

CHAPITRE I

RÉSEAUX LOCAUX INDUSTRIELS

1. Les automates programmables

Les API, automates programmables industriels, aussi appelés PLC, *Programmable Logic Controllers*, sont apparus dans les années 60 pour répondre à plusieurs besoins :

- ⇔ concentrer les entrées/sorties d'un système industriel pour faciliter sa gestion et son câblage. Les ordinateurs étaient assez complexes, assez volumineux et assez coûteux.
- ⇔ supporter les conditions des milieux industriels comme le bruit électrique, la poussière, la température, l'humidité, etc. Les ordinateurs de l'époque exigeaient un environnement particulier et ne répondaient pas à cet impératif.

Supposons, par exemple, que quand un contact se ferme on doit alimenter une bobine pendant 5 s et ensuite couper cette alimentation. Cela exige un timer. Avec 10 contacts et 10 bobines, 10 timers sont nécessaires. Si en plus on doit compter le nombre de fermetures des contacts, la tâche se complique et le besoin d'un automate apparaît clairement. Les années 70 ont vu l'apparition de MODBUS, premier protocole de communication entre API. Les fabricants d'ordinateurs, avec l'apparition des PCs ont tâché de combler leur retard et les années 80 ont vu émerger une convergence entre APIs et PCs. Malgré cette tendance, les APIs restent assez populaires dans le milieu industriel.

1.1. Qu'est-ce qu'un automate programmable

Un automate programmable est un équipement qui remplace l'ensemble de circuits de contrôle d'une machine. Il lit les entrées et en fonction de leur état ouvre ou ferme les contacts de sortie de la machine. L'utilisateur y entre un programme, généralement par logiciel, pour définir les résultats voulus. Il est constitué essentiellement d'une unité centrale, de mémoire et de circuits pour gérer les entrées et les sorties. On peut considérer un automate programmable comme une boîte remplie de centaines ou de milliers de relais, compteurs, temporisateurs et de mémoire de stockage. Chaque relais est simulé par un bit de la mémoire de l'API.

Les relais d'entrée sont connectés à des contacts, capteurs, etc. Souvent ce ne sont pas des relais réels, mais des transistors qui assurent la même fonction. Les relais de sortie ou bobine (coils), envoient des signaux d'activation ou d'arrêt aux bobines, lumières, etc. Ils peuvent être des vrais relais, des transistors ou triacs.

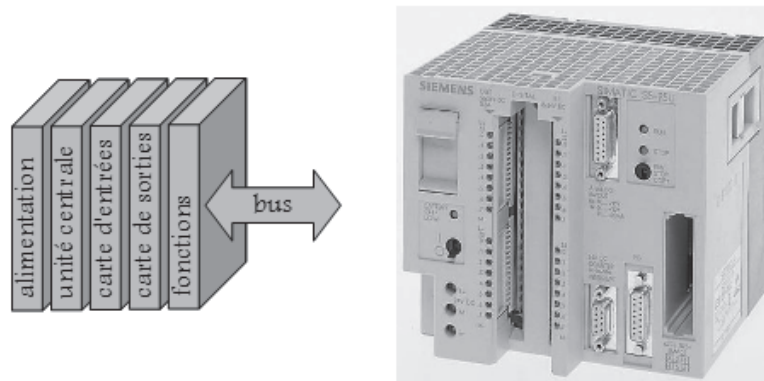


figure I.1 : structure d'un automate programmable

Un automate programmable exécute continuellement son programme qui peut se résumer en 3 phases données par le diagramme suivant :

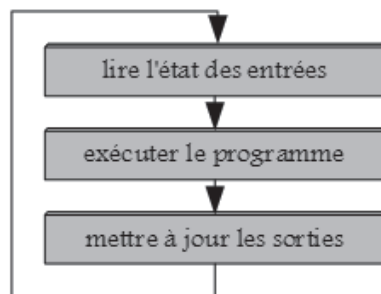


figure I.2 : cycle de traitement séquentiel d'un API

L'automate lit l'état de tous les capteurs et enregistre le résultat dans sa mémoire, ensuite il exécute le programme par lequel l'utilisateur a consigné l'opération qui, en fonction des capteurs, doit être accomplie sur chaque sortie. Finalement il écrit les données de mise à jour sur les sorties et recommence son cycle. Le cycle est aussi appelé scrutation. Le temps de réponse d'un API est un paramètre important, car certaines applications industrielles y sont très sensibles. Il peut être schématisé de la façon suivante :

