

Système automatisé

1/ Système automatisé

Un système automatisé est composé de plusieurs éléments qui exécutent un ensemble de tâches programmées sans que l'intervention de l'homme ne soit nécessaire. Exemples : le passage à niveau automatique, la porte de garage, etc...

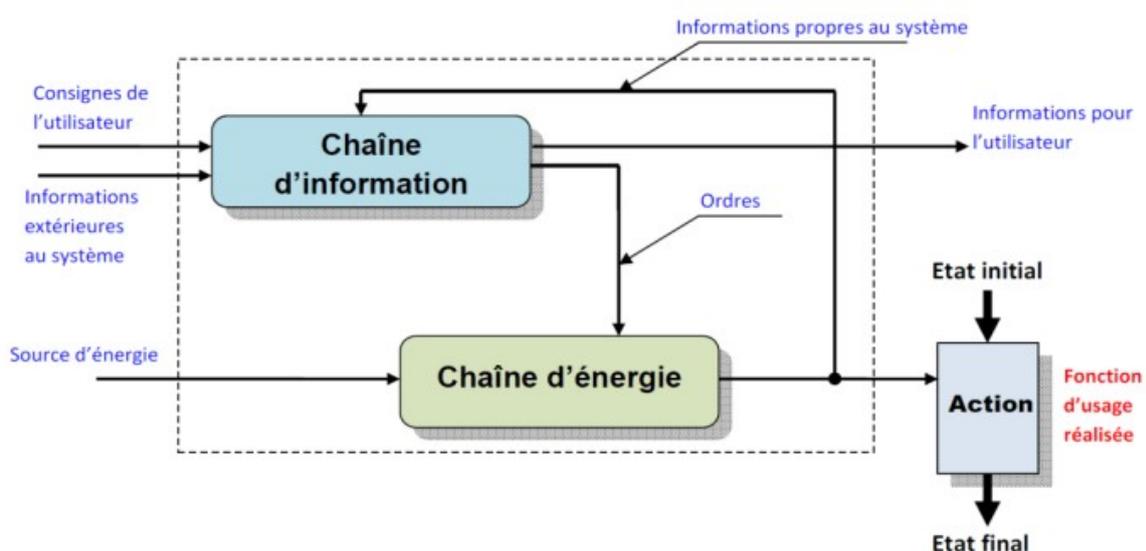
Il est composé de:

- Chaîne d'information / La partie commande (PC): elle donne les ordres et reçoit les informations de l'extérieur ou de la partie opérative. Elle peut se présenter sous 3 manières différentes : un boîtier de commande, un **microprocesseur** (cerveau électronique), ou un **ordinateur**
- Chaîne d'énergie / La partie opérative (PO): c'est la partie d'un système automatisé qui effectue le travail. Autrement dit, c'est la machine. C'est la partie qui reçoit les ordres de la partie commande et qui les exécute. Elle comporte les capteurs et les actionneurs:

Un **actionneur** est un élément de la partie opérative qui est capable de produire une action physique tel qu'un déplacement, un dégagement de chaleur, une émission de lumière ou de son à partir de l'énergie qu'il a reçu.

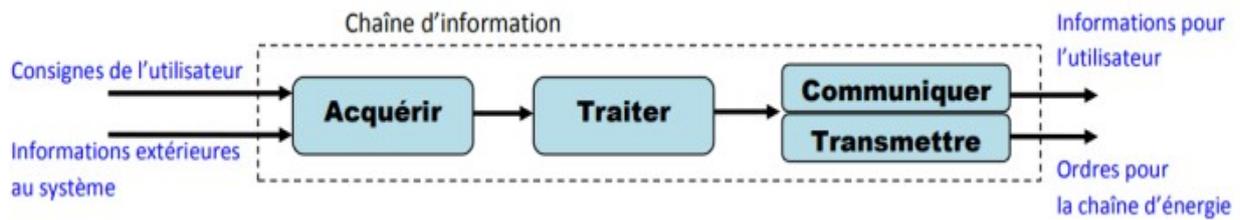
Un **capteur** est un élément de la partie opérative qui permet de recueillir des informations et de les transmettre à la partie commande. Les capteurs sont choisis en fonction des informations qui doivent être recueillies (température, son, lumière, déplacement, position).

