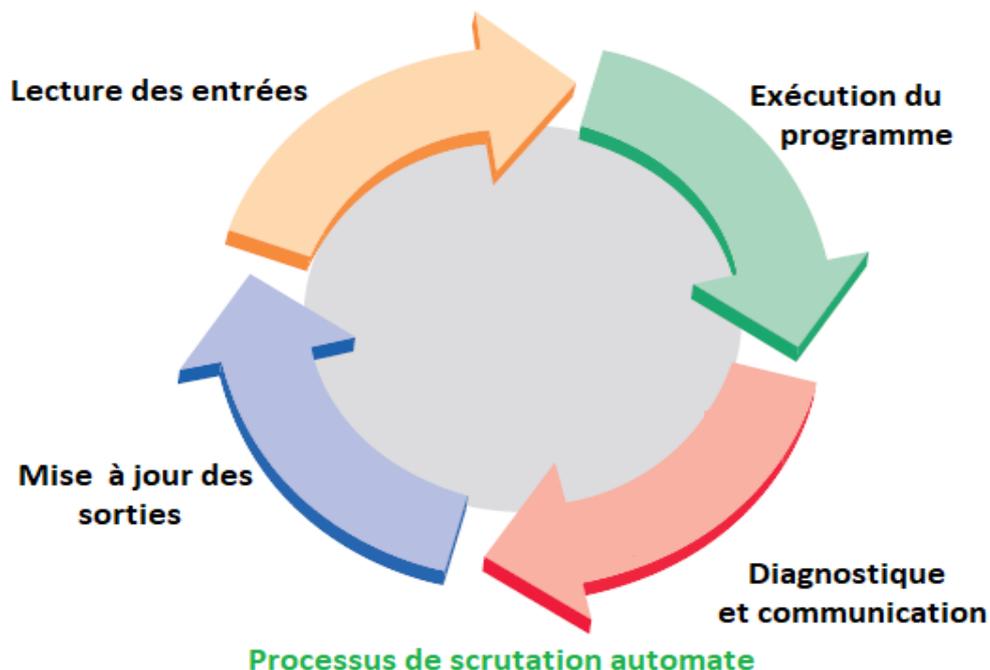
Si le cycle de scrutation se termine avant la fin de la minuterie, tout va bien, et l'automate continue son travail.

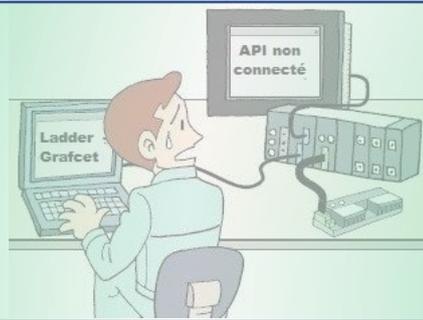
Si la minuterie se termine avant la fin du cycle de scrutation, une alarme se déclenche, se traduisant, suivant l'automate, par :

- L'allumage ou le clignotement d'un voyant,
- Le positionnement d'un indicateur dans la mémoire de l'automate (bit système),
- La retombée d'un contact électrique (souvent câblé dans la chaîne d'arrêt d'urgence du système à automatiser),
- Le passage de l'API en mode arrêt.

Ce défaut peut être provoqué par plusieurs causes :

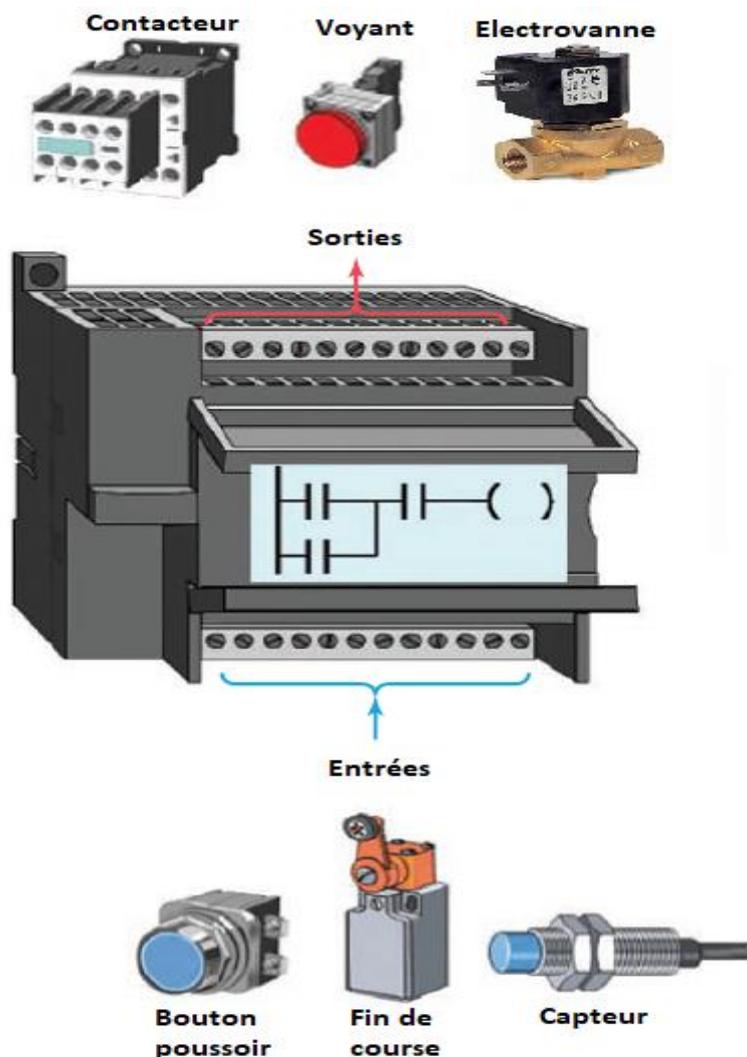
- Problème d'accès sur une carte d'entrées/sorties,
- Boucle sans fin dans le programme, consécutive à un saut en arrière ou une instruction répétitive dont la condition de fin est mal définie,

