

Automatique et informatique industrielle

Bases théoriques, méthodologiques et techniques

J. Perrin

F. Binet

J.-J. Dumery

C. Merlaud

J.-P. Trichard

AVANT-PROPOS

Ouvrage de base pour ceux qui débutent l'étude des systèmes automatiques, ce manuel est aussi une référence utile pour tous ceux qui poursuivent des études dans le domaine des sciences et techniques industrielles.

Son contenu est adapté :

- Aux élèves de lycée des séries technologiques « sciences et techniques industrielles » ou de la série scientifique à dominante « sciences de l'ingénieur »;
- Aux étudiants des sections de techniciens supérieurs et instituts universitaires de technologie du secteur industriel débutant l'étude des systèmes automatiques;
- Aux étudiants des classes préparatoires scientifiques et technologiques aux grandes écoles qui comportent un enseignement de « sciences industrielles pour l'ingénieur »;
- Aux étudiants de certaines licences et maîtrises technologiques et des grandes écoles;
- Aux auditeurs de la formation continue dans le domaine des sciences industrielles.

Il regroupe sous forme de cours synthétique l'essentiel de ce qu'il y a à connaître en matière de bases de l'automatique en l'état actuel de l'art. Les auteurs ont délibérément restreint le champ d'investigation aux systèmes automatiques combinatoires et séquentiels, câblés et programmés. Les aspects relatifs aux systèmes asservis ou régulés ne sont que cités dans ces pages, leur étude nécessitant à elle seule un ouvrage spécialisé.

Le manuel comporte également des fiches synthétiques destinées soit à présenter des caractéristiques techniques essentielles (« fiches techniques »), soit des outils de compréhension, de choix ou de résolution des problèmes de base de l'automaticien (« fiches méthodes »). L'ouvrage comprend aussi deux « fiches systèmes » décrivant des systèmes automatiques servant d'exemples tout au long de l'ouvrage et de fil conducteur dans la construction du savoir.

Cet ouvrage est pour le professeur à la fois un support de cours qui lui laisse une grande liberté pédagogique et un outil de référence auquel il peut renvoyer ses auditeurs.

Pour l'élève ou l'étudiant, il est un aide-mémoire et un manuel de référence tout au long de sa formation technologique secondaire ou supérieure.

Les auteurs

Nous tenons à dédicacer cet ouvrage à Christian Merlaud.

Nous tenons à remercier, comme pour la précédente édition, les différents constructeurs pour leurs documents.

Édition : Clarisse Darras
Coordination artistique : Isabelle Jalfre
Conception intérieure : Fiat Lux
Composition : JPMsa
Fabrication : Françoise Leroy

© NATHAN 2004 - 9, rue Méchain - 75014 Paris
ISBN 2-09-179452-X



"Le photocopillage, c'est l'usage abusif et collectif de la photocopie sans autorisation des auteurs et des éditeurs. Largement répandu dans les établissements d'enseignement, le photocopillage menace l'avenir du livre, car il met en danger son équilibre économique. Il prive les auteurs d'une juste rémunération. En dehors de l'usage privé du copiste, toute reproduction totale ou partielle de cet ouvrage est interdite."

Sommaire

1.	Introduction à l'étude des systèmes automatiques	4
2.	L'information et ce qui la porte	19
3.	La chaîne d'acquisition	35
4.	Description des systèmes automatiques	49
5.	Comportement des systèmes logiques combinatoires	66
6.	Comportement des systèmes logiques séquentiels	80
7.	Le GRAFCET	93
8.	Réalisation d'une commande	118
9.	Communication	126
10.	Chaîne d'action pneumatique	134
11.	Commande de la puissance électrique	146
12.	Sûreté de fonctionnement	162
FICHE SYSTÈME 1	Palettiseur	174
FICHE SYSTÈME 2	Machine de conditionnement automatique	199
FICHE TECHNIQUE 1	Codeurs optiques industriels	221
FICHE TECHNIQUE 2	Détecteurs industriels	228
FICHE TECHNIQUE 3	Structure et fonctionnement des automates programmables industriels	236
FICHE TECHNIQUE 4	Ateliers logiciels pour les automates programmables industriels	244
FICHE TECHNIQUE 5	Actionneurs électriques	250
FICHE TECHNIQUE 6	Actionneurs pneumatiques	256
FICHE TECHNIQUE 7	Interfaces Homme-Machine	264
FICHE TECHNIQUE 8	Normes, définitions, caractéristiques	270
FICHE MÉTHODE 1	Déterminer les équations d'un circuit combinatoire	273
FICHE MÉTHODE 2	Changer de base numérique	279
FICHE MÉTHODE 3	Décoder un grafcet	285
FICHE MÉTHODE 4	Représenter une évolution temporelle	289
FICHE MÉTHODE 5	Élaborer un schéma de commande ou de puissance	290
FICHE MÉTHODE 6	Choisir un détecteur de position	301
FICHE MÉTHODE 7	Choisir un actionneur électrique et sa protection	305
FICHE MÉTHODE 8	Déterminer les éléments d'une chaîne d'action pneumatique	312
FICHE MÉTHODE 9	Lire un GEMMA	320
Solutions des exercices		324
Lexique		330
Index		334

1

Introduction à l'étude des systèmes automatiques

I. Du besoin au système automatique

1. De l'homo habilis au technicien

Pour assouvir les besoins vitaux des premiers âges, l'homme a utilisé les produits de la nature. Très tôt, il a inventé des outils et des techniques, qu'il n'a eu de cesse de perfectionner. Pour sauvegarder ses forces dans le travail et les déplacements, il a utilisé l'énergie animale... et l'esclavage, avant de domestiquer les multiples sources naturelles. Le temps et l'énergie ainsi économisés ont été réservés pour d'autres activités.

Dans le domaine des technologies, le XIX^e siècle a été marqué par le développement de la mécanisation. Le XX^e siècle, à travers l'émergence du concept d'information, a vu apparaître l'automatisation et l'informatisation qui permettent de réaliser des actions de plus en plus complexes, le plus rapidement et le plus facilement possible.

Cependant, pour obtenir ces performances, que de travail sur de multiples générations humaines, que d'ingéniosité et de créativité, que d'améliorations permanentes des méthodes et des techniques, que de savoirs à mémoriser et à transmettre l'humanité a dû faire preuve pour satisfaire des besoins sans cesse renouvelés, sans cesse plus importants et sophistiqués.

2. Un exemple : le store

Cet exemple du domaine domestique répond au besoin de régler le niveau d'ensoleillement d'une pièce ou d'une aire (terrasse) et illustre, autour de produits comme le parasol et le store, des évolutions ou variantes, fruits de demandes en perpétuelles mutations.

a) Système non mécanisé : parasol

Présent depuis longtemps, ce produit est peu coûteux. Très commode en usage individuel, il devient lourd, difficile à manœuvrer, peu réglable dès que les dimensions augmentent. La présence d'un pied central peut se révéler gênante. Chaque mise en place et chaque rangement demandent de l'effort et du temps. De plus le caractère provisoire de l'installation ne permet pas de garantir la sûreté de fonctionnement vis-à-vis, par exemple, des sautes de vent (Figure 1.1).

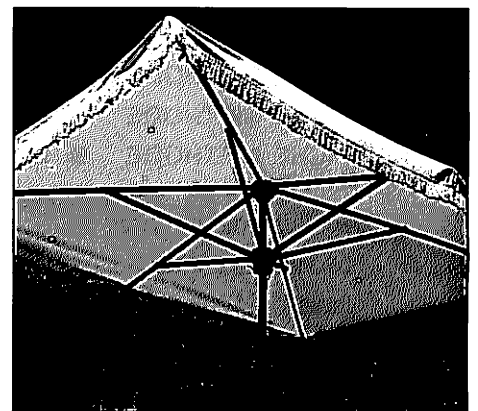


Figure 1.1. Parasol (Marbella, document Triconfort).

Une solution escamotable, mais installée à demeure, existe avec le store : rideau ou panneau souple qui le plus souvent s'enroule.

b) Store mécanisé à manœuvre manuelle : la chaîne cinématique et l'effecteur

Le produit est conçu autour d'un ensemble mécanique assurant, à partir de la manœuvre manuelle d'une manivelle, le déroulement ou l'enroulement d'un tambour ainsi que l'entraînement du store dans ses guides, cela au moyen de transformations de mouvements adéquats (Figure 1.3).

L'action opérative «enrouler ou dérouler» est liée à la décision d'une personne, fruit d'une appréciation de la situation (ensoleillement, vent, état du store).

La transmission mécanique (manivelle, tige, anneau de liaison) permet à l'opérateur sur la terrasse de dérouler facilement le store grâce à un **amplificateur d'effort** (le réducteur). L'élément terminal de la chaîne cinématique, qui convertit l'énergie mécanique de l'opérateur en déplacement du store, est appelé **effecteur** (Figure 1.2).

