William Bolton

AUTOMATES PROGRAMMABLES INDUSTRIELS

Traduction de Hervé Soulard

2e édition

DUNOD

This sixth edition of the work entitled
PROGRAMMABLE LOGIC CONTROLLERS
[ISBN 9780128029299] by William BOLTON
is published with arrangement with ELSEVIER LIMITED
of The Boulevard, Langford Lane, Kidlington, Oxford, OX 1GB, UK.

Cet ouvrage est la traduction en langue française, par les Éditions Dunod,
de *Programmable Logic Controllers, sixth edition*
de William Bolton

Illustration de couverture : © GraphicObsession - Fotolia.com

11, rue Paul Bert, 92240 Malakoff

ISBN 978-2-10-080615-7

TABLE DES MATIÈRES

PRÉFACE

Ces dernières années, les avancées technologiques ont conduit au développement des automates programmables industriels (API) et à une révolution importante dans l'automatique. Cet ouvrage, une introduction aux API, a pour objectif de faciliter le travail des ingénieurs praticiens qui font leurs premiers pas dans le domaine des automates programmables industriels. Il servira également de support de cours aux étudiants des écoles d'ingénieurs.

Ce livre s'intéresse aux problèmes liés à l'utilisation de nomenclatures et de programmes variés par les différents fabricants. Pour cela, il décrit les principes sous-jacents et les illustre par des exemples provenant de plusieurs fabricants. Voici les thèmes abordés :

– l'architecture de base des API et les caractéristiques des entrées-sorties souvent employées avec ces systèmes ;
– la présentation des systèmes de numération : décimal, binaire, octal, hexadécimal et BCD ;
– une introduction méthodique et minutieuse, avec de nombreuses illustrations, qui décrit la programmation des API, quel que soit le fabricant, et comment utiliser les relais, les temporisateurs, les compteurs, les registres à décalage, les séquenceurs et les fonctions de gestion des données ;
– un examen de la norme CEI 61131-3 et les méthodes de programmation, comme le langage à contacts (*Ladder*), les diagrammes de schémas fonctionnels, les listes d'instructions, le texte structuré et les graphes de fonction séquentielle ;
– de nombreux exemples, des questions à choix multiples et des problèmes pour aider le lecteur à développer les connaissances nécessaires à l'écriture de programmes pour des API (les réponses aux questions et aux problèmes sont données à la fin de l'ouvrage).

Connaissances requises

Le lecteur n'est pas supposé posséder une expérience en informatique. Toutefois, il est préférable qu'il ait des connaissances de base en électricité et en électronique.

Changements par rapport à la précédente édition

Cette deuxième édition en français est la traduction de la sixième édition anglaise de *Programmable Logic Controllers* de William Bolton. Cette nouvelle édition actualisée a été complétée et enrichie de nouveaux cas d'étude. Dans la précédente édition ont été ajoutés des explications, des exemples et des problèmes, ainsi qu'un résumé des points essentiels à la fin de chaque chapitre. Dans cette nouvelle édition, le Chapitre 1 a été revu afin de comparer les systèmes de commande à relais, à microprocesseur et à API. Par ailleurs, les API commerciaux décrits ont été actualisés, et la présentation des méthodes de programmation des API établies par la norme CEI-61131 a été développée. Au Chapitre 2, nous avons développé le contenu sur les capteurs, avons réécrit la présentation des graphes de fonction séquentielle au Chapitre 6 afin de détailler plus en profondeur cette méthode. La partie du Chapitre 10 qui concerne le séquenceur a été révisée. Le principe du forçage expliqué au Chapitre 13 a été développé, et de nouveaux cas d'études ont été ajoutés au Chapitre 14.