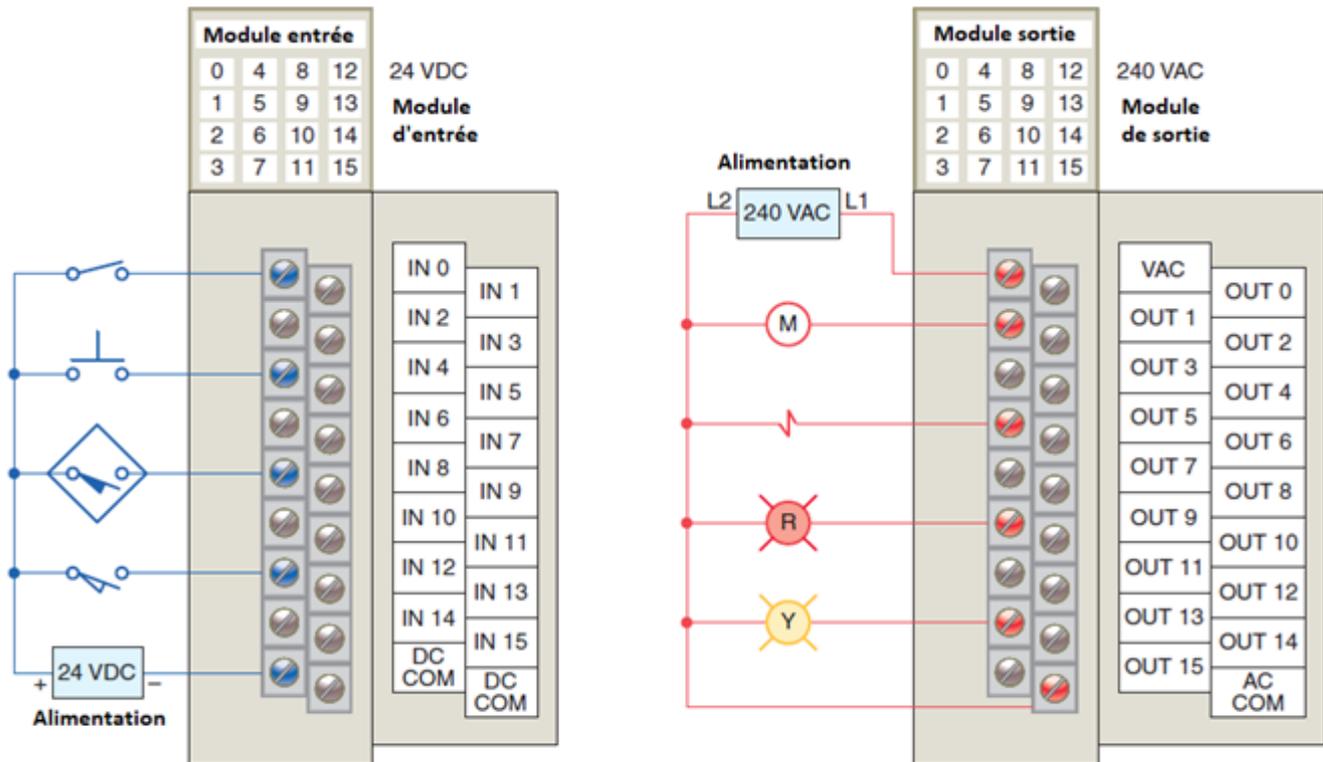
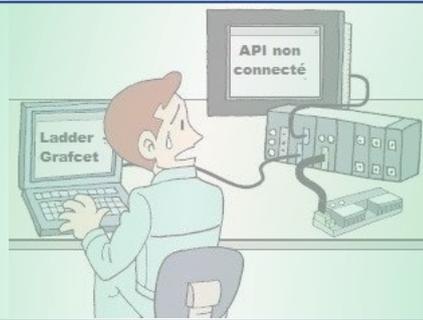




Lors de la scrutation, le CPU lit l'état des entrées, ensuite, le programme est exécuté. Une fois l'exécution du programme terminée, le processeur effectue des tâches de diagnostic et de communication interne. Ensuite, l'état de toutes les sorties est mise à jour. Ce processus est répété en continu tant que l'automate est en mode de fonctionnement.