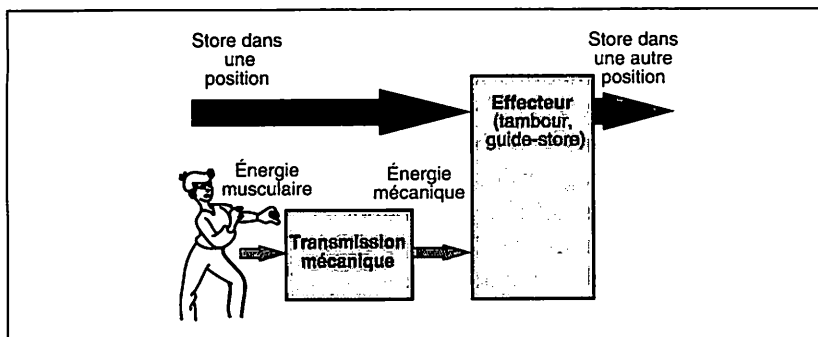


Figure 1.2. Représentation schématique du store à manœuvre manuelle.

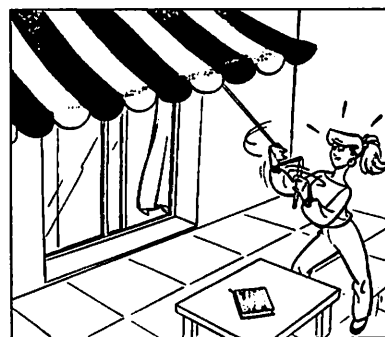


Figure 1.3. Store à manœuvre manuelle (Somfy).

c) Store motorisé

Pour des questions de confort ou de difficulté de manœuvre (personnes âgées, etc.), le besoin initial peut être complété par la contrainte de manœuvrer le store avec un apport énergétique minimal : par exemple, se contenter de maintenir appuyé un bouton-poussoir électrique (Figure 1.5).

Ce bouton-poussoir commande la distribution de l'énergie électrique, prélevée sur une source, à un actionneur (un moteur électrique) qui convertit celle-là en énergie mécanique nécessaire pour faire tourner le tambour et provoquer les mouvements du store (Figure 1.4).

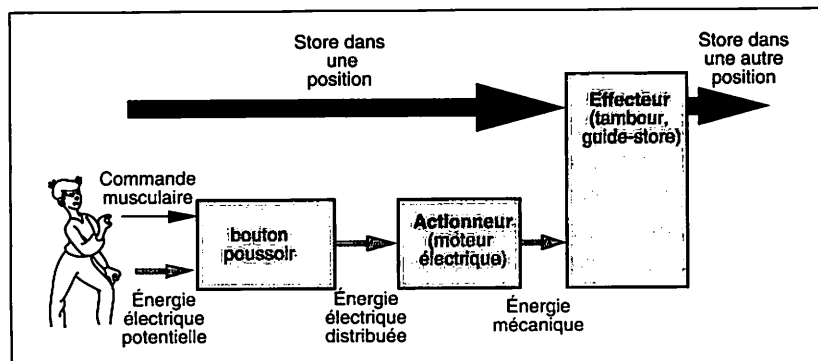


Figure 1.4. Représentation schématisée du store mécanisé.



Figure 1.5. Store mécanisé (Somfy).

L'effecteur n'est pas modifié. La transmission mécanique s'est enrichie d'une motorisation électrique. L'énergie motrice est véhiculée à l'aide de fils conducteurs à partir de coffrets d'alimentation (Figure 1.6).

Il est à noter :

- que la commande peut s'effectuer à distance du store, par les boutons-poussoirs ou des moyens de commande à distance (commande infrarouge) ;
- que l'opérateur doit maintenir le bouton appuyé tant que la manœuvre n'est pas terminée, ce qui peut prendre du temps ;
- qu'en absence d'énergie, le fonctionnement n'est plus possible, à moins de conserver aussi le fonctionnement par manivelle.

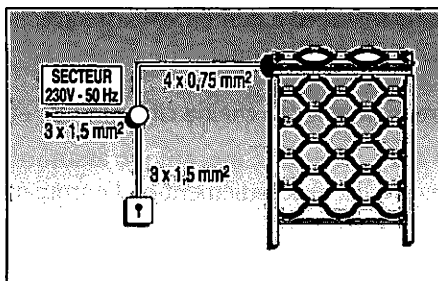


Figure 1.6. Raccordement du store mécanisé (Somfy).

d) Le store automatisé, la partie commande et son modèle de fonctionnement

Les fonctionnements précédents nécessitent la présence humaine en permanence pour provoquer la manœuvre du store. Cette contrainte peut être levée par l'utilisation d'un dispositif de commande capable, à partir d'une consigne de début d'action délivrée à un instant donné par l'opérateur et d'informations sur la position du store, de générer les ordres de commande du store (« dérouler ou enrouler ») en respectant un modèle de fonctionnement. Les informations sont obtenues à partir d'objets techniques capables d'élaborer des signaux, images informationnelles des grandeurs physiques surveillées et exploitables par le dispositif de commande : les capteurs ou détecteurs de position. L'intervention humaine, après que le store est installé, se limite alors à l'émission des consignes de début d'action (Figure 1.7).

Système automatisé
 À l'origine le store mécanisé a été progressivement automatisé pour les raisons évoquées précédemment, les équipements nécessaires à l'automatisation ont donc été ajoutés à un mécanisme existant. Le système mécanique est automatisé (partiellement ou totalement).

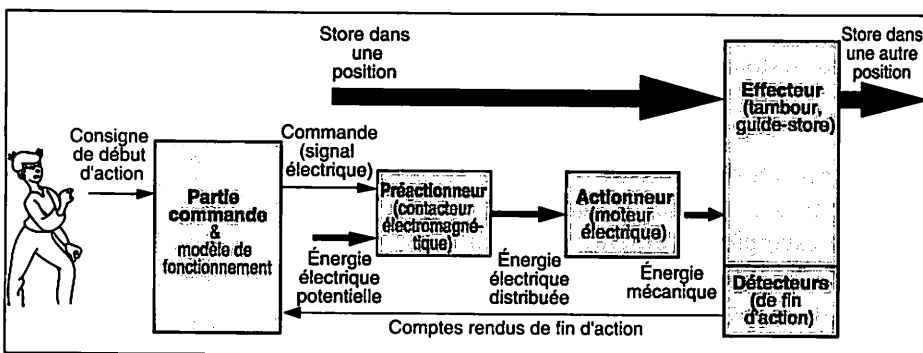


Figure 1.7. Représentation schématique d'un store semi-automatique.

Si cette solution est intéressante lorsque la durée de manœuvre du store est importante (une ou plusieurs minutes), libérant ainsi plus vite l'utilisateur pour d'autres activités, il faut cependant remarquer qu'elle présente un inconvénient important : celui de ne plus pouvoir régler la position du store avec précision. Avec cette solution le store ne peut plus qu'être ouvert ou fermé. Pour obtenir la possibilité de commander à volonté la position du store, il faudra utiliser des moyens supplémentaires, comme par exemple : un temporisateur, un pupitre à plusieurs boutons (positions : fermé, mi-ouvert, trois-quarts ouvert, ouvert) avec autant de détecteurs sur le store, un sélecteur-curseur et un capteur de position, etc. Ce sera plus coûteux et moins facile à utiliser qu'un simple bouton. Le temps gagné d'un côté risque d'être perdu de l'autre.

e) Le store automatique, les capteurs d'environnement et le modèle de décision

Le fonctionnement précédent nécessite encore une présence humaine transitoire pour lancer la manœuvre du store. Cette contrainte peut être levée par l'utilisation d'un objet décidant, à la place de l'homme, des actions à exécuter à un

instant donné : la **partie commande**. Celle-ci permet d'élaborer les ordres de pilotage du store en exploitant des informations sur l'environnement extérieur (état de l'ensoleillement, vitesse du vent, etc.) selon un **modèle de comportement issu du savoir-faire humain**. Ces informations sont obtenues à partir de capteurs : une cellule solaire pour mesurer l'ensoleillement, un anémomètre pour mesurer la vitesse du vent.