2/ Schéma d'un système automatisé



3/ Chaîne d'information

Définition : c'est la partie du système automatisé qui capte l'information et qui la traite. On peut découper cette chaîne en plusieurs blocs fonctionnels.



Acquérir : Fonction qui permet de prélever des informations à l'aide de capteurs.

Traiter : C'est la partie commande composée d'un automate ou d'un microcontrôleur.

Communiquer : Cette fonction assure l'interface l'utilisateur et/ou d'autres systèmes.

Transmettre : Cette fonction assure l'interface avec l'environnement de la partie commande.

Exemple de chaîne d'information : la porte de garage

L'opérateur appuie sur le bouton de la télécommande pour fermer la porte du garage (*consigne de l'utilisateur*). **La chaîne d'informations**, composée d'un boîtier électronique et de capteurs, détecte le signal et ordonne, lorsqu'elle en reçoit l'ordre, la mise en route du moteur afin d'ouvrir la porte (*ordre*).

Automates programmable industriel



Deux automates programmables industriels & leurs périphériques, montés en volant, pour test et analyse. Automate industriel WAGO pour un système de monitoring en industrie pharmaceutique.

Un **automate programmable industriel**, ou **API** (en anglais **programmable logic controller, PLC**), est un système [électronique numérique programmable](#) destiné au [contrôle](#) de [procédés industriels](#) par traitement séquentiel. Il envoie des instructions à des *préactionneurs* (partie active ou *PA* côté actionneur) à partir d'entrées (signaux entrant provenant de [capteurs](#)) (partie contrôle ou *PC* côté capteur), de [consignes](#) et d'un [programme informatique](#).

Lorsqu'un automate programmable est orienté sur la Sécurité, il est alors appelé **automate programmable de sécurité** ou **APS**.

Présentation

[[modifier](#) | [modifier le code](#)]



Automate de Allen-Bradley installé dans une armoire.



Automate dans une armoire électrique.

On nomme automate programmable industriel (API) un type particulier d'[ordinateurs](#), robuste et réactif, équipés de prises en entrée et en sortie, utilisés pour automatiser des procédés, notamment pour contrôler / réguler des [machines](#) sur une chaîne de montage, dans une usine, ou pour contrôler des systèmes de manutention automatique. Là où les systèmes automatisés plus anciens employaient des centaines ou des milliers de [relais](#) et de [cames](#), un simple automate suffit. On nomme [automaticiens](#) les [programmeurs](#) de ces API.

L'API est structuré autour d'un système de calcul ou [processeur](#) (en anglais *Central Processing Unit*, CPU), d'une alimentation par des courant alternatifs (AC) ou continu (DC), et de modules dépendant des besoins du procédé, notamment :

- Des cartes d'entrées - sorties (en anglais Input - Output, I/O) numériques ([tout ou rien](#)) pour des signaux à deux états ou [analogiques](#) pour des signaux à évolution continue
 - Cartes d'entrées pour brancher des [capteurs](#), [boutons poussoirs](#), etc.
 - Cartes de sorties pour brancher des [actionneurs](#), [voyants](#), [vannes](#), etc.
- Des modules de communication obéissant à divers protocoles [Modbus](#), Modbus Plus, [Profibus](#), [InterBus](#), [DeviceNet](#), [LonWorks](#), [Ethernet](#), FIPIO, [FIPWAY](#), [RS232](#), [RS-485](#), [AS-i](#), [CANopen](#), pour dialoguer avec d'autres automates, des entrées/sorties déportées, des supervisions ou autres interfaces homme-machine ([IHM](#), en anglais Human Machine Interface, [HMI](#)), etc.
- Des modules spécifiques aux métiers, tels que comptage rapide, pesage, etc.
- Des modules d'[interface](#) pour la commande de mouvement, dits modules Motion, tels que démarreurs progressifs, [variateurs de vitesse](#), commande d'axes.
- Des modules locaux de dialogue homme-machine tels qu'un [pupitre](#) (tactile ou avec clavier), un terminal de maintenance, reliés à l'automate via un réseau industriel propriétaire ou non et affichant des messages ou une représentation du procédé.

D'autres automates, plus anciens, étaient constitués d'une simple [mémoire](#) dont l'adresse d'entrée était constituée d'une concaténation de données d'entrée (senseurs, horloge) et de l'état précédent. Beaucoup moins onéreux, ils se prêtaient en revanche mal à une augmentation rapide du nombre d'états. Ils sont restés très utilisés pour des automatisations simples du style [système anti-blocage des roues \(ABS\)](#) ou [feux de signalisation](#) aux carrefours.