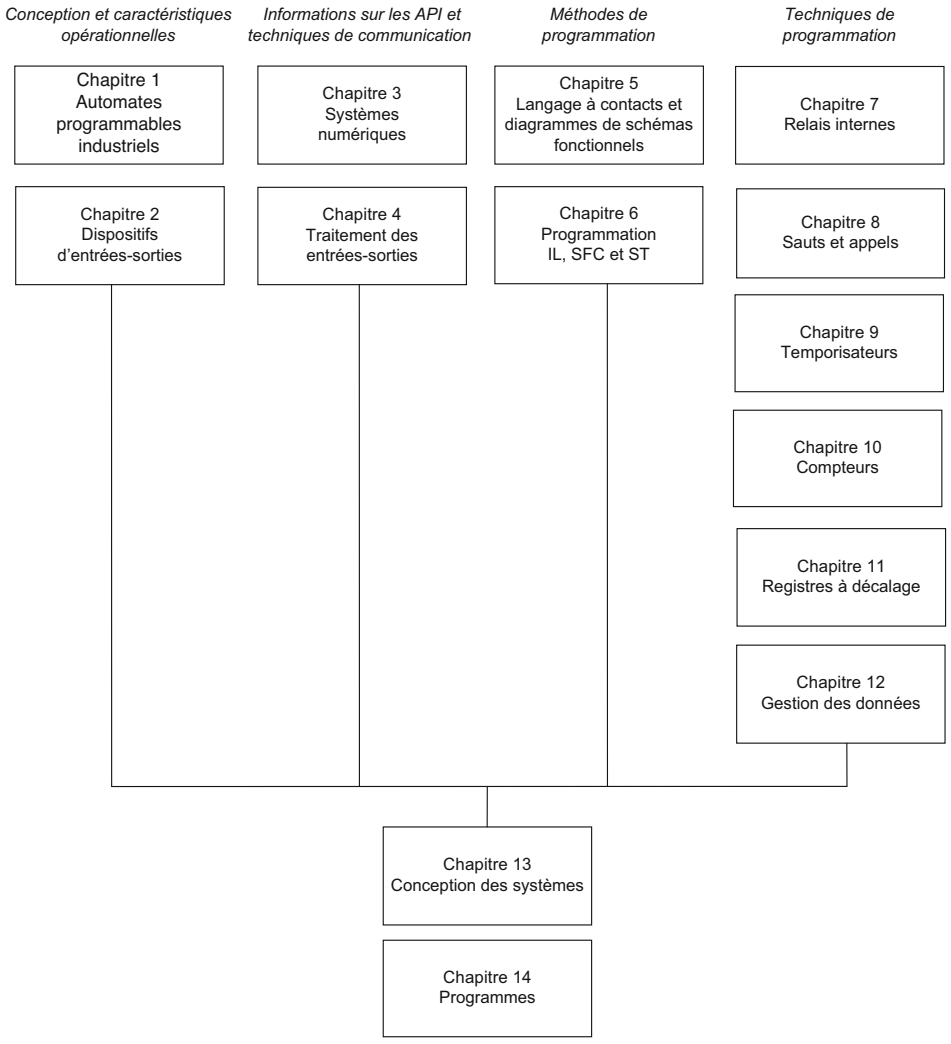
Objectifs du livre

Cet ouvrage a pour objectif d'aider le lecteur à développer les capacités suivantes :

- identifier et expliquer les principales caractéristiques de conception, l'architecture interne et les principes de fonctionnement des automates programmables industriels ;
- utiliser des API de différentes tailles et de différents fabricants ;
- employer les dispositifs d'entrée et de sortie couramment utilisés avec les API, en tenant compte de leurs caractéristiques ;
- expliquer le traitement des entrées et des sorties par les API afin que les systèmes d'entrées-sorties soient correctement utilisés ;
- décrire les liens de communication impliqués dans les systèmes API, en reconnaissant les protocoles et les méthodes de mise en réseau ;
- développer des applications à partir de programmes à contacts fondés sur des relais, des temporisateurs, des compteurs, des registres à décalage, des séquenceurs et des fonctions de gestion des données ;
- identifier les problèmes de sécurité dans les systèmes API ;
- identifier les méthodes employées pour le diagnostic des défauts, les tests et le débogage.

Organisation du livre

La figure de la page suivante illustre la structure de cet ouvrage.



Remerciements

Je remercie les nombreux relecteurs des différentes éditions de cet ouvrage pour leurs remarques et leurs commentaires.

– W. Bolton

1 • AUTOMATES PROGRAMMABLES INDUSTRIELS

Ce chapitre constitue une introduction aux automates programmables industriels (API), ainsi qu'à leur fonction générale, leurs formes matérielles et leur architecture interne. Les API sont employés dans de nombreuses tâches d'automatisation, dans différents domaines, comme les processus de fabrication industriels. Cette vue d'ensemble sera complétée par des explications plus détaillées dans les chapitres suivants. Vous trouverez une description résumée des automates programmables industriels sur la page Wikipédia qui leur est consacrée.

1.1 Systèmes de commande

Quelles tâches un système de commande peut-il prendre en charge ? Il peut être chargé de contrôler une séquence d'événements, de maintenir constante une certaine variable ou de suivre un changement prévu. Par exemple, le système de commande d'une perceuse automatique (voir Figure 1.1a) peut démarrer la descente du foret lorsque la pièce est en position, démarrer le perçage lorsque le foret arrive sur la pièce, stopper le perçage lorsque la profondeur du trou souhaitée est atteinte, retirer le foret, arrêter la perceuse, puis attendre que la pièce suivante arrive en position avant de répéter le processus. Un autre système de commande (voir Figure 1.1b) pourrait être utilisé pour vérifier le nombre d'articles convoyés par une bande transporteuse et les diriger vers une caisse. Les entrées de ces systèmes de commande peuvent provenir d'interrupteurs, qui sont ouverts ou fermés. Par exemple, la présence de la pièce peut être signalée par son contact avec un interrupteur, qui se ferme, ou par d'autres capteurs, par exemple de température ou de débit. Le contrôleur peut être chargé de commander un moteur pour déplacer un objet vers un certain emplacement ou pour tourner une vanne, ou encore d'allumer ou d'éteindre un radiateur.

Quelles formes un système de commande peut-il prendre ? Pour la perceuse automatique, il est possible de câbler des circuits électriques dans lesquels l'ouverture ou la fermeture de l'interrupteur conduisent au démarrage de moteurs ou au déclenchement de vannes. En conséquence, les contacts d'un relais peuvent ainsi être fermés ou ouverts (voir Figure 1.2), ce qui met un moteur sous tension et déclenche la rotation du foret (voir Figure 1.3). Un autre interrupteur peut être utilisé pour

activer un relais et mettre sous tension une vanne pneumatique ou hydraulique, qui met sous pression un vérin afin de pousser la pièce dans la position requise. De tels circuits électriques sont spécifiques à la perceuse automatique. Pour contrôler le nombre d'articles placés dans un carton, nous pouvons également concevoir des circuits électriques qui mettent en œuvre des capteurs et des moteurs. Cependant, le circuit de chacun de ces systèmes de commande sera différent. Dans un système de commande « traditionnel », les règles de fonctionnement et les actions exécutées sont déterminées par le câblage. Lorsque les règles qui déterminent les actions de commande sont modifiées, le câblage doit également être revu.