Si cette solution présente l'avantage de ne plus nécessiter de présence humaine, libérant l'utilisateur pour d'autres activités, elle présente également des inconvénients :

- nécessitant des éléments complémentaires (capteurs, modèle de décision), elle est évidemment plus coûteuse ;

- plus complexe, elle est aussi moins fiable, donc plus susceptible de tomber en panne ;
- le modèle de décision est difficile à élaborer : que faire par exemple s'il s'agit d'un ciel d'été au soleil ardent mais avec des nuages orageux fréquents (ce qui conduirait à enrouler ou dérouler le store fréquemment) ? Le modèle retenu sera-t-il valable pour toutes les régions et tous les acheteurs ? Et si quelqu'un veut dérouler le store même en l'absence de soleil ? Un modèle universel étant peu réaliste, on sera donc conduit à conserver au moins la possibilité de forcer le déroulement ou l'enroulement par des consignes manuelles... d'où un coût encore plus élevé.

Système automatique

Actuellement ces stores sont produits en grande série et le besoin d'automatisation partielle ou totale est pris en compte dès la conception. Les différents constituants utilisés dans le système sont fortement intégrés. Le système est automatique (tout ou partie des fonctions assurées par le système sont automatiques).

f) Un exemple de réalisation industrielle de store automatique

L'ensemble Somfy Line est constitué (Figure 1.8) :

- d'une motorisation électrique (moto-réducteur intégré dans un tambour enrouleur de toile) ;
- d'un boîtier de commande électronique à micro-contrôleur programmé, avec sélecteur de marche (automatique et manuelle) ;
- d'une détection des niveaux d'ensoleillement et de vent.

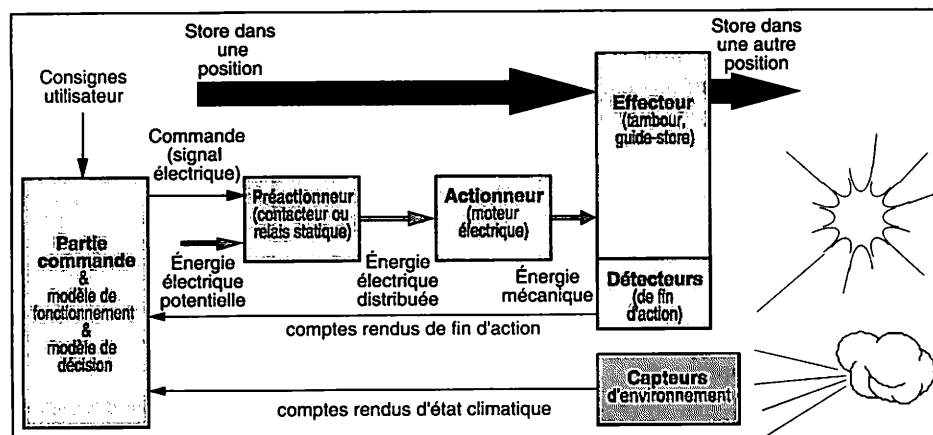


Figure 1.8. Représentation schématique d'un store automatique.

Lorsque la luminosité du soleil, captée par la cellule solaire, dépasse un seuil prédéterminé et réglable, la commande fait descendre le store automatiquement.

Lorsque la vitesse du vent dépasse un seuil prédéterminé et réglable, la commande permet de protéger l'installation contre les risques de détérioration de la toile en faisant monter le store. Ce fonctionnement de sécurité est prioritaire sur tous les autres.

Les positions « toile complètement enroulée » et « toile complètement déroulée » sont indiquées par deux micro-contacts réglables indépendamment. La commande manuelle de montée ou de descente de la toile est possible en utilisant les boutons-poussoirs « montée ou descente » (Figure 1.9). Un appui simultané sur les deux provoque l'arrêt de la toile dans la position où elle se trouve.

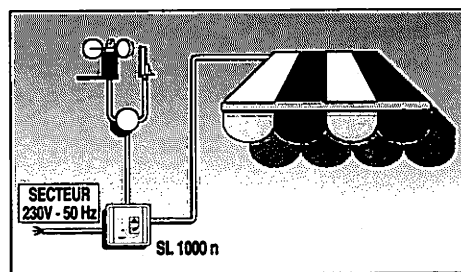


Figure 1.9. Raccordement d'un store automatique (Somfy).

II. Les systèmes automatiques

1. Fonction globale d'un système automatique

a) Élaborer un produit à partir de matières d'œuvre...

Un système automatique a pour but d'élaborer des biens ou des services destinés à la consommation ou à l'équipement de clients.

b) Apporter de la valeur ajoutée à des produits...

Un produit est ce qui est, ou sera, fourni à un utilisateur pour répondre à son besoin (définition de la norme NF X 50-150). La valeur ajoutée apportée aux produits entrant dans le système automatique est la réponse à ce besoin.

Les systèmes automatiques apportent, sans intervention humaine en fonctionnement normal, de la valeur ajoutée sur des produits qui peuvent être de la matière, de l'énergie et (ou) de l'information.

Cette valeur ajoutée est apportée par des procédés de transformation, de transport ou encore de stockage (quelques exemples sont donnés dans le tableau ci-dessous – Figure 1.10).

Type de produit	Fonction globale		
	Transformer	Transporter	Stocker
MATIÈRE – Matières premières (minerai de fer, méthane, bois, eau, sable, fourrage, céréales, ...); – Biens intermédiaires ou matières d'œuvre, plus ou moins transformés (barres ou plaques d'acier, plâtre, granulés de polyéthylène, papiers et cartons d'emballage, plaquettes de silicium, lait, farine, ...); – Articles manufacturés (pièces mécaniques, plaques de plâtre, sacs plastique, bouteilles d'eaux, ordinateur, plaquettes de beurre, yaourts, paquets de gâteaux, ...).	Exemples : Production sidérurgique, usinage par machine outil, découpe, pliage, cambrage, forgeage, pasteurisation, lyophilisation, dégazage, ...	Exemples : Transport routier, transport par tapis roulant, par chaîne, par bol vibrant, ...	Exemples : Ensilage, embouteillage, palettisation, ...
ÉNERGIE Energie sous de multiples formes : – distribuée par réseau (énergie électrique, essence, gaz, ...); – par lot (piles, bouteilles de gaz, ...).	Exemples : Production hydroélectrique, génération dynamométrique, combustion, explosion, ...	Exemples : Transport par réseau électrique, oléoduc, gazoduc, transport routier, ...	Exemples : Stockage dans des batteries, des barrages, des réservoirs, de bouteilles de gaz liquéfié, ...
INFORMATION – Images ; – Sons ; – Textes ; – Grandeur physique ; – ...	Exemples : Transformer des informations en vue de les imprimer, de modifier et de projeter des images, de diffuser un son de synthèse. Réaliser des calculs logiques et numériques.	Exemples : Transport à l'aide de lignes téléphoniques, réseaux de télévision, réseaux informatiques, ...	Exemples : Stockage sur supports numériques (disques magnétiques, cédérom) ou sur papier (revue, livres, ...)

Figure 1.10. Classification des fonctions globales des systèmes techniques.

EXEMPLE

Voir les FICHES SYSTÈMES 1 et 2.

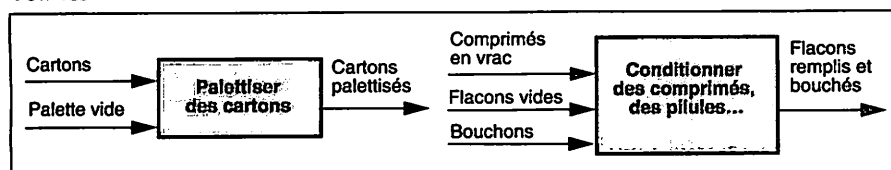


Figure 1.11. Exemples de fonctions globales (Palettiseur et machine de conditionnement).

c) Au moyen d'un système automatique...

L'élaboration du produit s'effectue (Figure 1.12) :

- au moyen d'un système automatique (constitué d'une partie opérative et d'une partie commande : voir II-2) ;
- en relation avec un environnement humain, physique et informationnel.

Une partie de cet environnement humain, physique et informationnel constitue des ressources pour le système :

- ensemble des intervenants sur le système : utilisateurs, agents d'exploitation et de maintenance ;
- réseaux d'énergie et d'utilitaires, eau, vapeur ;
- réseaux d'informations (banques de données techniques ou de gestion, ordres de fabrication, stocks de matières d'œuvre, d'outillages, de pièces de rechange).

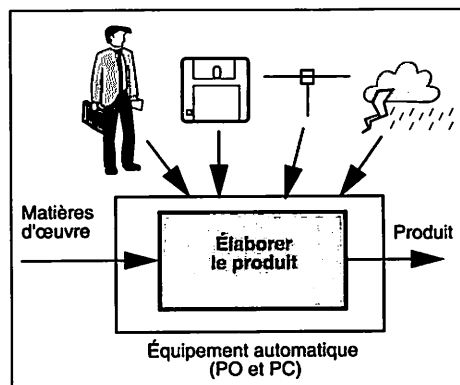


Figure 1.12. Environnement technique et humain d'un système automatique.

d) En fonction d'objectifs, de contraintes et de risques

À tout système automatique sont associés des objectifs, des contraintes et des risques.