Par rapport aux ordinateurs, les API se caractérisent par :

- leur robustesse : conçus pour pouvoir travailler en milieu hostile, ils utilisent des circuits durcis et sont prévus pour résister aux vibrations, aux températures des ateliers, etc ;
- leur réactivité aux indications fournies par les capteurs (dispositifs anticollision, alarmes diverses) ;
- leur facilité de maintenance (bien que les ordinateurs industriels atteignent également un très bon degré de fiabilité). Les modules peuvent être changés très facilement et le redémarrage des API est très rapide.

L'absence d'Interface Homme-machine (IHM) permanente pour visualiser l'action et le fonctionnement du programme sur la partie opérative font que les automates sont souvent reliés à un pupitre opérateur, une interface graphique (écran d'affichage ou écran tactile) ou un PC. Dans ce dernier cas, on parle de supervision. Le PC peut d'ailleurs être utilisé seul en regroupant les fonctions de l'API et de la supervision, grâce à l'utilisation d'un [softPLC](#).

En automatisme industriel, on parle aussi beaucoup d'automates de télégestion. Dans ce cas, on vient, via Internet, modifier ou visualiser à distance les données ou le programme des automates de gestion des installations commandées: chaudières collectives, stations d'épuration, etc. Cela se fait par le biais de modem-routeurs souvent associés à un logiciel assurant une liaison sécurisée ([VPN](#)).

En général, si API et PC coexistent dans un atelier, les API fonctionnent au plus près des processus physiques et prennent en charge les questions de sécurité, les PC s'occupant plutôt de supervision et des rapports extérieurs. Les PC peuvent ainsi

fixer au mieux les consignes aux API, qui donnent les ordres détaillés, traitent les urgences, et rendent compte de l'état des processus.

Programmation

Les programmes des API sont traités selon un cycle précis, le plus souvent¹ :

1. Diagnostic (auto-test) ;
2. Acquisition de toutes les entrées (recopie dans une mémoire image) ;
3. Traitements du programme ;
4. Mise à jour des sorties.

Le temps d'un cycle d'API varie selon la taille du programme, la complexité des calculs, le nombre d'entrées/sorties, la puissance de l'API, et les besoins du procédé piloté. Il varie de une à quelques dizaines de millisecondes et est protégé par un [chien de garde](#), au cas par exemple où l'algorithme exécuterait indéfiniment une même boucle de programme.

Lecture des capteurs et commande des actionneurs sont réalisés par scrutation, la gestion d'interruptions pouvant être victime d'un effet d'avalanche en cas d'incident.

Différents langages de programmation

[[modifier](#) | [modifier le code](#)]

Il existe différents [langages de programmation](#) définis par la [CEI 61131-3](#) :

- [IL \(Instruction List\)](#), le langage List est très proche du [langage assembleur](#) on travaille au plus près du [processeur](#) en utilisant l'[unité arithmétique et logique](#), ses [registres](#) et ses [accumulateurs](#)
- [ST \(Structured Text\)](#), Ce langage structuré ressemble aux langages de haut niveau utilisés pour les [ordinateurs](#)
- [LD \(Ladder Diagram\)](#), le [langage Ladder](#) (échelle en anglais) ressemble aux [schémas](#) électriques et permet de transformer rapidement une ancienne application faite de [relais électromécaniques](#) en un programme. Cette façon de programmer exploite une approche visuelle du problème longtemps appréciée en industrie, mais qui s'appuie sur une logique de moins en moins adaptée mais toujours utilisée (2013). On parle également de [langage à contacts](#) ou de [schéma à contacts](#) pour désigner ce [langage Ladder](#).
- [Boîtes fonctionnelles](#) (FBD), le [FBD](#) se présente sous forme [diagramme](#) : suite de blocs, connectables entre eux, réalisant des opérations, simples ou très sophistiquées.