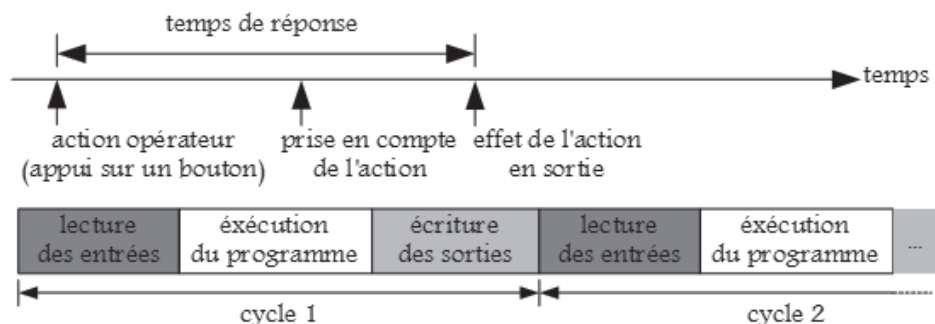


figure I.3 : temps de réponse d'un automate programmable

L'automate ne peut lire l'état des entrées que pendant la phase de lecture même si l'action de l'opérateur a eu lieu avant. De même pour l'exécution du programme et de la mise à jour des sorties. Tout cela contribue au temps de réponse.

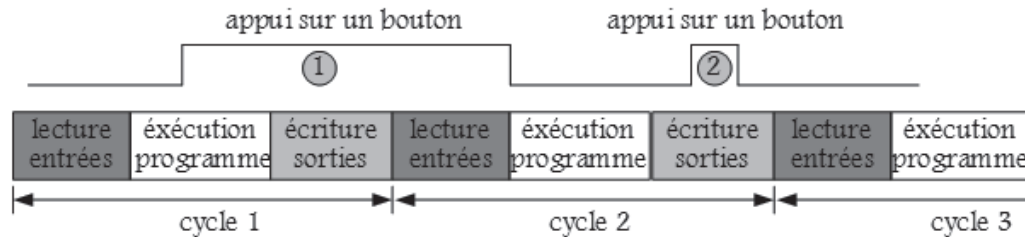


figure I.4 : exemple d'action et de sa prise en compte par l'API

Dans l'exemple ci-dessus, l'entrée 1 (appui sur un bouton) ne sera vue par l'automate que pendant la phase de lecture du deuxième cycle, alors qu'elle a commencé bien avant. Pire encore, l'entrée 2 ne sera jamais vue, car elle commence et se termine en dehors d'une phase de lecture.

Pour éviter ceci, les automates programmables procèdent de deux façons. Ils peuvent utiliser une fonction d'extension, *Pulse stretch function*, qui étend la durée de l'entrée jusqu'à la phase de lecture du cycle suivant. Ils peuvent aussi procéder par interruption, auquel cas, dès qu'une entrée est présente, l'automate exécute une routine utilisateur qui prend en charge l'entrée et ceci quelle que soit la phase du cycle.

Si on considère une action opérateur (appui sur un bouton) qui arrive juste après la phase de lecture, sa prise en compte sera effective lors de la prochaine phase de lecture et l'action en sortie qui en découle sera réalisée lors de la phase d'écriture suivante. Il en découle un délai maximum de 2 cycles ou un temps de lecture.