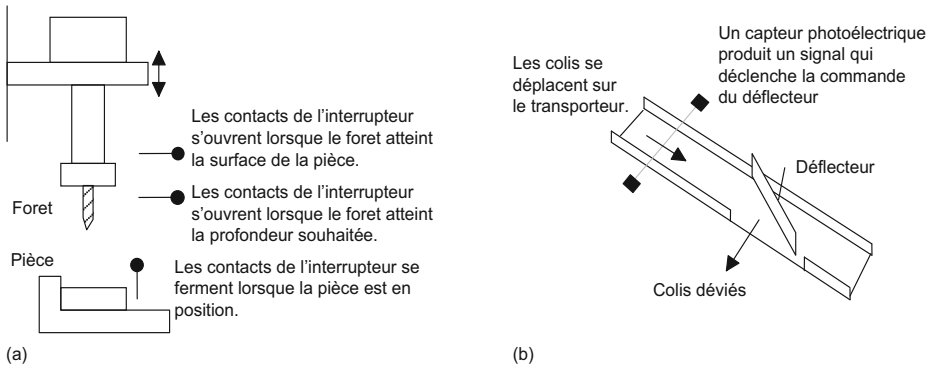


Figure 1.1 – Exemples de tâches de commande et de capteurs d'entrée : (a) une perceuse automatique et (b) un système de conditionnement.

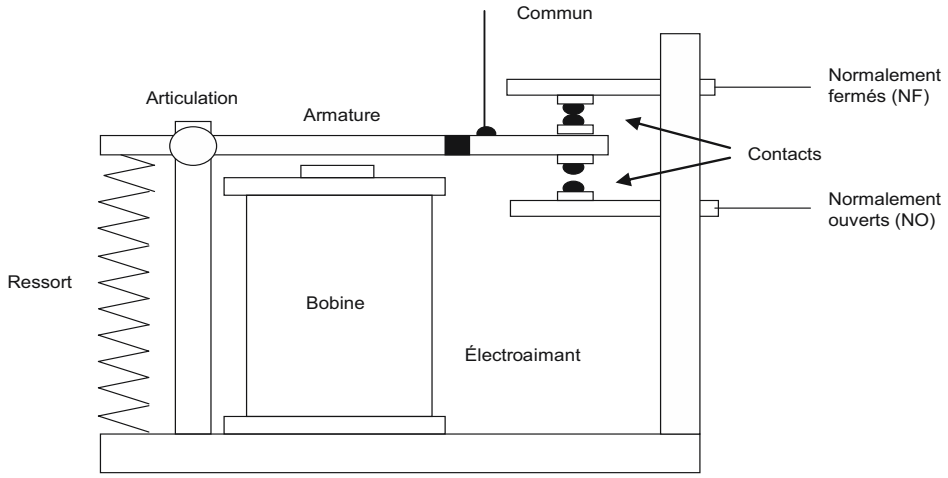


Figure 1.2 – Principe de base d'un relais.

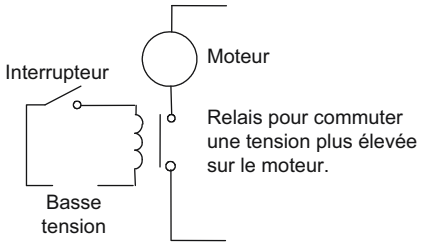


Figure 1.3 – Un circuit de commande.

1.1.1 Systèmes de commande à relais

Les systèmes de commande à relais sont des systèmes câblés. La Figure 1.2 illustre les constituants d'un relais simple. Lorsqu'un courant traverse la bobine du relais, les contacts normalement fermés (NF) s'ouvrent et les contacts normalement ouverts (NO) se ferment. Ces contacts peuvent servir à commander un système. Prenons par exemple un relais utilisé pour déclencher une vanne pneumatique ou hydraulique, ce qui conduit à l'application d'une pression à un piston pour déplacer une pièce. Il est possible de représenter ce fonctionnement par un schéma de commande. La Figure 1.4 donne les symboles standard pour la représentation des relais. Dans le schéma de commande illustré à la Figure 1.5, les lignes verticales représentent les sources d'alimentation tandis que les lignes horizontales relient les systèmes entre eux. La séquence des événements se lit en partant de la ligne horizontale supérieure et en allant vers le bas. Ainsi, dans la ligne supérieure de la Figure 1.5a, lorsque l'interrupteur Marche-arrêt est fermé, le relais est activé. Cela déclenche la fermeture des contacts représentés sur la seconde ligne et, par conséquent, l'alimentation de l'électrovanne. Un schéma plus classique est donné à la Figure 1.5b. Dans ce cas, le relais est activé par un bouton-poussoir normalement ouvert à appui momentané. Son action ferme deux jeux de contacts. Les contacts 1 mémorisent l'appui sur le bouton-poussoir afin que son relâchement ne coupe pas l'alimentation du relais. Les contacts 2 alimentent l'électrovanne. Le relais et, par conséquent, l'alimentation de l'électrovanne, sont coupés lors de l'appui sur le bouton-poussoir normalement fermé. Ces schémas ne représentent évidemment qu'une partie du système de commande car il faudrait d'autres lignes pour détecter le moment où l'électrovanne a déplacé la pièce de la distance requise et pour stopper son fonctionnement.