Dans la programmation d'un automate, il est possible également de choisir de programmer en [SFC](#), dérivé du [grafcet](#). À chaque action élémentaire est associé un programme écrit en IL, ST, LD ou FBD. Le grafcet, très populaire en France, est un outil graphique de définition de l'automatisme séquentiel, en un nombre fini d'étapes, séparées par des conditions de transition. Il utilise une représentation graphique claire, permettant par exemple au réalisateur de montrer au donneur d'ordre comment il a compris le cahier des charges. Langage universel, indépendant (dans un premier temps) de la réalisation pratique, il peut se "câbler" par séquenceurs, être programmé sur automate voire sur ordinateur. De plus, il permet :

- de hiérarchiser les séquences ;
- de coordonner au sein d'un cycle des séquences interdépendantes se déroulant simultanément ;
- d'appliquer des conditions de validité sécurisant le cycle de pilotage ;
- enfin, d'exploiter la méthode GEMMA, méthode sécurisant la gestion des modes de marche et d'arrêt.

Dans le cas des automates programmables logiciels ([softplc](#)), il existe également différents langages de programmation non définis par la [CEI 61131-3](#) qui étendent considérablement les possibilités de configuration, par exemple:

- C/C++ : [Proview](#); Pascal : [Visual PLC](#). Grâce à leur flexibilité, ces logiciels sont utilisés sur les chaînes de fabrication automobile et sur les trains de laminoirs.

Toutefois, la popularité de ces langages ne doit pas masquer leurs faiblesses en matière de sécurité des processus.

Différents types de réseaux d'automates :

1- Réseau en étoile :

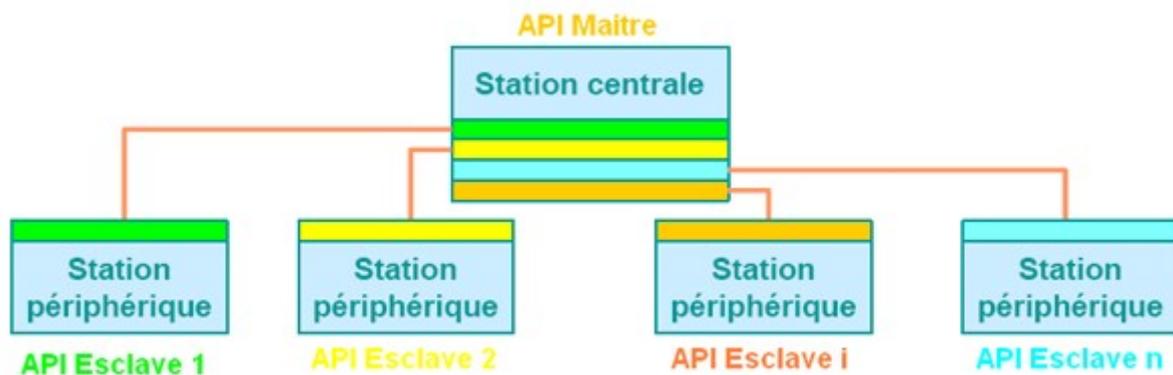
Un centre de traitement commun échange avec chacune des autres stations. Deux stations ne peuvent pas échanger directement entre elles (Figure 4.15). Exemple le réseau de terrain BITBUS de la société INTEL

Avantages :

- Grande vitesse d'échange.
- Différent types de supports de transmission.
- Pas de gestion d'accès au support.

Inconvénients :

- Coût global élevé.
- Evolutions limitées.
- Tout repose sur la station centrale.



Interconnexion par entrées/sorties déportées

2- Réseau en anneau :

Chaque station peut communiquer avec sa voisine. Cette solution est intéressante lorsqu'une station doit recevoir des informations de la station précédente ou en transmettre vers la suivante (Figure 4.16).

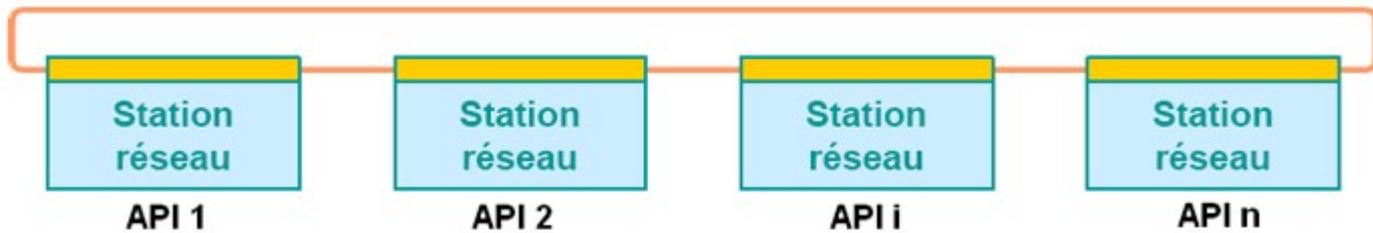


Figure 4.16: Topologie Anneau

Avantages :