- **Les objectifs** définissent les attentes de l'utilisateur vis-à-vis du système automatique. Ils sont en relation avec le produit, la production, l'exploitation, la maintenance, etc.
- **Les contraintes** définissent les conditions imposées pour l'exploitation du système. Elles sont également liées au produit, à la production, au processus et à l'équipement de production, à l'exploitation et à la maintenance.
- **Les risques** sont les risques sur les personnes, sur les biens ou sur l'environnement liés à l'exploitation du système, les risques économiques que l'utilisateur peut subir en cas de dysfonctionnement, de panne prolongée ou de détérioration du système.

EXEMPLE

Machine de conditionnement (voir FICHE SYSTÈME 2).

- **Objectifs :**
 - adaptation à une grande variété de produits (comprimés, gélules, pilules, bonbons, etc.) et de récipients (flacons verre, tubes plastiques, etc.) ;
 - cadence minimale de 10 000 comprimés par heure ;
 - exploitation par un personnel sans qualification particulière ;
 - aucun risque d'accident pour les opérateurs, maintenance rapide.
- **Contraintes :**
 - énergies d'alimentation : triphasé 400 V, air comprimé 7 bars maximum ;
 - commande par automate programmable.
- **Risques :** accident opérateur en maintenance, indisponibilité prolongée.

Ces objectifs, ces contraintes et ces risques, qui influencent le fonctionnement du système, se traduisent en pratique par un ensemble d'informations de contrôle (ordres de fabrication, consignes d'exploitation, conditions, données techniques, etc.).

La fonction globale d'un système automatique, à savoir «élaborer le produit», est ainsi définie par quatre éléments : le produit entrant, le système automatique, les informations de contrôle et, évidemment, le produit final (Figure 1.13).

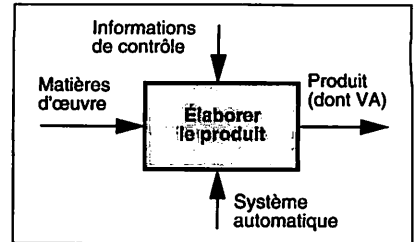


Figure 1.13. Fonction globale d'un système automatique.

EXEMPLE

Fonction globale du palettiseur figure 1.14. (voir FICHE SYSTÈME I).

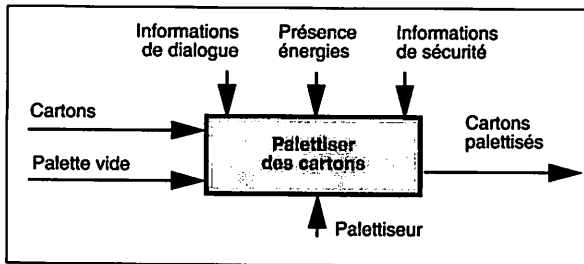


Figure 1.14. Fonction globale du palettiseur.

REMARQUE

Le terme de système est très utilisé, mais recouvre des concepts divers qui nuisent à une bonne compréhension. Dans cet ouvrage on s'attachera à utiliser des mots plus précis.

2. Quelques termes clés

● **Procédé**

Méthode utilisée pour réaliser une opération de production de bien (transformation de matière d'œuvre, conditionnement, manutention, stockage, etc.) ou une opération de production de service (communication de données, traitement de données, etc.)

● **Processus**

Enchaînement et évolution temporelle ordonnés de procédés.

● **Système du domaine technique**

C'est un ensemble technique conçu pour répondre à un besoin. Il assure une fonction. D'une façon générale, il opère sur des matières d'œuvre (produit(s) entrant(s)) pour les faire passer d'un état initial à un état final, créant ainsi une valeur ajoutée.

● **Frontière d'un système**

C'est une limite fictive qui peut être purement fonctionnelle (dans le cas où le système est à créer ou si la frontière ne correspond pas à un découpage matériel du système) et qui permet de caractériser le système étudié et ses entrées/sorties. La mise en place de cette frontière est un préalable indispensable à toute étude.

● **Entrées et sorties d'un système** (Figure 1.15)

Tout système réalise un traitement sur des entrées pour élaborer des sorties. Les entrées et les sorties sont :

- des matières;
- des énergies;
- des informations.

Parmi ces entrées/sorties on distingue :

- les entrées qui sont utiles pour le système (pour l'obtention de la valeur ajoutée) et les sorties qui permettent de répondre au besoin;
- de celles qui constituent des perturbations ou des nuisances.

Entrées de matières d'œuvre : la fonction du système est le traitement de ces matières d'œuvre, pour les faire passer d'un état initial à un état final.

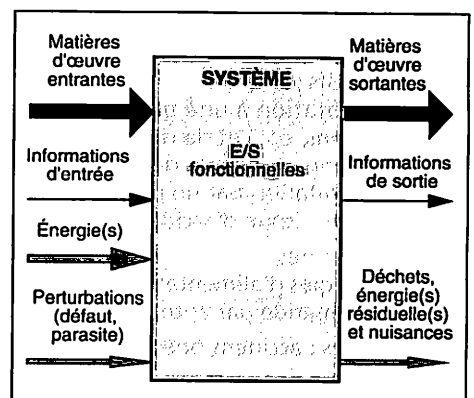


Figure 1.15. Inventaire des entrées et sorties d'un système industriel.

Entrées/sorties d'informations : elles sont essentielles pour le dialogue homme/système ou pour la communication avec l'extérieur (renseignements sur l'environnement ou en provenance d'autres systèmes).

Entrées/sorties d'énergie : l'entrée d'énergie(s) – plusieurs sources sont utilisées sur les équipements industriels – est indispensable à la mise en œuvre des processus physiques (la sortie d'énergie correspond soit à la sortie principale, soit à des énergies résiduelles ou perdues lors des conversions énergétiques ou par frottement et dans ce cas leur représentation n'est pas indispensable).

Entrées/sorties intempestives : il s'agit là de perturbations (défaut sur une ou plusieurs entrées du système, parasite énergétique) et de sorties de type déchets ou nuisances (projections, bruit, chaleur, fuites, etc.). La conception du système cherche à minimiser leur influence au moindre coût. La nature, souvent aléatoire, de ces perturbations est une source importante de difficultés pour l'établissement de modèles de fonctionnement et de décision fiables. Sauf cas particulier d'étude ou de traitement spécifique de ces nuisances, elles ne seront pas représentées sur les modèles de descriptions utilisés dans l'ouvrage.

● **Structure d'un système automatique**

Il est constitué d'une **partie opérative (PO)** et d'une **partie commande (PC)**.

Les informations échangées entre la partie commande et la partie opérative sont des informations internes au système, alors que les consignes, signalisations et autres messages proviennent ou sont destinés à l'extérieur.

Sauf pour les systèmes asservis et pour la lisibilité des schémas, on ne représente pas les entrées et sorties intempestives qui ne font pas partie des objectifs de conception (Figure 1.16).

REMARQUE

Pour des raisons de clarté, seules sont représentées les entrées et sorties de nature informationnelle ou matérielle. Les flux d'énergie sont ainsi implicites (sauf pour les systèmes traitant de l'énergie comme matière d'œuvre).

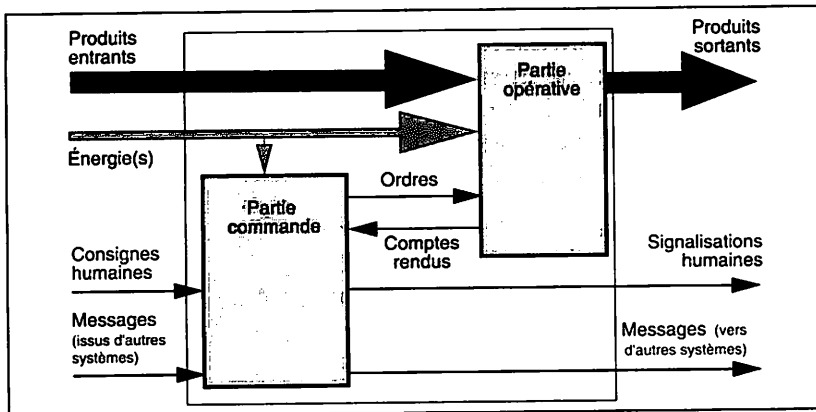


Figure 1.16. Structuration en deux parties d'un système automatisé.