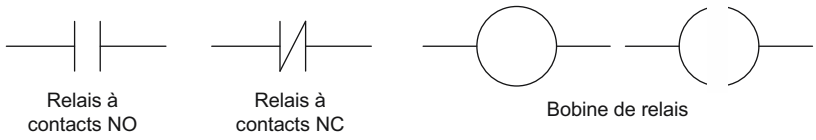


Figure 1.4 – Représentations symboliques d'un relais.

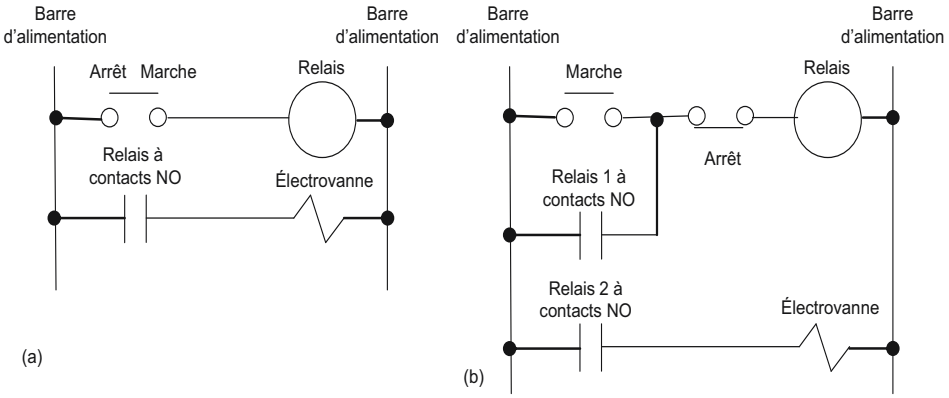


Figure 1.5 – Schémas de systèmes de commande à relais.

La Figure 1.6 montre un autre exemple de système de commande à relais. Lorsque le bouton-poussoir Marche est fermé, la bobine du relais est alimentée et mémorise l'actionnement du bouton afin que le relais reste activé jusqu'à l'appui sur le bouton Arrêt. Les contacts NO du relais se ferment et ses contacts NF s'ouvrent. La lampe verte s'allume, contrairement à la lampe rouge qui s'éteint. Suite à l'appui sur le bouton-poussoir Arrêt, le courant dans la bobine du relais est coupé. Les contacts NO s'ouvrent alors, tandis que les contacts NF se ferment ; la lampe verte s'éteint et la lampe rouge s'allume. L'étape suivante du schéma pourrait impliquer la mise en marche d'un moteur par les contacts NO. Dans ce cas, la lampe verte indiquerait que le moteur tourne et la lampe rouge qu'il est arrêté.

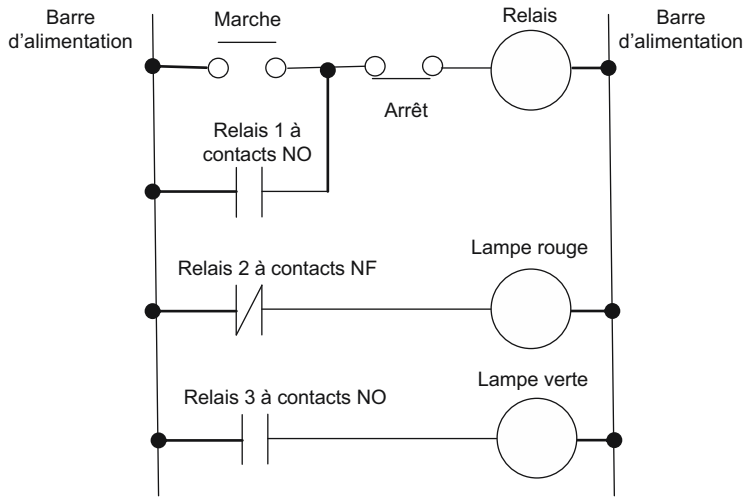


Figure 1.6 – Circuit à relais pour la commande de lampes rouge et verte.

1.1.2 Systèmes de commande à microprocesseur

Au lieu de câbler un circuit de commande différent pour chaque situation, une approche permet d'employer le même système de base dans tous les cas. Elle se fonde sur un système à base de microprocesseur et sur un programme qui explique à ce microprocesseur comment réagir aux signaux d'entrée, par exemple issus d'interrupteurs, et comment produire les sorties requises, par exemple sur des moteurs ou des vannes. Ainsi, nous pourrions avoir un programme de la forme suivante :

```
Si l'interrupteur A se ferme
Activer la sortie du circuit du moteur
Si l'interrupteur B se ferme
Activer la sortie du circuit de la vanne
```

En modifiant les instructions du programme, nous pouvons conserver le même système à microprocesseur pour commander une grande diversité de cas.

Prenons l'exemple d'une machine à laver le linge moderne à base de microprocesseur. Ses entrées proviennent des cadrans de sélection du programme de lavage, d'un interrupteur qui indique la fermeture de la porte, d'un capteur qui détermine la température de l'eau et d'un interrupteur qui détecte le niveau d'eau. Le programme du microprocesseur utilise ces entrées pour positionner les sorties de manière à démarrer le moteur du tambour et contrôler sa vitesse, à ouvrir et à fermer les vannes d'eau chaude et d'eau froide, à mettre en marche la pompe de vidange, à commander la résistance de chauffage et à contrôler le verrou de la porte pour que la machine ne puisse pas être ouverte avant la fin du cycle.