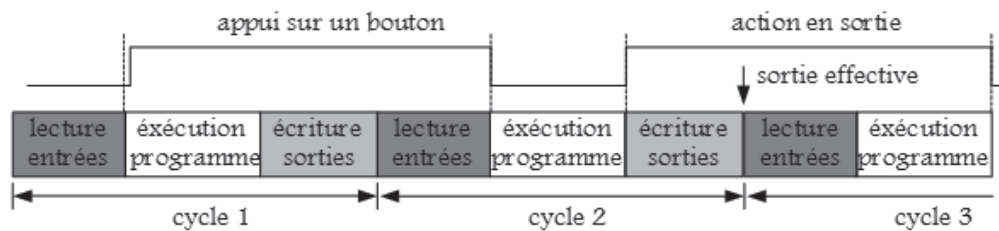


figure I.5 : délai maximum de réponse

1.2. Capteur

Un capteur est un dispositif transformant l'état d'une grandeur physique observée en une grandeur utilisable comme une tension électrique, une hauteur de mercure, une intensité, la déviation d'une aiguille, etc.

Le capteur se distingue de l'instrument de mesure par le fait qu'il ne s'agit que d'une simple interface entre un processus physique et une information manipulable. Par opposition, l'instrument de mesure est un appareil autonome se suffisant à lui-même, disposant d'un affichage ou d'un système de stockage des données. Le capteur, lui, en est dépourvu. Les capteurs sont les éléments de base des systèmes d'acquisition de données.

1.3. Actionneur

Dans une machine ou un système de commande à distance, semi-automatique ou automatique, un actionneur est l'organe qui reçoit un ordre de la partie commande sous forme d'énergie et qui convertit cette énergie en un travail utile à l'exécution de tâches du système. En d'autres termes, un actionneur est l'organe fournissant la force nécessaire à l'exécution d'un travail ordonné par une unité de commande distante.

1.4. Relais

Un relais est un commutateur électromagnétique qui agit comme un interrupteur fermé ou ouvert. Quand une tension est appliquée aux bornes de sa bobine, un champ électromagnétique apparaît et pousse les contacts du relais à se coller l'un contre l'autre formant ainsi un contact fermé. L'absence de courant dans la bobine permet d'ouvrir le contact.

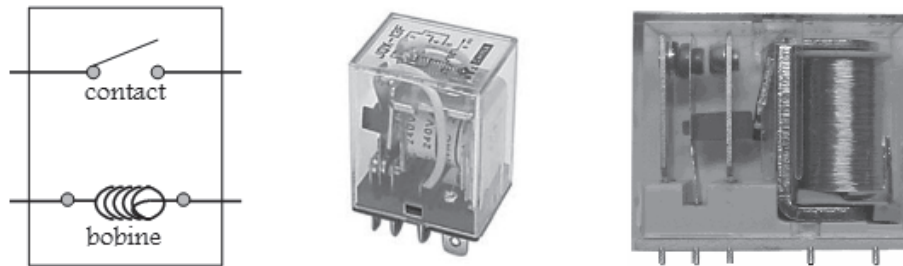


figure 1.6 : schéma d'un relais

1.5. Les modules d'entrées

Les APIs offrent plusieurs types de modules d'entrée, analogiques, numériques (digitaux) ou tout ou rien (logique discrète). Typiquement, un module d'entrées continues ou DC ou digitales est disponible en 5, 12, 24 et 48 volts. On peut y connecter des capteurs à transistor de type NPN (sinking) ou PNP (sourcing), selon que le courant doit être fourni ou absorbé par l'API.

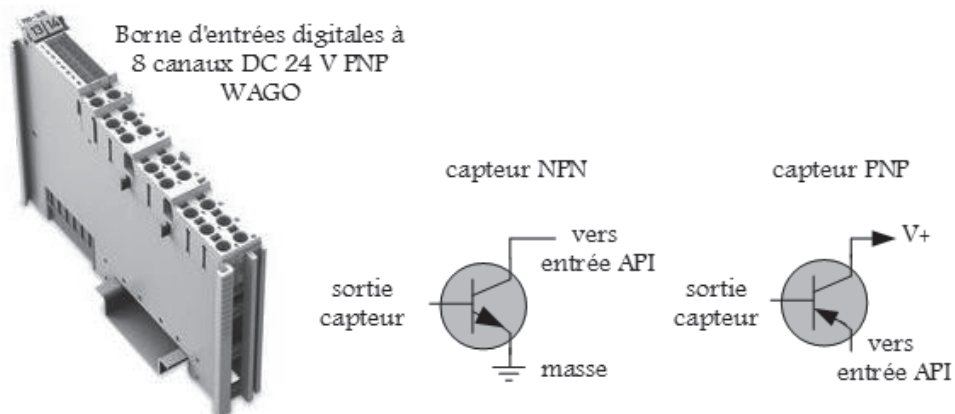


figure 1.7 : entrées continues d'un API

Typiquement un module d'entrées analogique ou AC est disponible en 24, 48, 110 et 220 volts. Il n'y a pas de problème de polarité comme pour les entrées digitales. Les fils sont généralement codés par leur couleur pour faciliter le câblage.