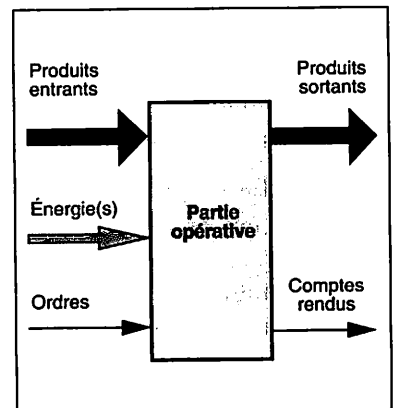


Figure 1.17. Entrées et sorties de la partie opérative.

● **La partie opérative**

C'est elle qui assure les transformations des produits entrants permettant d'élaborer la valeur ajoutée recherchée (Figure 1.17).

Elle est le siège des conversions énergétiques. C'est, en règle générale, un ensemble mécanisé bien qu'il existe des systèmes dont la partie opérative n'est pas animée (système de traitement de l'information, actionneurs statiques, dispositifs de modulation ou de conversion de l'énergie, etc.). Les principales technologies des chaînes d'action, mises en œuvre, utilisent les énergies générées par de l'électricité, de l'air comprimé (pneumatique) et des fluides hydrauliques.

La définition, la réalisation et la mise au point de la partie opérative est une difficulté majeure dans l'élaboration d'un équipement et une condition essentielle à la réussite du projet.

Au plan informationnel, elle émet vers la partie commande des comptes rendus (informations caractéristiques des modifications opérées sur les produits et/ou sur les mécanismes). Elle reçoit des ordres (informations à caractère énergétique déclenchant, arrêtant ou modulant les transformations énergétiques nécessaires aux actions sur les matières d'œuvre).

● La partie commande

Elle commande la partie opérative pour obtenir les effets voulus, par l'émission d'ordres (sorties de la partie commande) en fonction d'informations disponibles, comptes rendus, consignes, etc. (entrées de la partie commande) et du modèle de fonctionnement. Elle peut échanger des informations avec des humains ou d'autres systèmes (Figure 1.18).

Les parties commandes sont construites à partir de constituants électroniques et électriques et s'appuient essentiellement sur des technologies programmées (automates programmables, micro-contrôleurs, etc.). Les traitements sont réalisés au plus près des actions à réaliser sur les produits. Les parties commandes sont ainsi de plus en plus réparties et distribuées et sont alors reliées à l'aide de réseaux.

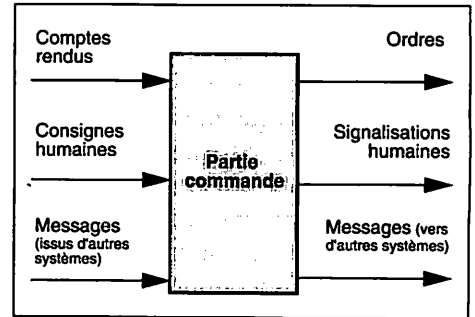
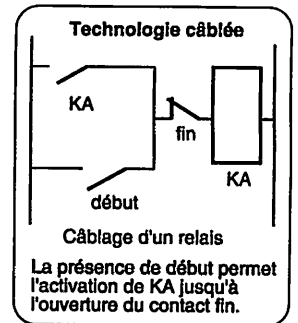


Figure 1.18. Entrées et sorties de la partie commande.

● Technologies câblées et programmées

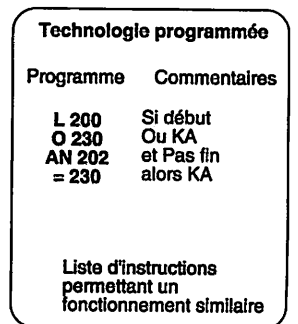
Avec une technologie câblée, la réalisation de la partie commande est obtenue par l'utilisation de composants standard et de liaisons adéquates (câblage), caractéristiques du fonctionnement attendu (Figure 1.19).

À l'inverse, avec une technologie programmée, le fonctionnement est obtenu à partir d'un programme (ensemble d'instructions) implanté dans un constituant programmable adapté à la famille d'applications envisagée. Les modifications sont aisées et les capacités de traitement sans cesse en croissance. Les développements de la micro-électronique ont permis à cette technique de s'imposer dans les domaines des plus simples aux plus complexes.



● Architecture topo-fonctionnelle

L'organisation détaillée d'un système automatisé (voir par exemple celle du store automatique, Figure 1.8) montre qu'elle résulte de l'assemblage d'un ensemble de constituants en relation topographique les uns avec les autres et assurant chacun une fonction particulière.



Il s'agit d'une architecture commune à tous les systèmes automatisés. Elle est dite « topo-fonctionnelle » car elle décrit aussi bien l'arrangement topographique des constituants du système, que l'organisation des relations entre les fonctions techniques du système ou fonctions d'automatisation (Figure 1.20).

Figure 1.19. Représentation d'une technologie câblée et d'une technologie programmée.

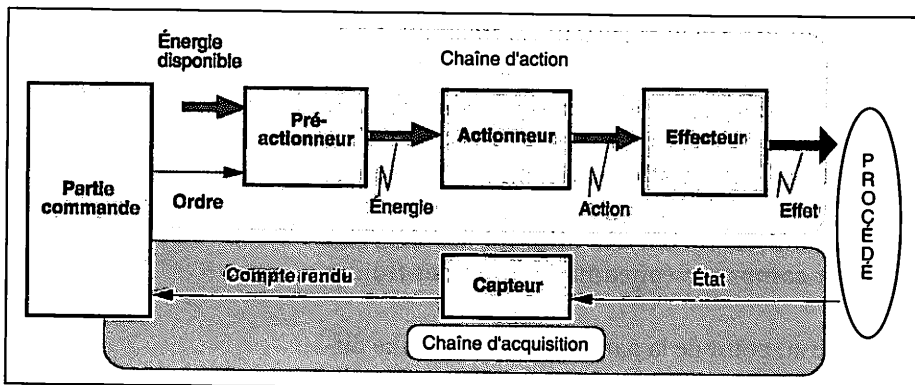


Figure 1.20. Architecture topo-fonctionnelle des constituants d'un système automatisé.

3. Chaîne fonctionnelle

La chaîne fonctionnelle constitue l'unité élémentaire de conception et d'étude du fonctionnement d'un système automatisé.

Elle se caractérise par un agencement fonctionnel de constituants sous forme de chaîne, en regroupant tous les éléments de la partie opérative et de la partie commande du système concourant à la réalisation d'une fonction opérative. C'est parce que tous ces éléments sont liés fonctionnellement que le regroupement s'appelle chaîne fonctionnelle. Elle comporte en général trois parties : la chaîne d'action, la (ou les) chaîne(s) d'acquisition, la partie traitement.

● La chaîne d'action

Elle assure, à partir de l'élaboration des ordres au sein du constituant de traitement, les animations de la partie opérative nécessaires aux actions sur la matière d'œuvre (Figure 1.21).

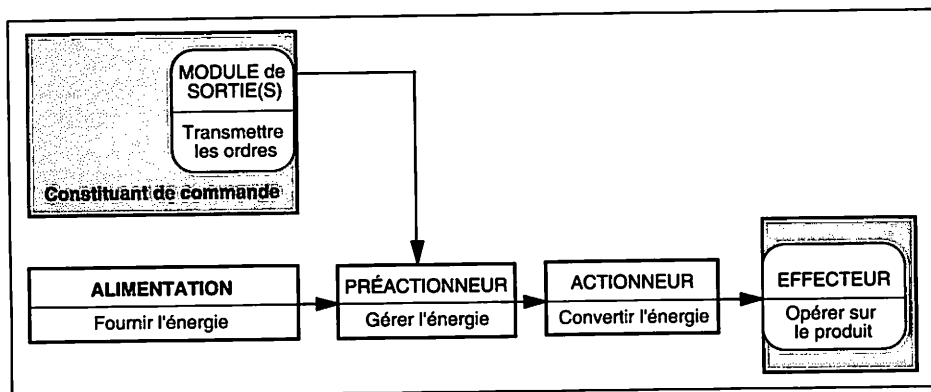


Figure 1.21. Chaîne d'action.