1.1.3 Automates programmables industriels

Un *automate programmable industriel* (API) est une forme particulière de contrôleur à microprocesseur qui utilise une mémoire programmable pour stocker les instructions et qui implémente différentes fonctions, qu'elles soient logiques, de séquencement, de temporisation, de comptage ou arithmétiques, pour commander les machines et les processus (voir Figure 1.7). Il est conçu pour être exploité par des ingénieurs, dont les connaissances en informatique et langages de programmation peuvent être limitées. La création et la modification des programmes de l'API ne sont pas réservées aux seuls informaticiens. Les concepteurs de l'API l'ont préprogrammé pour que la saisie du programme de commande puisse se faire à l'aide d'un langage simple et intuitif (voir Chapitre 4). La programmation de l'API concerne principalement la mise en œuvre d'opérations logiques et de commutation, par exemple, si A *ou* B se produit, alors allumer C, ou si A *et* B se produisent, alors allumer D. Les dispositifs d'entrée, c'est-à-dire des capteurs, comme des interrupteurs, et les dispositifs de sortie, c'est-à-dire des moteurs, des vannes, etc., du système sont connectés à l'API. L'opérateur saisit une séquence d'instructions, le programme, dans la mémoire de l'API. L'automate surveille ensuite les entrées et les sorties conformément aux instructions du programme et met en œuvre les règles de commande définies.

Les API présentent un avantage majeur : le même automate de base peut être employé avec une grande diversité de systèmes de commande. Pour modifier un

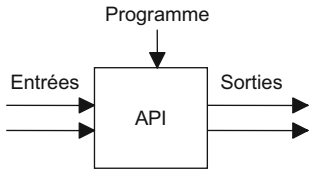


Figure 1.7 – Un automate programmable industriel.

système de commande et les règles appliquées, un opérateur doit simplement saisir une suite d'instructions différente. On obtient ainsi un système flexible et économique, utilisable avec des systèmes de commande dont la nature et la complexité peuvent varier énormément. En comparaison des systèmes à relais, les API :

– simplifient les modifications car elles sont mises en œuvre de façon logicielle et non pas par des solutions matérielles plus complexes ;
– peuvent être facilement étendus par l'ajout de nouveaux modules, alors que des changements matériels sont sinon requis ;
– sont plus robustes et plus fiables en raison d'un nombre de composants mécaniques moindre ;
– sont plus compacts ;
– exigent une maintenance moindre ;
– sont plus rapides.

Les API sont comparables aux ordinateurs. Toutefois, alors que les ordinateurs sont optimisés pour les tâches de calcul et d'affichage, les API le sont pour les tâches de commande et les environnements industriels. Voici ce qui différencie les API par rapport aux ordinateurs :

– Ils sont solides et conçus pour supporter les vibrations, les températures basses ou élevées, l'humidité et le bruit. *A contrario*, les ordinateurs personnels ne sont pas faits pour opérer dans des milieux hostiles.
– Les interfaces des entrées et des sorties sont intégrées à l'automate. Les API au format modulaire peuvent être facilement étendus pour recevoir un plus grand nombre d'entrées-sorties.
– Ils sont simples à programmer et leur langage de programmation d'apprentissage facile est principalement orienté sur les opérations logiques et de commutation. Ils sont par conséquent plus conviviaux.
– Ils sont moins adaptés au stockage à long terme et à l'analyse des données.
– Ils sont d'une fiabilité supérieure et moins sujets aux dysfonctionnements que les ordinateurs personnels.

Les premiers API ont été conçus en 1969 ; ils sont à présent largement utilisés. Ils prennent la forme de petites unités autonomes pour environ vingt entrées-sorties numériques ou de systèmes modulaires qui peuvent être employés pour des entrées-sorties très nombreuses, analogiques ou numériques, et qui disposent des

modes de régulation PID (proportionnel, intégral et dérivé). On les retrouve dans l'automatisation des processus industriels, comme l'usinage, la manutention, l'assemblage automatique et le conditionnement. Cependant, dans le cas des tâches d'automatisation très simples, par exemple pour un lave-linge domestique, une autre solution moins coûteuse sera probablement retenue. Lorsque les contraintes sont fortes, par exemple dans le cas du contrôle du vol d'un avion, un ordinateur aura la préférence car il est capable d'effectuer des opérations mathématiques complexes avec une vitesse de traitement élevée.