Les modules d'entrées/sorties servent d'interface entre les équipements de terrains (boutons poussoirs, capteurs, actionneurs etc..) et l'automate.



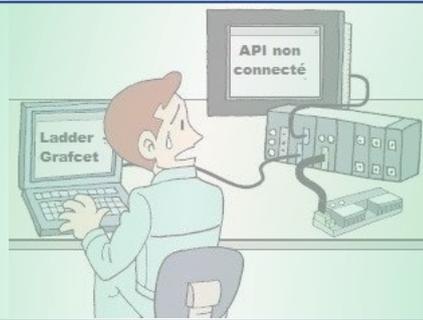


### Modules entrée/sortie sur lesquels sont connectés les capteurs et actionneurs

Leur rôle est de conditionner les différents signaux reçus ou envoyés vers des dispositifs externes. Ainsi, le module d'entrée permet de conditionner les signaux collectés depuis les boutons poussoirs, interrupteurs de fin de course et autres capteurs câblés aux bornes d'entrée.

Le module de sortie permet d'envoyer des signaux aux actionneurs tels que des moteurs, électrovannes et voyants qui sont câblés aux bornes de sortie.

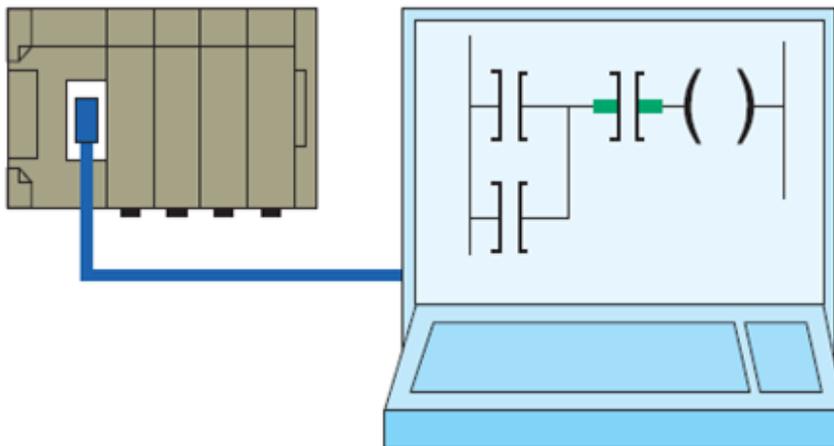
Pour isoler électriquement les composants internes des bornes d'entrée et de sortie, les automates emploient couramment un isolateur optique, qui utilise la lumière pour coupler les deux circuits ensemble.



Console de programmation

Une console de programmation est utilisée pour entrer le programme désiré dans l'automate. Elle est souvent utilisée avec les petits automates car peu coûteux et facile à utiliser comparé à un PC de programmation. Une fois branchée sur l'automate, elle peut être utilisée pour entrer et consulter le programme de l'automate.

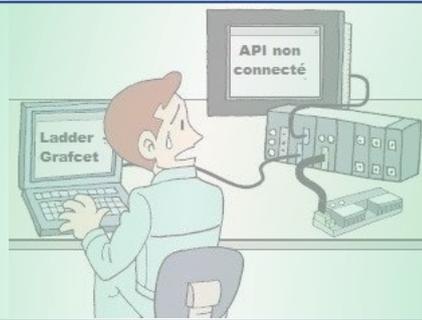
En usine, vous utiliserez souvent des consoles ou PC de programmation pour le dépannage, la modification ou le transfert de programme automate. Les PC de programmation communiquent avec l'automate via un port série, parallèle ou