● La ou les chaîne(s) d'acquisition

Elle regroupe l'ensemble des constituants permettant d'élaborer et de transmettre les informations exploitables par l'unité centrale et réalise la fonction « acquérir les informations » (Figure 1.22). Elle peut comporter :

- une chaîne d'acquisition d'information(s) sur le produit ou sur le processus (souvent appelée chaîne de retour). La mission de cette chaîne consiste à prélever l'information au plus proche des produits subissant des transformations (information source) et à élaborer un signal (image informationnelle) compatible avec les énergies utilisées par l'unité de traitement (pneumatique, électrique, électronique, etc.). Ces chaînes sont constituées autour des détecteurs (image informationnelle logique) et des capteurs (images informationnelles numériques et analogiques) ;

- une chaîne d'acquisition d'informations extérieures (consignes de l'opérateur, messages transmis par une autre partie commande) ;

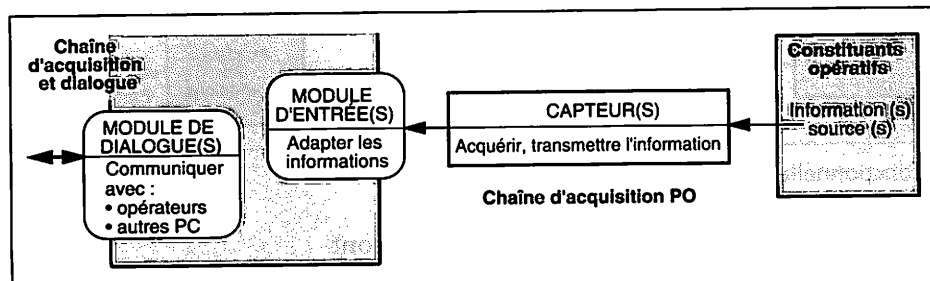


Figure 1.22. Chaîne d'acquisition.

REMARQUE

La prise d'informations n'est pas toujours réalisée directement au voisinage du produit (impératifs techniques et économiques), mais indirectement au niveau des divers éléments de la chaîne. Par exemple, l'attestation du déploiement du store est faite à partir de la détection de positions limites du tambour, ce qui ne garantit pas que la toile soit déroulée (ou enroulée).

● La partie traitement de la chaîne fonctionnelle

Elle traite les informations des chaînes d'acquisition pour élaborer les ordres à destination de la chaîne d'action.

EXEMPLE

Chaîne fonctionnelle « saisir les cartons » du palettiseur (Figure 1.23).

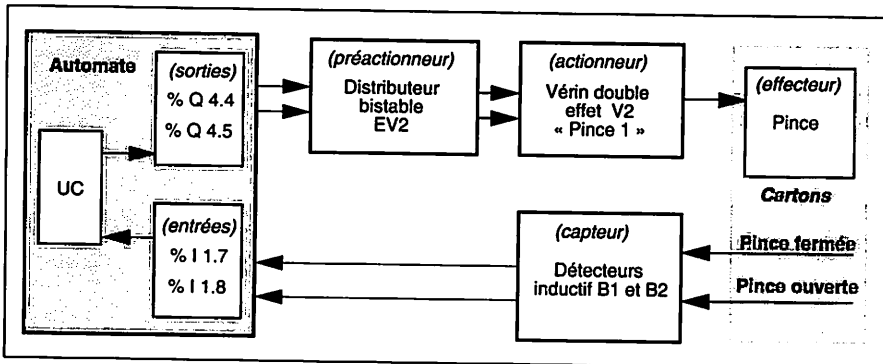


Figure 1.23. Chaîne fonctionnelle CF31.

Chaîne fonctionnelle CF421 « aspiration par ventouse » de la machine de conditionnement (Figure 1.24).

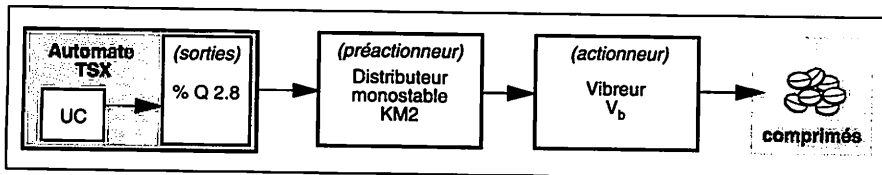


Figure 1.24. Chaîne fonctionnelle CF421.

REMARQUE

Une chaîne fonctionnelle peut :
 – être associée à plusieurs actions (exemples : « sortir tige » et « rentrer tige » pour un vérin double effet, « marche avant » et « marche arrière » pour un moteur deux sens de marche);
 – comporter plusieurs capteurs (deux détecteurs fin de course par exemple) ou encore n'en comporter aucun (elle se réduit alors à la chaîne d'action);
 – comporter deux ou plusieurs actionneurs (exemple : deux vérins de levage symétriques pour une charge très lourde).

4. Représentation des fonctions d'automatisation

● Architecture topo-fonctionnelle générale

Chacun des constituants de la structure organique réalise une fonction d'automatisation qu'il est possible de représenter dans une structure semblable.

Il est possible de regrouper sur un même schéma l'architecture générale d'un système automatisé. Afin de ne pas alourdir la représentation, on convient de ne pas détailler les fonctions de dialogue et de communication (Figure 1.25).

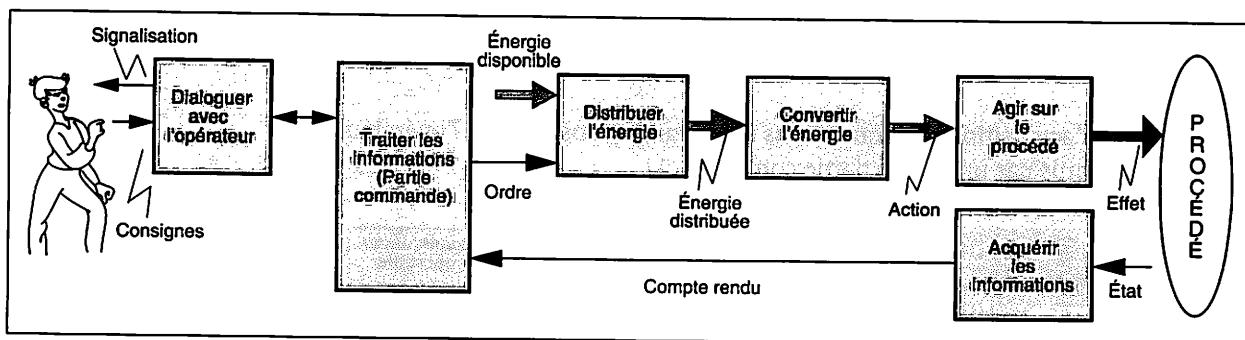


Figure 1.25. Architecture topo-fonctionnelle générale.

Il s'agit d'une forme extrêmement générale. D'autres représentations, qui sont nécessaires pour définir au mieux un équipement, particulièrement dans la relation ordres et comptes rendus, seront vues plus loin.

● **Fonctions et constituants**

Le tableau de la figure 1.26 ci-après regroupe, pour le store automatique Somfy, les principales fonctions d'automatisation, les missions spécifiques réalisées dans le cadre de l'application et les constituants utilisés.

Fonctions d'automatisation	Missions spécifiques	Objets techniques associés
Acquérir l'information	Détecter les positions extrêmes du tambour, l'intensité solaire, la vitesse du vent.	Détecteurs : – de position ; – photo-électrique ; – de vitesse.
Distribuer l'énergie	Alimenter en énergie électrique le moteur monophasé à deux sens de marche.	Préactionneurs (contacteurs électromagnétiques).
Convertir l'énergie	Convertir l'énergie électrique en mouvement de rotation de l'arbre de sortie du moteur et adapter le couple et la vitesse.	Moteur électrique monophasé à deux sens de marche. Réducteur de vitesse.
Agir sur la matière d'œuvre	Assurer la rotation du tambour enrouleur et ainsi agir sur le déplacement de la toile.	Effecteur : Tambour et ensemble de guidage de la toile.
Traiter les informations	Mettre en oeuvre le modèle de fonctionnement.	Boîtier électronique (carte microprogrammée).
Dialoguer avec l'opérateur	Informar l'opérateur. Choisir les modes de marches. Réaliser les réglages (introduire les paramètres).	Objets de dialogue : Boutons poussoirs, sélecteurs de commande, potentiomètres, voyants.

Figure 1.26. Fonctions et constituants du store automatique.

5. Une classification des systèmes automatiques

Il est possible de distinguer, entre autres, deux grandes familles de systèmes automatiques :

- les systèmes de grande diffusion ;
- les systèmes de diffusion limitée.

● **Les premiers**, encore appelés « grand public », couvrent un usage général. Ils affectent notre existence quotidienne au travers de l'électroménager, de la domotique, de la distribution (distributeurs de billets, stations-service...), du secteur médical (pompes, respirateurs, épurateurs...), du transport, etc. Produits en grand nombre, leur conception vise à la recherche de solutions optimales fortement intégrées (voir la notion de système automatique au paragraphe 1 2), compatibles avec la production de série et les contraintes de la concurrence.

● **Les seconds** couvrent un usage spécifique. On rencontre, notamment, ce type d'équipement dans le domaine de la production industrielle : ce sont les **systèmes automatiques de production**. Le nombre limité, souvent unitaire, de ces systèmes interdit des études longues et coûteuses qui seraient difficiles à amortir. Ils sont, en règle générale, réalisés par l'assemblage de constituants et composants standard, pour la partie opérative et la partie commande. Ils doivent être évolutifs (modularité, commandes réparties et distribuées, installations et câblages rapides, intégration de plus en plus de logiciels applicatifs permettant une évolution aisée, ...) pour s'adapter aux évolutions des besoins des utilisateurs.