1.2 Matériel

De manière générale, un API est structuré autour de plusieurs éléments de base que sont l'unité de traitement, la mémoire, l'unité d'alimentation, les interfaces d'entrées-sorties, l'interface de communication et le périphérique de programmation (voir Figure 1.8) :

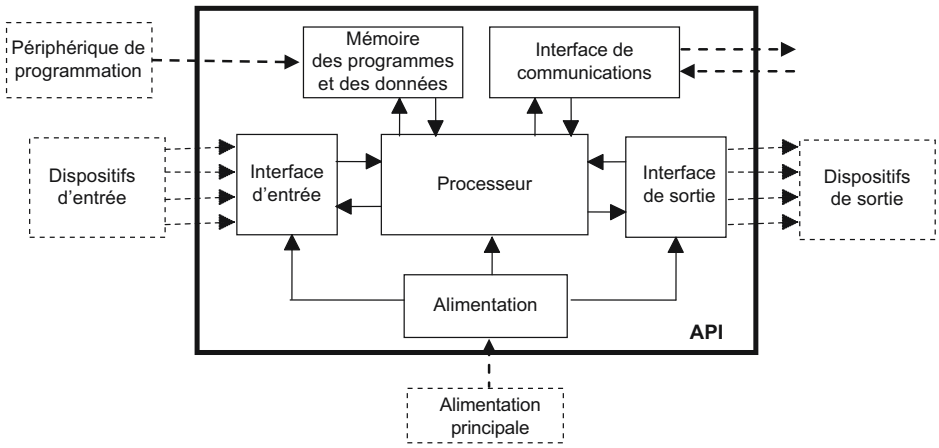


Figure 1.8 – Structure d'un API.

– Le *processeur* ou *unité centrale de traitement* (CPU, *Central Processing Unit*) contient le microprocesseur. Le CPU interprète les signaux d'entrée et effectue les actions de commande conformément au programme stocké en mémoire, en communiquant aux sorties les décisions sous forme de signaux d'action.
– L'*unité d'alimentation* est indispensable puisqu'elle convertit une tension alternative en une basse tension continue (5 V) nécessaire au processeur et aux modules d'entrées-sorties.
– Le *périphérique de programmation* est utilisé pour entrer le programme dans la mémoire du processeur. Ce programme est développé sur le périphérique, puis transféré dans la mémoire de l'API.

– La *mémoire* contient le programme qui définit les actions de commande effectuées par le microprocesseur. Elle contient également les données qui proviennent des entrées en vue de leur traitement, ainsi que celles des sorties.

– Les *interfaces d'entrées-sorties* permettent au processeur de recevoir et d'envoyer des informations aux dispositifs extérieurs. Les entrées peuvent être des interrupteurs, comme dans le cas de la perceuse automatique (voir Figure 1.1a), ou d'autres capteurs, comme des cellules photoélectriques dans le cas du mécanisme de comptage (voir Figure 1.1b), des sondes de température, des débitmètres, etc. Les sorties peuvent être des bobines de moteur, des électrovannes, etc. Nous reviendrons sur les interfaces d'entrées-sorties au Chapitre 2. Les dispositifs d'entrées-sorties peuvent être classés en trois catégories, selon qu'ils produisent des signaux discrets, numériques ou analogiques (voir Figure 1.9). Les dispositifs qui génèrent des *signaux discrets* ou *numériques* sont ceux dont les sorties sont de type tout ou rien. Par conséquent, un interrupteur est un dispositif qui produit un signal discret : présence ou absence de tension. Les dispositifs *numériques* peuvent être vus comme des dispositifs discrets qui produisent une suite de signaux tout ou rien. Les dispositifs *analogiques* créent des signaux dont l'amplitude est proportionnelle à la grandeur de la variable surveillée. Par exemple, un capteur de température peut produire une tension proportionnelle à la température.

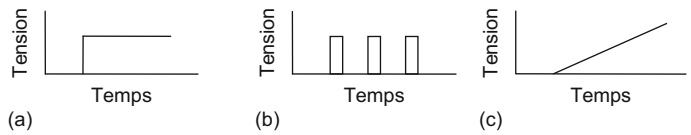


Figure 1.9 – Les signaux : (a) discrets, (b) numériques et (c) analogiques.

– L'*interface de communication* est utilisée pour recevoir et transmettre des données sur des réseaux de communication qui relient l'API à d'autres API distants (voir Figure 1.10). Elle est impliquée dans des opérations telles que la vérification d'un périphérique, l'acquisition de données, la synchronisation entres des applications et la gestion de la connexion.

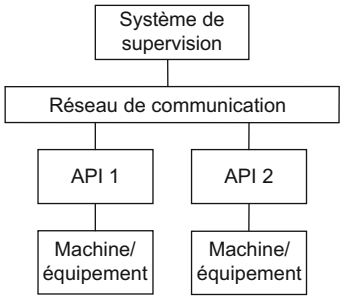


Figure 1.10 – Modèle de base des communications.

1.3 Architecture d'un API

Un API est généralement constitué d'une unité centrale de traitement (CPU, *Central Processing Unit*) qui comprend le microprocesseur, la mémoire et les entrées-sorties du système. Il peut en réalité être vu comme une entité composée d'un grand nombre de relais, compteurs, temporisateurs et unités de stockage de données, même si ces éléments n'existent pas physiquement dans l'API et sont simulés de façon logicielle.

La capacité de stockage d'une unité de mémoire est déterminée par le nombre de mots qu'elle peut enregistrer. Si la taille d'une mémoire est de 256 mots, elle peut stocker $256 \times 8 = 2\,048$ bits, pour des mots de huit bits, et $256 \times 16 = 4\,096$ bits, pour des mots de seize bits. Le terme *octet* (*byte*) est utilisé pour désigner un mot de huit bits. La taille de la mémoire est souvent indiquée en fonction du nombre d'emplacements disponibles. 1 K représente le nombre 2^{10}, c'est-à-dire 1 024. Une mémoire de 4 K octets peut donc enregistrer 4 096 octets, et une mémoire de 50 K octets en stockera 51 200.