PC de programmation

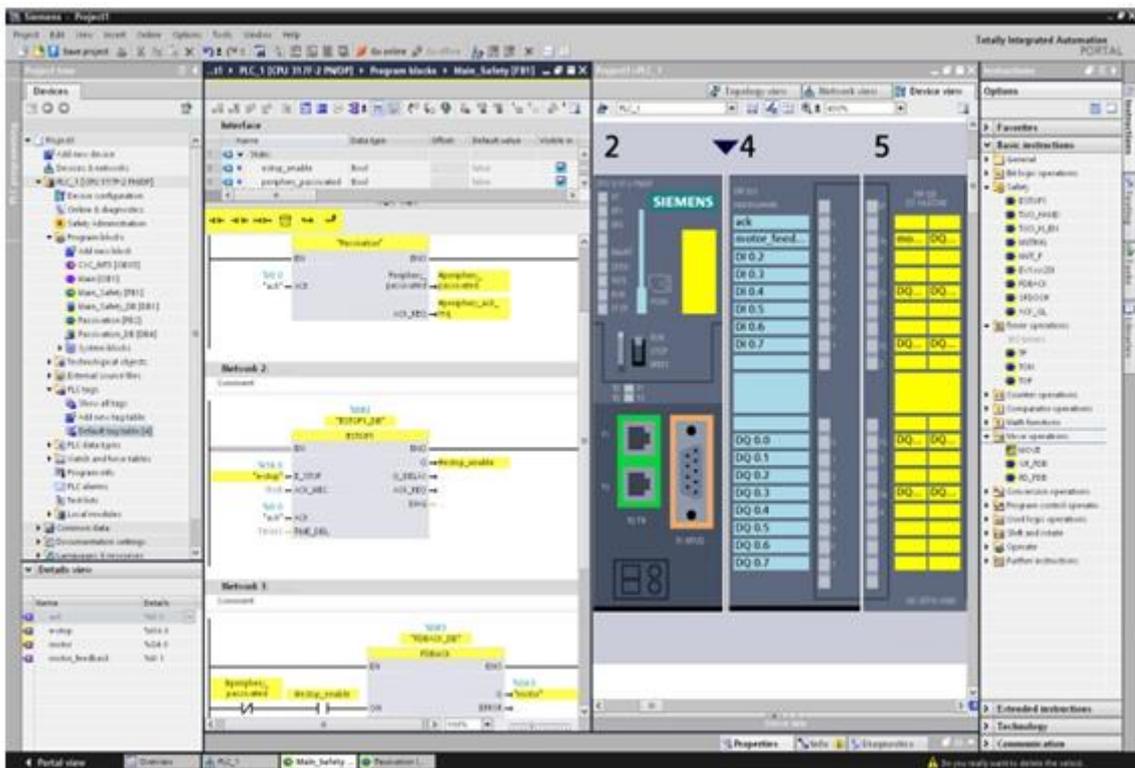
Ethernet. Une fois que le programme est rentré dans l'automate, on peut débrancher la console ou le PC de programmation sans affecter le bon fonctionnement de l'automate.

Un programme est une série d'instructions développée par l'utilisateur et exécutée par le CPU de l'automate. Le langage ladder est le langage de programmation standard le plus utilisé pour programmer des automates. Il est basé sur le même principe



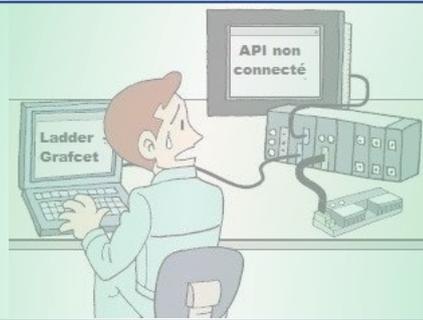
que les relais électromécaniques. Un programme Ladder est composé de contacts, de bobines et de blocs d'instructions spéciales.

La plupart des fabricants d'automates ont un logiciel de programmation utilisable sur PC. Ces logiciels permettent aux utilisateurs de créer, d'éditer, de documenter, d'archiver et de dépanner des programmes via PC. Parmi ces logiciels, on peut citer TIA portal de Siemens ou encore Unity Pro de Schneider Electric.