1.6. Les modules de sorties

De la même manière, les sorties peuvent être numériques, analogiques ou tout ou rien.



figure 1.8 : sorties digitales d'un API

2. Ancienne et nouvelle approche d'automatisation

Jusque dans les années 80, les automatismes s'appuyaient sur des automates programmables industriels (API) centralisés qui traitaient essentiellement des fonctions séquentielles et assuraient deux tâches principales :

- ⇒ Ils géraient des demandes d'exécution et d'état de l'automatisme (image des entrées);
- ⇒ et élaboraient des demandes d'exécution d'actions (positionnement des sorties).

Peu à peu, les API ont été amenés à gérer des fonctions complémentaires comme des fonctions métier, des fonctions de diagnostic système et application, etc.

Bien souvent, l'automatisation supplémentaire était réalisée avec le ou l'automate existant, même si elle n'avait aucun rapport avec celui-ci.

Ces automatismes centralisés amenaient alors de nombreuses contraintes :

- ⇒ les différents sous-ensembles n'étaient pas autonomes ;
- ⇒ la mise en service et la maintenance sont lourdes et difficiles à effectuer du fait de la quantité d'entrées/sorties gérées ;
- ⇒ Un défaut système de l'API ou son arrêt pour la maintenance d'outils de production, provoque l'arrêt de l'ensemble des fonctions gérées par l'API.

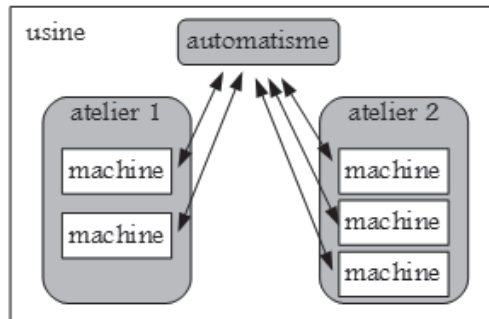


figure I.9 : ancienne approche, automatismes centralisés

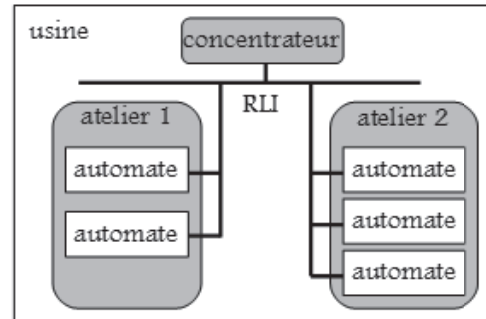


figure I.10 : nouvelle approche, automatismes décentralisés

Les contraintes des systèmes centralisés, ont poussé les utilisateurs vers une segmentation de l'architecture en découpant l'automatisme en entités fonctionnelles. Ceci a permis de simplifier les automatismes en réduisant le nombre d'E/S gérées et présente l'avantage de faciliter la mise en service et la maintenance. Cette segmentation a fait naître le besoin de communication entre les entités fonctionnelles, fonction qui est devenue la clef de voûte de la conception des architectures d'automatismes.

Les constructeurs d'API ont donc créé des offres de réseaux locaux industriels (RLI) afin d'assurer une communication efficace entre les différents API.