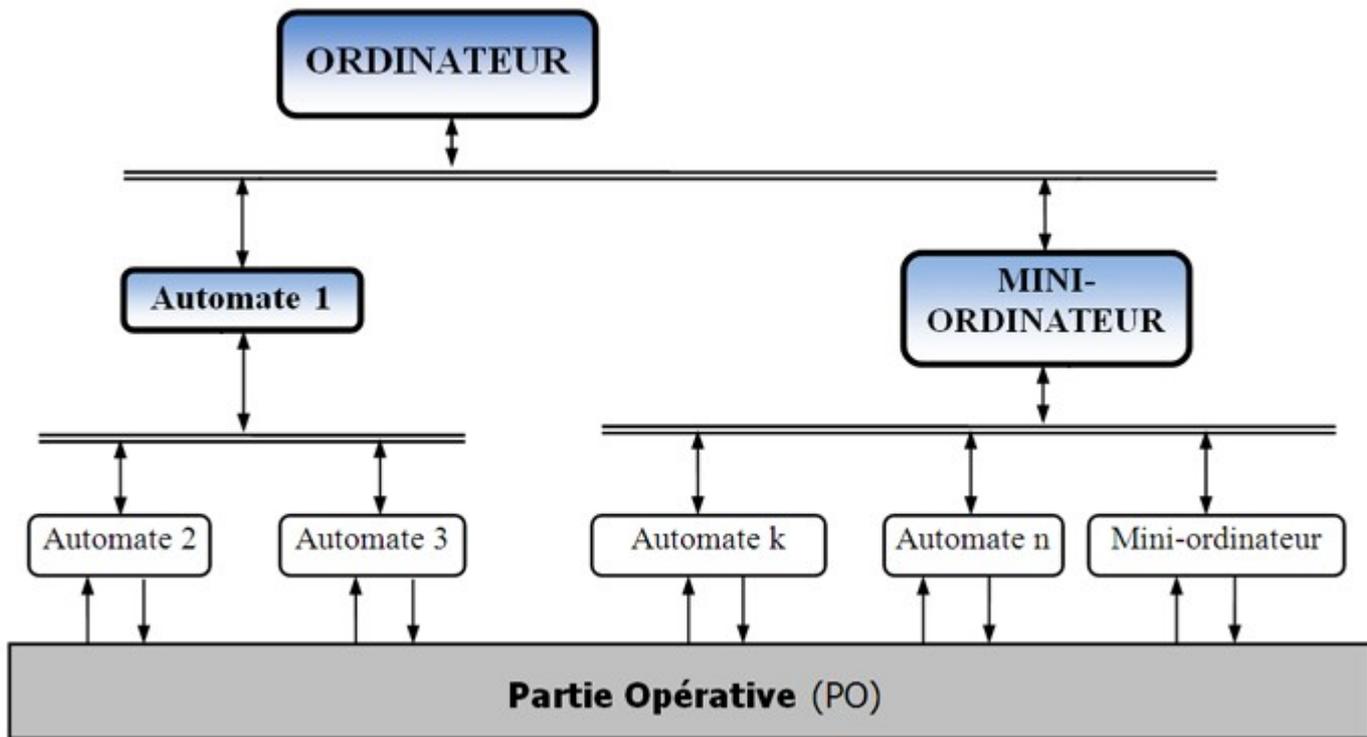
- Signal régénéré donc fiable.
- Contrôle facile des échanges (le message revient à l'émetteur).

Inconvénients :

- Chaque station est bloquante.
- Une extension interrompt momentanément le réseau.

8.3.3- Réseau hiérarchisé :

C'est la forme de réseaux la plus performante. Elle offre une grande souplesse d'utilisation, les informations pouvant circuler entre-stations d'un même niveau ou circuler de la station la plus évoluée (en général un ordinateur) vers la plus simple, et réciproquement (Figure 4.17).



Réseau hiérarchisé