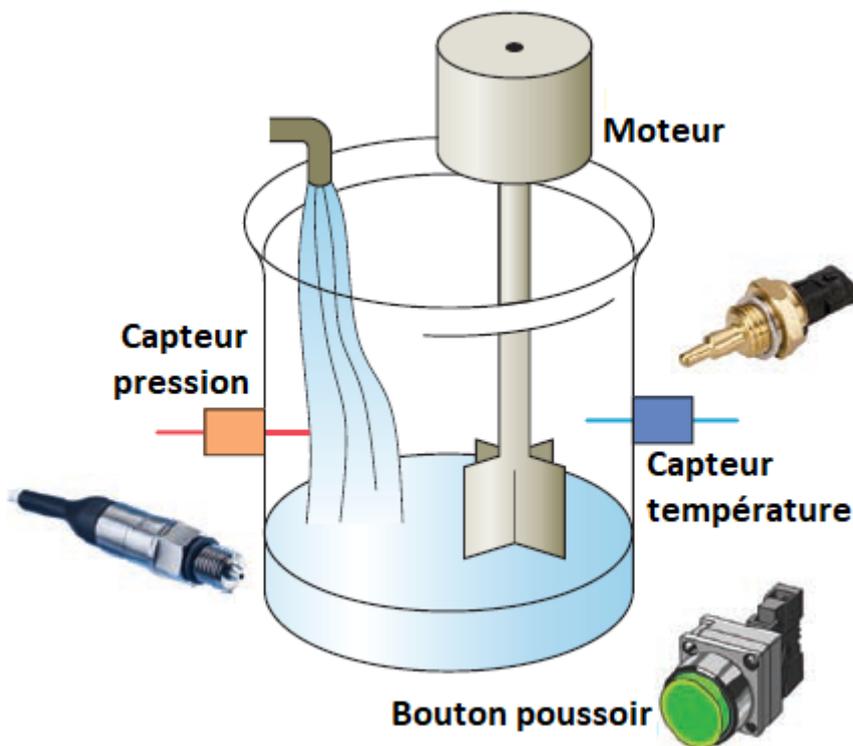
### Logiciel TIA Portal de Siemens

Le programme d'un automate peut être réalisé en langage Ladder ou langage à contacts qui est l'un des langages de programmation les plus populaires, en Grafset, ST, FBD ou IL. Le langage Ladder utilise des symboles graphiques suivant le même principe que les schémas de commande à base de relais. C'est un langage très facile à prendre en main surtout pour les personnes habituées aux schémas de commande électrique.



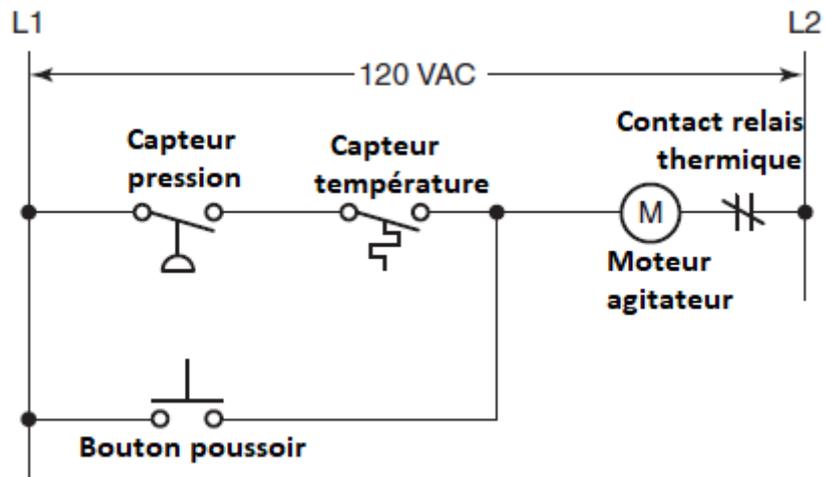
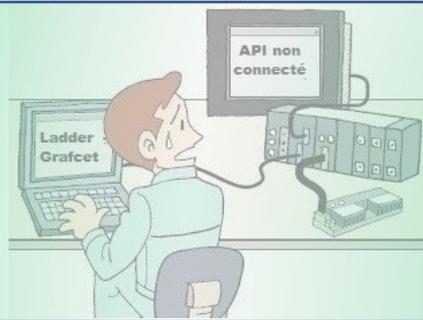
## V) Exemple de cahier des charges

Pour avoir une idée plus nette du fonctionnement d'un automate, on va prendre un exemple de cahier des charges. Considérons le processus illustré sur l'image ci-dessous.



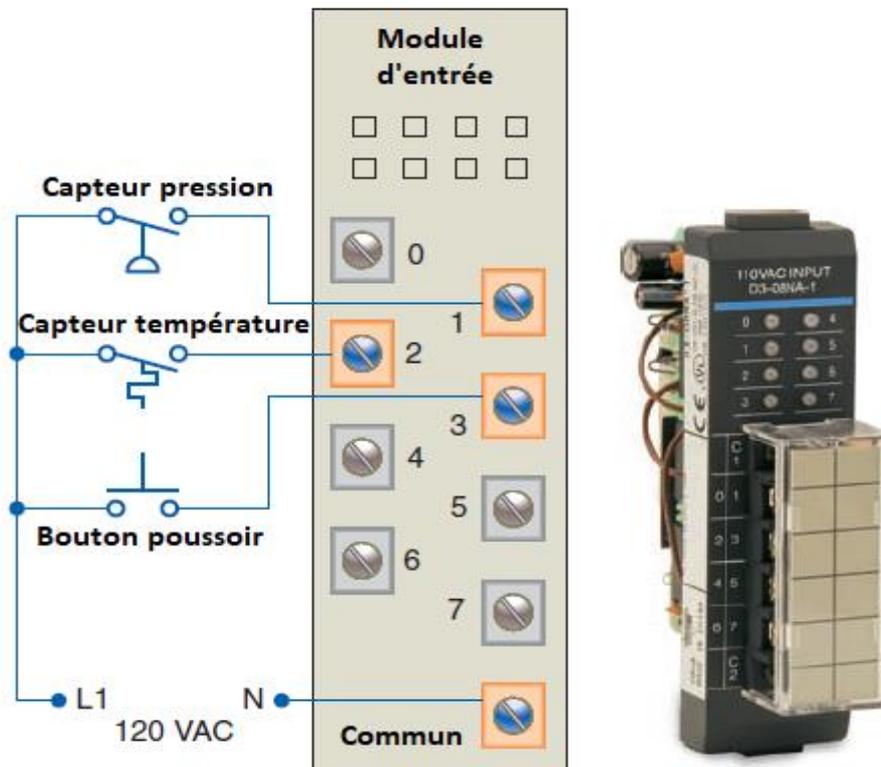
Nous avons un mélangeur constitué d'un moteur qui permet d'agiter automatiquement le liquide contenu dans une cuve lorsque la température et la pression atteignent une certaine valeur. Un bouton poussoir permet de démarrer manuellement le moteur de l'agitateur. Le processus est surveillé grâce à des capteurs de température et de pression qui ferment leurs contacts respectifs quand les valeurs des capteurs franchissent un certain seuil prédéfini.