Un bus de terrain est un système d'interconnexion d'appareils de mesure, de capteurs, actionneurs, etc. Le terme bus de terrain est utilisé par opposition au bus informatique. En effet, le bus de terrain est en général beaucoup plus simple, du fait des faibles ressources numériques embarquées dans les capteurs et actionneurs industriels. Il est également plus robuste face aux perturbations externes. Un autre élément clef des bus de terrain est leur aspect déterministe et temps réel. L'élément le plus couramment lié à un bus de terrain est l'automate programmable industriel aussi appelé API. Les réseaux de terrain ou bus de terrain ou bus industriels permettent :

- ⇔ la connexion entre plusieurs entités d'un même système sur un même support de communication et cela dans une zone géographique limitée (atelier, automobile, électronique embarquée, etc) ;
- ⇔ le transport fiable de données sous une forme numérique de n'importe quel composant vers un autre ;
- ⇔ l'ajout ou la suppression d'éléments au sein d'un même système (réduction ou extension du réseau) ;
- ⇔ le travail en temps réel avec des protocoles de communication rapides.

3. Décentralisation des entrées/sorties

Pour faire baisser les coûts de câblage, il a été nécessaire de prendre en compte la topologie des automatismes. Sur des sites étendus, il est souvent nécessaire de gérer un nombre important de points distants et de prendre en compte les fonctions métier réparties, comme la variation de vitesse, le dialogue homme/machine, le pesage, etc. Les constructeurs de produits d'automatismes ont répondu avec les réseaux et bus de terrain. Cela a permis de gérer d'abord des E/S décentralisées et par la suite les périphériques d'automatisme. Les réseaux de terrain ont contribué à réaliser des gains de câblage importants, mais surtout ils ont rendu accessibles des services de diagnostic, de programmation, etc sur tout le site.

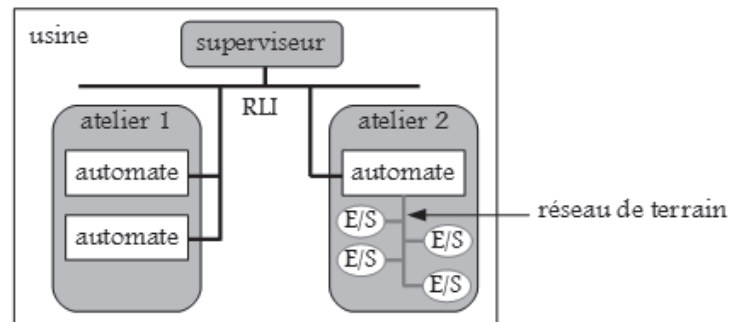


figure I.11 : entrées/sorties décentralisées

La figure suivante résume le cheminement qui a mené à l'utilisation poussée des bus de terrain. On est passé de l'utilisation du câblage fil à fil à l'utilisation du câblage en nappe pour aboutir à l'utilisation de bus de terrain.

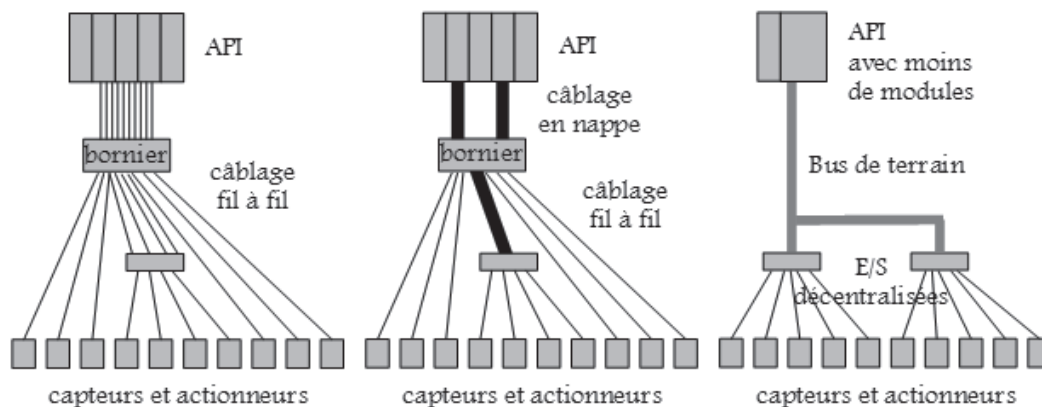


figure I.12 : évolution du câblage industriel

Actuellement un câblage simple et efficace est réalisé entièrement en bus jusqu'au niveau capteurs et actionneurs comme la figure suivante.