1.3.1 Unité d'entrées-sorties

L'unité d'entrées-sorties (E/S) d'un API apporte le circuit d'interface entre le système et le monde extérieur. Au travers de canaux d'entrées-sorties, elle permet d'établir des connexions avec des dispositifs d'entrée, comme des capteurs, et des dispositifs de sortie, comme des moteurs et des solénoïdes. C'est également par l'intermédiaire de cette unité que se fait la saisie des programmes depuis un terminal. Chaque point d'entrée-sortie dispose d'une adresse unique, que le CPU peut utiliser. Le principe est comparable à une rangée de maisons le long d'une rue ; le numéro 10 peut correspondre à la « maison » pour l'entrée d'un capteur particulier, tandis que le numéro 45 peut correspondre à la « maison » utilisé pour la sortie vers un moteur spécifique.

Puisque les canaux d'entrées-sorties mettent en place les fonctions d'isolation et de traitement des signaux, il est possible de connecter directement des capteurs et des actionneurs aux canaux, sans passer par un autre circuit d'interface (voir Figure 1.11). L'isolation électrique avec le monde extérieur est généralement réalisée par des *photocoupleurs* (également appelés *optocoupleurs*), dont le principe est illustré à la Figure 1.12. Lorsque la diode électroluminescente (LED, *Light Emitting Diode*) est traversée par une impulsion numérique, elle produit un rayonnement infrarouge. Ce rayonnement est détecté par le phototransistor, qui fait naître une tension dans son circuit. L'espace qui sépare la LED et le phototransistor crée une isolation électrique, mais une impulsion numérique dans le premier circuit permet néanmoins de produire une impulsion numérique dans le second circuit.

Le traitement du signal réalisé au niveau du canal d'entrée, avec l'isolation, permet de manipuler une grande diversité de signaux d'entrée. Il est converti en une tension compatible avec celle requise par le microprocesseur qui équipe l'API (voir Chapitre 3). Un API élaboré peut ainsi accéder à des entrées dont les signaux

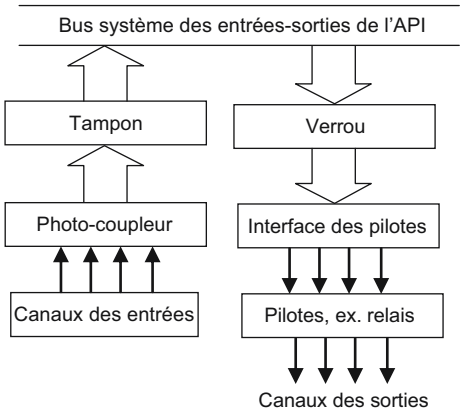


Figure 1.11 – Architecture des canaux d'entrées-sorties d'un API.

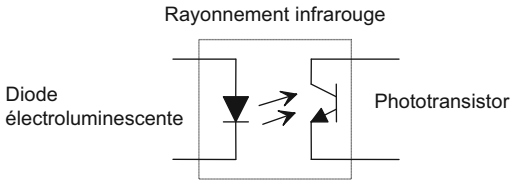


Figure 1.12 – Un photocoupleur.

numériques/discrets (c'est-à-dire tout ou rien) utilisent des tensions de 5 V, 24 V, 110 V et 240 V. Un API de base sera généralement en mesure d'utiliser une seule forme d'entrée, par exemple des signaux 24 V.

Les canaux de sortie permettent d'obtenir les sorties de l'API sous une forme adaptée à un connexion directe avec des circuits externes. Les sorties sont de type relais, transistor ou triac (voir Chapitre 3 pour plus de détails) :

– Avec le type *relais*, le signal issu d'une sortie de l'API est utilisé pour déclencher un relais et peut activer des courants de quelques ampères dans le circuit externe. Grâce au relais, non seulement des courants faibles peuvent commuter des courants forts, mais l'isolation entre l'API et le circuit externe est également assurée. En revanche, les relais sont relativement lents. Les sorties à relais sont adaptées à la commutation de courants alternatifs et continus. Ils peuvent supporter des surtensions transitoires et des courants de choc élevés.
– Avec le type *transistor*, la sortie se fonde sur un transistor pour commuter le courant dans le circuit externe. L'opération de commutation est donc extrêmement rapide. Toutefois, les sorties de ce type sont adaptées uniquement à la commutation d'un courant continu et sont détruites par les surintensités et les tensions inverses élevées. Pour leur protection, il faut utiliser un fusible ou un système électronique. Les photocoupleurs sont employés à des fins d'isolation.

– Avec le type *triac*, et des photocoupleurs pour l'isolation, les sorties peuvent servir à contrôler les charges externes connectées à une alimentation en courant alternatif. Elles prennent en charge uniquement les courants alternatifs et sont très facilement détruites par les surintensités. De manière générale, les sorties de ce type sont toujours protégées par des fusibles.

Ainsi, après le traitement du signal par des relais, des transistors ou des triacs, le canal de sortie est capable de fournir un signal 24 V et 100 mA, une tension continue de 110 V et 1 A, ou 240 V, une tension alternative de 240 V et 1 A ou 2 A. Dans le cas d'un petit API, toutes les sorties seront d'un même type, par exemple 240 V alternatif et 1 A. En revanche, avec des API modulaires, il est possible de proposer un éventail de sorties en panachant les modules connectés.