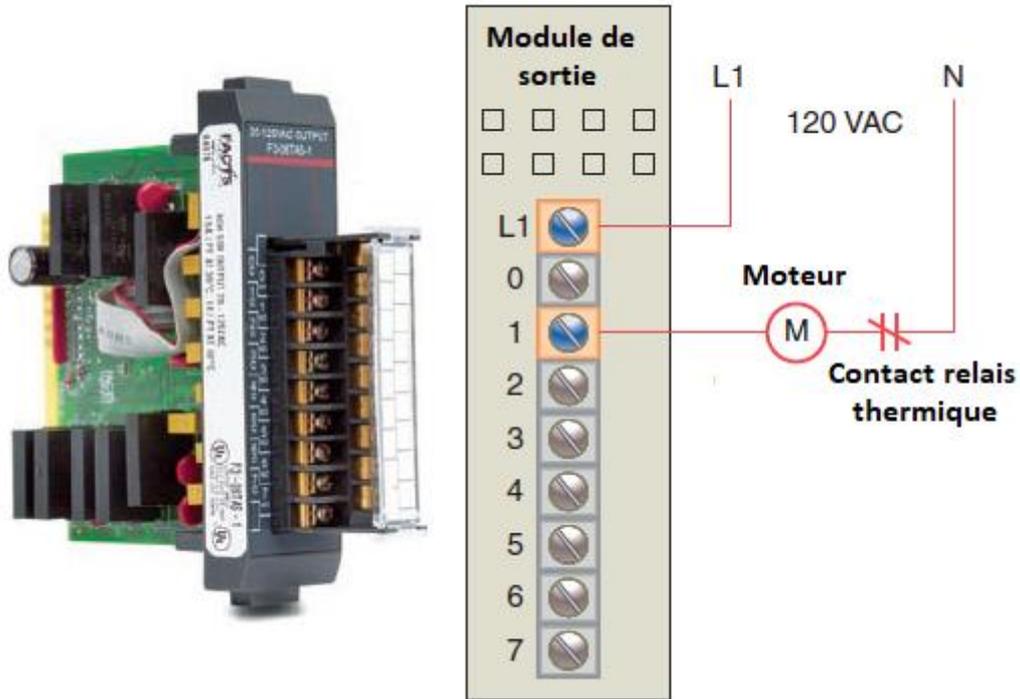
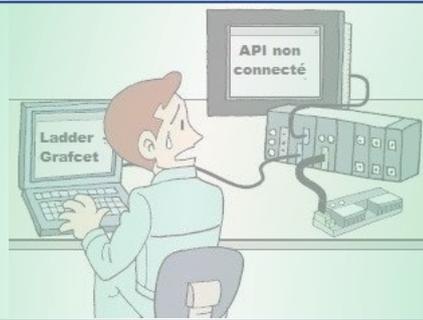
Afin de réaliser le système énoncé, on peut soit utiliser des relais soit utiliser un automate. Si on utilise des relais, on aura le schéma représenté ci-dessous.