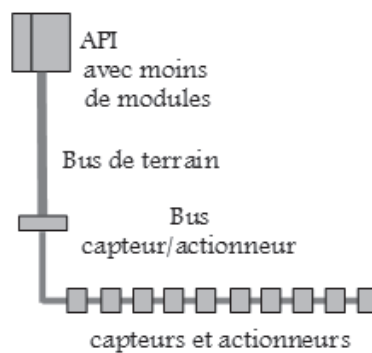


figure I.13 : câblage entièrement en bus

4. Pyramide CIM

Il convient tout d'abord de donner quelques définitions :

- ⇨ Terrain : zone limitée géographiquement, comme une usine, un atelier, un véhicule, etc ;
- ⇨ Bus : conducteur(s) commun(s) à plusieurs circuits permettant l'échange de données ;
- ⇨ Réseau : ensemble de lignes de communication qui desservent une même unité géographique.

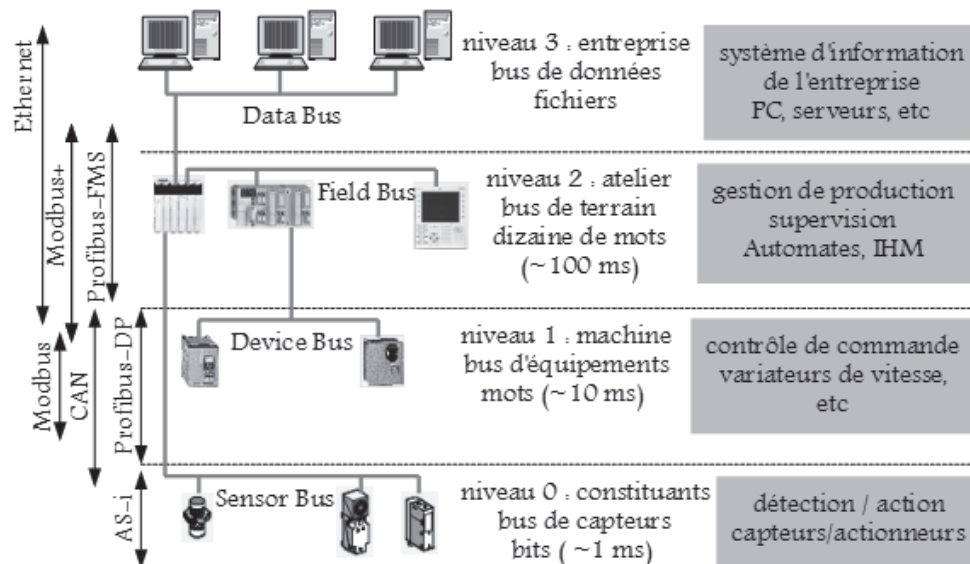


figure I.14 : classification et positionnement des principaux réseaux locaux

Le modèle CIM, *Computer Integrated Manufacturing*, se voulait être la réponse à la quête de performance en créant une segmentation verticale des réseaux et des bus. Le CIM décrit les différents niveaux de communication sous une forme quantitative des données à véhiculer. La pyramide du CIM représente 4 niveaux de décision dans l'entreprise. Le niveau 0, niveau capteur/actionneur, nécessite un transfert performant (quelques millisecondes) mais concerne peu d'informations (données binaires), alors que le niveau 4 nécessite quant à lui de véhiculer de gros volumes d'information.

Les constructeurs d'automates programmables ont créé des réseaux et des bus adaptés au besoin. Ainsi à chaque niveau, correspond un bus ou un réseau :

- ⇨ les *sensor bus*, bus capteurs et actionneurs unitaires simples ;
- ⇨ les *device bus*, bus et réseaux pour la périphérie d'automatisme : variateurs, robots, axes, etc ;
- ⇨ les *field bus*, bus de terrain ou réseaux de communication entre unités de traitement : automates programmables, superviseurs, commandes numériques, etc ;
- ⇨ les réseaux locaux industriels, pour l'établissement de la communication entre l'automatisme et le monde informatique.