6. Conception et réalisation d'un système automatique

● Étapes de la vie d'un produit, le cycle de vie

Le système automatique comme nombre de produits obéit à une loi d'existence. Sa «vie» peut se caractériser par quelques phases types (Figure 1.27) :

- découverte et formalisation du besoin;
- conception;
- réalisation;
- exploitation avec fréquentes évolutions;
- extinction (destruction, recyclage).

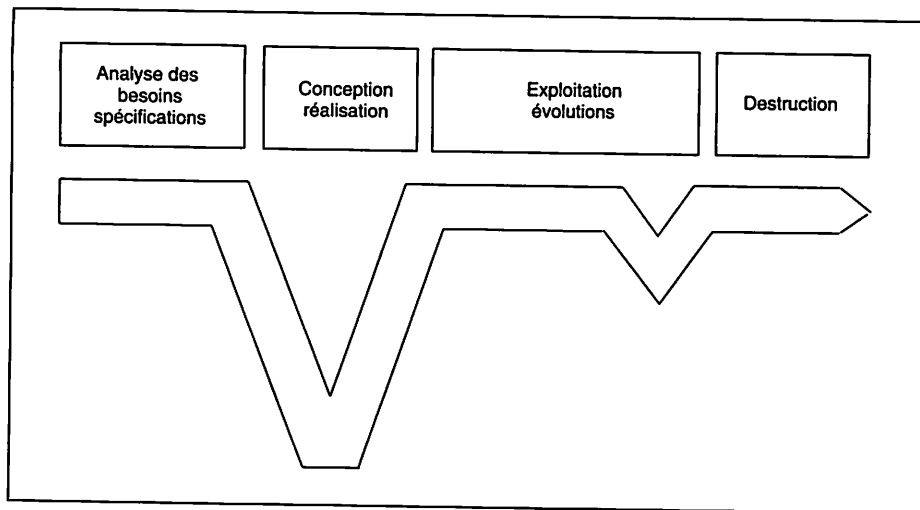


Figure 1.27. Étapes de la vie d'un système automatique.

L'automaticien intervient surtout en phases de conception, de réalisation et de plus en plus en phase d'exploitation tant pour la conduite que pour la maintenance (dépannages, améliorations, etc.).

● Modèles

Ces grandes étapes de la vie du système automatique sont marquées par l'intervention d'équipes aux fonctions, aux métiers, aux cultures et aux intérêts très différents, mais qui doivent concourir à la réalisation d'un objet de qualité (répondant le mieux possible au besoin). La nécessité d'échanges entre les divers acteurs, soit au sein d'une équipe, soit entre équipes, implique – afin de communiquer aisément et clairement – des représentations contractuelles : les schémas et langages de programmation normalisés ainsi que les modèles comportementaux.

Outre cette mission de faciliter le dialogue et les échanges, la représentation a une autre mission, celle d'aider le(s) spécialiste(s) qui l'a établie à mieux définir la solution.

La solution, compromis entre diverses contraintes économiques, techniques et humaines, fait appel à de nombreuses technologies différentes : mécaniques, électriques, pneumatiques, hydrauliques, informatiques, réseaux, etc.

Un des objectifs de cet ouvrage est de permettre la compréhension et la construction de modèles qui accompagnent le système automatique de sa réalisation à son exploitation.

III. Description d'un système automatique

1. Caractérisation d'un système automatique

Un système automatique peut, sur le plan technique, être observé et caractérisé de trois manières différentes (Figure 1.28) :

– par les **fonctions de service** qu'il assure, fonctions qui s'exercent vis-à-vis des produits et de l'environnement humain, physique et informationnel. Ces fonctions sont l'expression du besoin que le système satisfait ;

– par sa **structure**, c'est-à-dire la manière dont les différents constituants matériels et logiciels qui le composent (les effecteurs, les actionneurs, les préactionneurs, les capteurs, les automates programmables, les pupitres de dialogue, les réseaux de communication) sont organisés et reliés topographiquement pour réaliser les fonctions ;

– par les **tâches effectuées** pour assurer les fonctions, tâches qui se manifestent lors du fonctionnement du système par la succession chronologique des actions effectuées en fonction des informations reçues.

Structure et tâches expriment comment les fonctions sont réalisées dans l'espace et dans le temps.

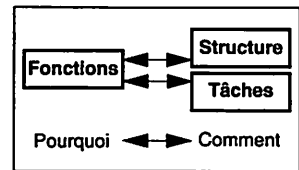


Figure 1.28. Trois points de vue.

2. Modèles de représentation d'un système automatique

Pour représenter un système automatique et son fonctionnement, il faut utiliser des modèles de représentation adaptés à chacun des points de vue précédents.

Il existe une grande variété de modèles et d'outils de représentation, ainsi que le montre la figure 1.29. Chacun présente des avantages et un champ d'utilisation propre. Le choix des modèles adaptés à un usage donné dépasse le cadre de cet ouvrage.

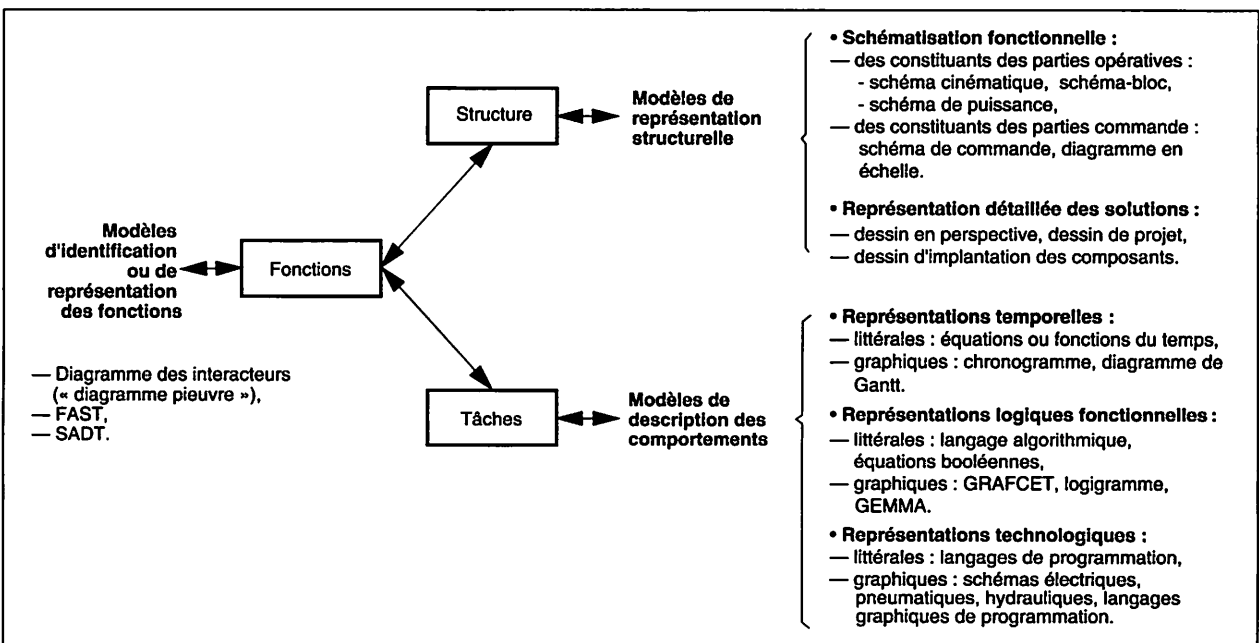


Figure 1.29. Modèles de représentation.