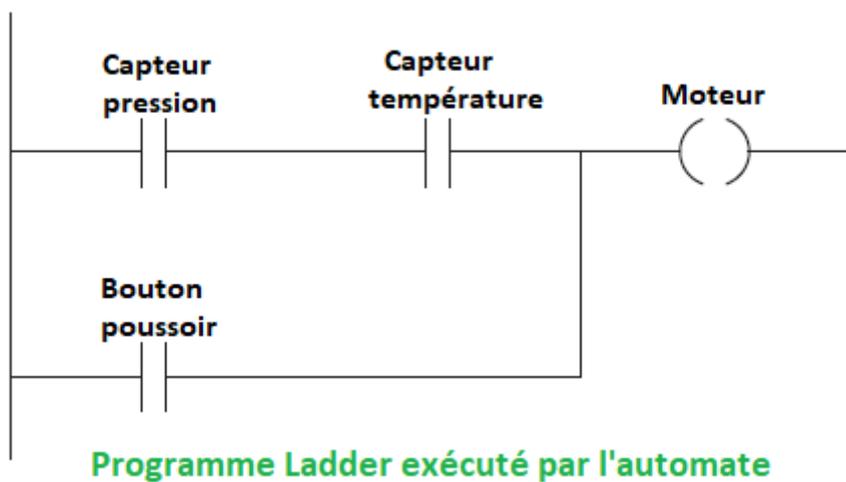
Le moteur de l'agitateur est en marche que lorsque les interrupteurs des capteurs de pression et de température sont fermés ou lorsque le bouton-poussoir manuel est enfoncé.

Maintenant, regardons comment un automate pourrait être utilisé pour cette application. On aura les schémas de câblage et le programme automate ci-dessous.

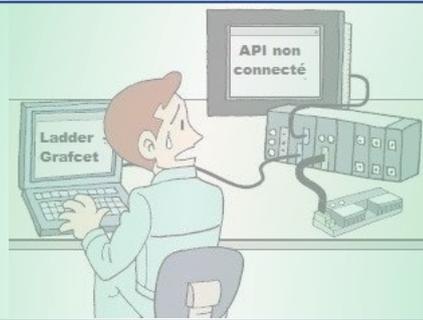




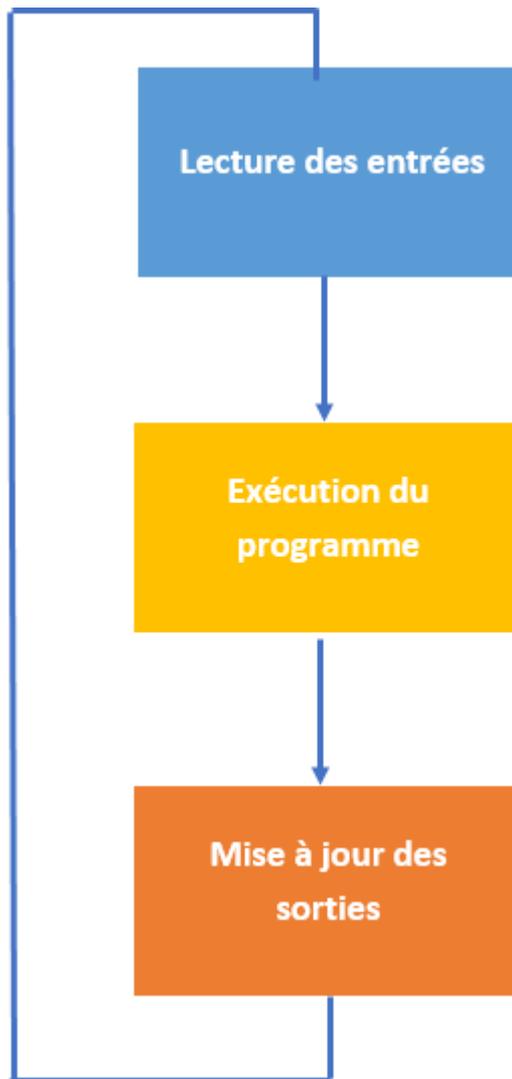
Ci-dessous, le programme Ladder qui sera exécuté dans l'automate.

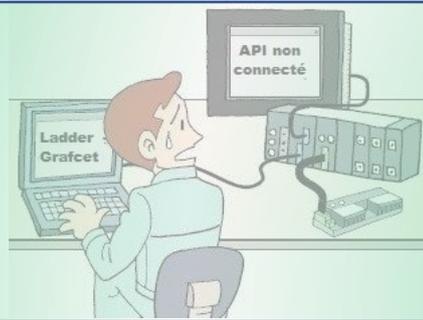


Au niveau de l'automate, chaque entrée et chaque sortie est affectée à une adresse. Cela permet à l'automate de savoir où elles sont physiquement connectées.



Comme on l'a vu un peu plus haut, lors du démarrage de l'automate, celui-ci lit ses entrées, exécute le programme de contrôle et modifie ses sorties en conséquence.