1.3.2 Fourniture et absorption de courant

Les termes *fourniture* (*sourcing*) et *absorption* (*sinking*) décrivent la manière dont les appareils à courant continu sont connectés à un API. Dans le cas de la fourniture, en supposant le sens conventionnel du courant du plus vers le moins, un dispositif d'entrée reçoit le courant à partir du module d'entrée. Autrement dit, le module d'entrée est une source de courant et le fournit au dispositif d'entrée (voir Figure 1.13a). Dans le cas de l'absorption, un dispositif d'entrée fournit le courant au module d'entrée. Autrement dit, le module d'entrée reçoit et absorbe le courant (voir Figure 1.13b). Si le courant va du module de sortie vers une charge de sortie, le module de sortie fournit le courant (voir Figure 1.14a). En revanche, si le courant va de la charge de sortie vers le module de sortie, celui-ci absorbe le courant (voir Figure 1.14b).

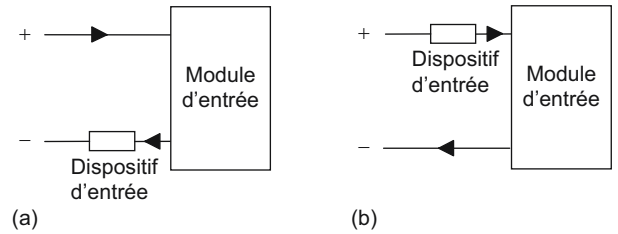


Figure 1.13 – Les entrées : (a) à fourniture et (b) à absorption de courant.

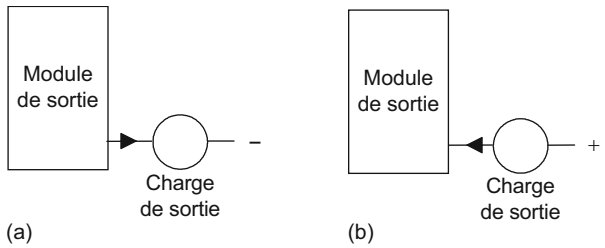


Figure 1.14 – Les sorties : (a) à fourniture et (b) à absorption de courant.

Il est important de connaître le type de la sortie ou de l'entrée afin qu'elle soit branchée correctement à l'API. Ainsi, les capteurs avec sorties à fourniture de courant doivent être connectés aux entrées à absorption de courant de l'API, tandis que les capteurs avec sorties à absorption de courant doivent être connectés aux entrées à fourniture de courant. Si ces directives ne sont pas suivies, l'interface avec l'API ne fonctionnera pas et risque d'être endommagée.

1.4 Systèmes API

Les systèmes API sont principalement disponibles sous deux formes : en *boîtier unique* et en *version modulaire/rack*. Le modèle en boîtier unique, ou *coffret*, est souvent utilisé pour les petits automates programmables et correspond à un système compact complet équipé des unités d'alimentation, de processeur, de mémoire et d'entrées-sorties. Ces API peuvent disposer de six, huit, douze ou vingt-quatre entrées, de quatre, huit ou seize sorties et d'une mémoire permettant d'enregistrer entre 300 et 1 000 instructions. Par exemple, l'API compact TAR 116-6S de Toshiba dispose de huit entrées 120 V à courant alternatif, six sorties à relais et deux sorties à triax, alors que le modèle TDR140-6S plus important possède 24 entrées 24 V à courant continu, 14 sorties à relais et deux sorties à triac. Certains systèmes compacts peuvent être étendus afin d'offrir un plus grand nombre d'entrées et de sorties, simplement en connectant des modules d'extension.

La Figure 1.15 présente l'API compact CP1L d'Omron pour la commande de machines. Pour ce modèle précis, quatre tailles de processeur sont disponibles, chacune avec la possibilité de choisir entre des sorties à relais ou à transistor. La tension, la sortie et le nombre de points d'E/S peuvent être sélectionnés afin de répondre aux besoins. Le boîtier de base, selon le modèle choisi, dispose de 10, 14, 20 ou 30 entrées-sorties (E/S). La version à 10 E/S offre six entrées numériques et quatre sorties, la version à 14 E/S dispose de huit entrées numériques et six sorties, la version à 20 E/S propose douze entrées numériques et huit sorties, tandis que la version à 30 E/S offre dix-huit entrées numériques et douze sorties. Il est possible d'opter pour des sorties à relais, à transistor absorbeur de courant ou à transistor fournisseur de courant. Les modèles à 14, 20 et 30 E/S peuvent être étendus afin d'offrir un plus grand nombre d'entrées-sorties. Par exemple, l'extension du modèle 14 E/S permet d'obtenir 54 entrées-sorties.

La Figure 1.16 présente l'API compact FX3U de Mitsubishi. Les tableaux 1.1 et 1.2 détaillent les modèles de cette gamme de Mitsubishi.

Les systèmes qui offrent un plus grand nombre d'entrées et de sorties ont toutes les chances d'être modulaires et conçus pour s'intégrer dans des racks (voir Figure 1.17). Un système modulaire est constitué de modules séparés pour l'alimentation, le processeur, etc., souvent montés sur des rails dans une armoire métallique. Ce type de système peut être employé pour toutes les tailles d'automates programmables et les différentes unités fonctionnelles sont fournies sous forme de modules individuels qui se branchent à des prises sur un rack de base. Le choix des modules nécessaires à la réalisation d'un projet précis est décidé par