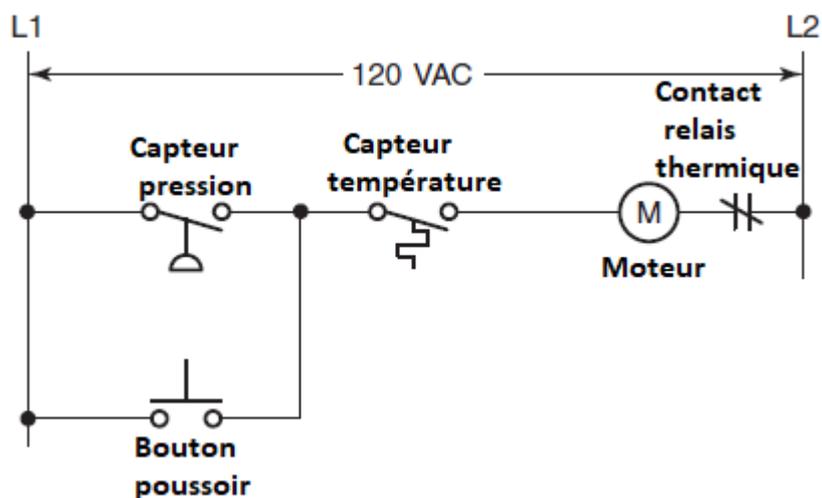
Pour que le programme fonctionne, l'automate doit être placé en mode RUN. Une fois que le CPU de l'automate est en état de marche, l'état du capteur de pression, du capteur de température et du bouton-poussoir est examiné et enregistré dans la mémoire de l'automate :

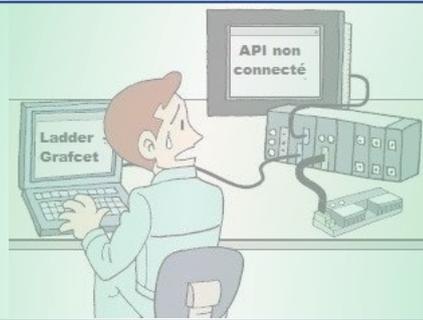
- Un contact fermé est enregistré en mémoire en tant que « 1 logique »
- Un contact ouvert comme « 0 logique »

Ensuite, le programme Ladder est évalué, chaque contact interne OUVERT ou FERMÉ recevra respectivement les valeurs 0 ou 1 en fonction de leur état.

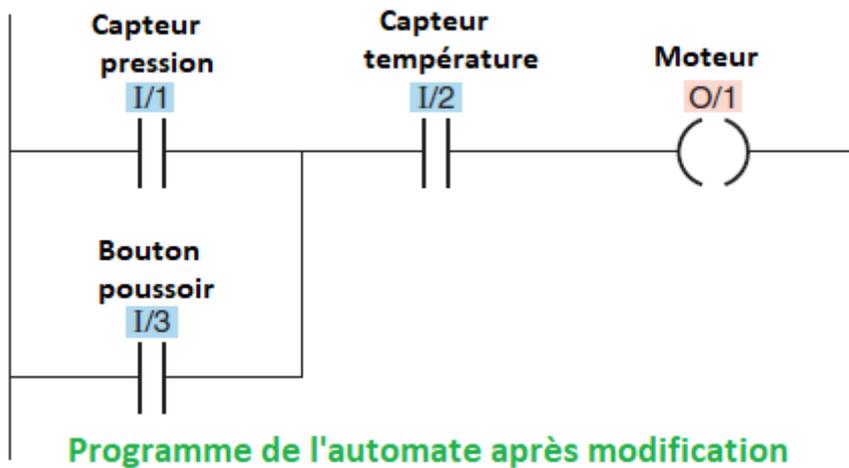
## VI) **Avantage d'un automate face aux systèmes automatisés à base de relais**

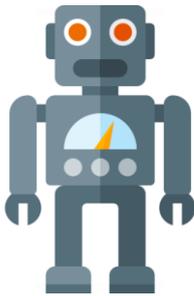
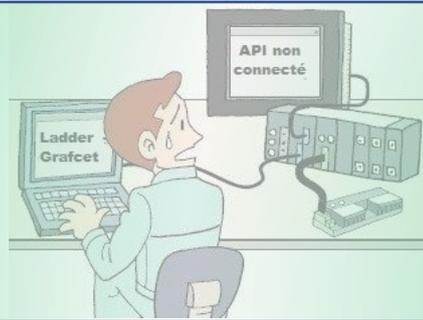
Comme on a pu le voir précédemment, l'un des avantages d'un automate programmable est sa flexibilité. En effet, le fonctionnement d'un système automatisé peut être facilement changé juste en modifiant le programme automate. Si l'on prend l'exemple du système de mélange précédent, si celui-ci devait être modifié comme indiqué sur le schéma ci-dessous.





En logique câblée, on aurait besoin de recâbler le système en prenant en compte les modifications. Cependant, avec un automate (logique programmée), on aura pas besoin de recâbler le système. Les entrées et sorties resteront inchangées, seul le programme de l'automate va changer.





### **BILAN :**

**Dans cette fiche, vous avez pu découvrir les différents éléments qui composent un automate et son principe de fonctionnement.**

**Dans la prochaine fiche, nous rentrerons plus en détails sur la constitution matérielle d'un automate.**

**Vous pouvez effectuer sans plus tarder le test QCM afin d'évaluer vos connaissances.**