- Pour répondre à ses besoins l'homme a su créer les objets et les systèmes artificiels qui lui étaient nécessaires, des systèmes purement manuels aux systèmes automatiques, en passant par les systèmes simplement mécanisés. Les systèmes automatiques apportent une valeur ajoutée à un flux de produits (matière, énergie, information) au moyen de divers équipements et en fonction d'objectifs et de contraintes exprimés dans un cahier des charges.
 - Les systèmes automatiques associent un ensemble opératif (partie opérative) à une commande (partie commande) centralisée ou répartie. La partie commande génère des ordres en respectant un modèle de fonctionnement, en s'appuyant sur des informations d'état de la partie opérative et de son environnement.
 - Dans un système automatique, on peut distinguer des chaînes d'action (ou d'énergie) et des chaînes d'acquisition des informations. Ces informations sont traitées par la partie commande pour produire les ordres. Cette représentation fonctionnelle peut revêtir aussi un caractère structurel où l'on distingue :
 - pour la chaîne d'action, le module de sortie de la partie commande, un préactionneur, qui distribue, l'énergie disponible vers l'actionneur, qui la convertit pour l'effecteur qui l'utilise pour opérer sur la matière d'œuvre ;
 - pour la chaîne d'acquisition, un élément de partie opérative (ou d'environnement) sur lequel est prélevée l'information source utile, un capteur qui reçoit, convertit et transmet l'information vers le module d'entrée de la partie commande qui l'adapte au traitement.
 - Selon la complexité du système automatique et de son comportement attendu, les informations recueillies sur la partie opérative et son environnement sont plus ou moins nombreuses et complexes. Des systèmes très simples, comme un éclairage automatique qui s'allume en présence d'un individu et s'éteint en son absence, jusqu'à des systèmes robotisés versatiles et capables d'auto-apprentissage, tous ces systèmes sont l'objet de l'automatique. Il existe deux grandes catégories de systèmes automatiques : les systèmes de grande diffusion (destiné à un large public) et les systèmes spécifiques à un usage restreint donné (comme les systèmes de production industrielle).
 - La conception, la mise en œuvre, la maintenance d'un système automatique nécessitent d'en comprendre les fonctions (à quoi il sert), la constitution (sa structure matérielle, donc la technologie : qu'est-ce qu'il est possible de faire avec les moyens d'aujourd'hui), et son comportement.
- Pour cela il est nécessaire, après avoir caractérisé l'objet de l'étude par sa frontière et ses entrées/sorties, d'utiliser des « modèles » et des représentations adaptés au « point de vue » retenu, c'est-à-dire à l'objectif visé par l'analyse (représentation des fonctions, programmation, maintenance, etc.).

Exercices

EXERCICE 1. Identifier, dans l'environnement quotidien, cinq exemples de systèmes manuels, mécanisés, partiellement ou complètement automatisés. Préciser, pour chacun, les produits entrants et sortants ainsi que la valeur ajoutée apportée.

EXERCICE 2. Sur un magnétoscope, pour la fonction « dialoguer avec l'opérateur », préciser les consignes et les paramètres à régler pour effectuer un enregistrement.

2

L'information et ce qui la porte

I. L'information et son traitement

1. L'information

Le mot « information » est associé à des définitions différentes selon son contexte d'utilisation et son domaine d'application (biologie, théorie du signal, thermodynamique et notion d'entropie, ...).

Dans le cadre de cet ouvrage on retiendra la définition suivante :

Une information est un élément de connaissance susceptible d'être codé pour être conservé, traité ou communiqué.

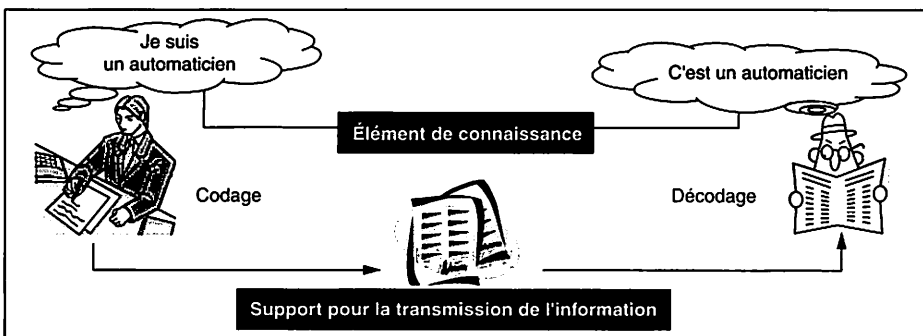


Figure 2.1. L'information élément de connaissance communicable

Un exemple typique est celui de l'écriture humaine. L'élément de connaissance à transmettre (l'information) est codé grâce au langage écrit. Le support de l'information peut être un document papier qui peut alors être transmis par divers moyens au destinataire de l'information appelé le récepteur. L'émetteur de l'information est celui qui écrit le message afin de le transmettre.

Les systèmes automatiques, pour pouvoir évoluer et apporter de la valeur ajoutée, ont besoin d'informations issues de leur environnement humain et matériel, d'information sur l'état du (ou des) produit(s) ou d'énergie sur lequel ils agissent et sur leur propre état. Compte tenu des technologies utilisées, le support d'information est dans la plupart des cas un signal électrique, une onde (lumineuse, radio, ...), plus rarement un signal pneumatique.

L'information qui est destinée à être transmise relève d'une typologie à trois classes (Figure 2.2), selon le besoin ou l'usage qui en est fait :

- à un besoin élémentaire, du type « la porte du palettiseur est fermée », « un carton est arrivé en fin de convoyeur », etc., correspond une proposition du type « vrai ou faux ». Il s'agit d'une information logique ;
- si le besoin correspond à une information plus complète et discriminante, comme une quantité (de couches, de flacons remplis, etc.) ou un rang (ordre de rangement), il faut exprimer un nombre. Cette information est numérique ;

REMARQUE

Dans le domaine de l'automatique, on associe généralement le mot « information » aux données d'entrée de la commande c'est-à-dire au récepteur. Lorsque la commande devient l'émetteur et adresse une information vers l'extérieur (préactionneurs, pupitre), on utilise plutôt le mot « ordre ».

– enfin, à un besoin de suivi permanent répond une information continue (ou analogique, par opposition à logique).

Type d'information	Information logique (deux états)	Information numérique (états dénombrables dans \mathbb{Q})	Information continue (états non dénombrables)
Exemples	Présence/Absence Vrai/Faux Fait/Non fait	Rang dans une liste Cardinal d'un ensemble Note d'un devoir	État d'une grandeur Vitesse Température

Figure 2.2. Les trois classes d'information.

La théorie de l'information, présentée par C. Shannon en 1948, alors ingénieur aux *Bell Telephone Laboratories*, a posé les bases d'une première théorie statistique de l'information visant non pas l'information en tant que telle, mais la quantité d'information. Lorsque l'on parle de quantité d'information et de mesure, on pense à la notion de contenu ou de valeur de l'information. Sa théorie trouve aujourd'hui des applications dans tous les réseaux et systèmes de communication et sort du cadre de cet ouvrage.

2. Les systèmes de traitement automatique de l'information

Nous avons vu que pour pouvoir prendre une décision d'évolution d'un système (lancement d'une opération, exécution d'une action programmée, élaboration d'un résultat, etc.), il est nécessaire de disposer d'informations sur l'état du système et de son environnement.

Deux cas doivent être distingués selon la responsabilité de la décision.

● Cas 1 : La décision relève de la responsabilité humaine

L'opérateur doit alors pouvoir disposer d'informations sensorielles (visuelles, auditives, etc.) reçues directement ou par l'intermédiaire de dispositifs artificiels, des capteurs, qui transmettent ces informations sous forme visuelle (voyants, indicateurs à aiguille, messages sur écrans et afficheurs, etc.) ou sonore (avertisseurs, messages vocaux, etc.).

À partir de ces informations, l'homme prend des décisions qu'il traduit sous forme d'actions ou de consignes pour le système.

EXEMPLE

Les informations nécessaires à l'utilisateur d'un store sont :

- des informations physiques sur le niveau d'ensoleillement ou la force du vent, qu'il exploite pour décider s'il déroule ou enroule la toile ;
- des informations temporelles sur la saison ou la période de la journée, qu'il exploite pour décider de la mise en service ou hors service du store ;
- des informations sur l'état du système, qu'il exploite pour l'utiliser et parfois pour décider d'actions de maintenance ou de réglage.

● Cas 2 : La décision relève du système de traitement de l'information

Il doit alors disposer, par l'intermédiaire de capteurs, d'informations sur l'état du système et de son environnement, sur les consignes émises par l'opérateur, sur les messages éventuels communiqués par d'autres systèmes (informations d'entrée) (Figure 2.3).

Dans ce cas, l'information, élément de connaissance pouvant prendre plusieurs valeurs différentes, nécessaire au système de traitement pour pouvoir prendre une décision, est appelée souvent variable.

À chaque type d'information logique, continue et numérique, correspond donc une variable du même type, respectivement : logique (ou binaire), analogique et numérique.

Le traitement de ces informations par la commande lui permet d'émettre les ordres de fonctionnement, les signalisations à destination de l'opérateur, ou encore d'émettre des messages vers d'autres systèmes de traitement de l'information (informations de sortie).

EXEMPLE

Les informations nécessaires à la commande du store Somfy pour élaborer les actions prévues par le modèle de fonctionnement sont :

- des consignes de l'opérateur (marche/arrêt, montée/descente manuelles);
- des comptes rendus des capteurs disposés sur la partie opérative (toile sortie, toile enroulée);
- des comptes rendus des capteurs d'environnement (ensoleillement, vent).

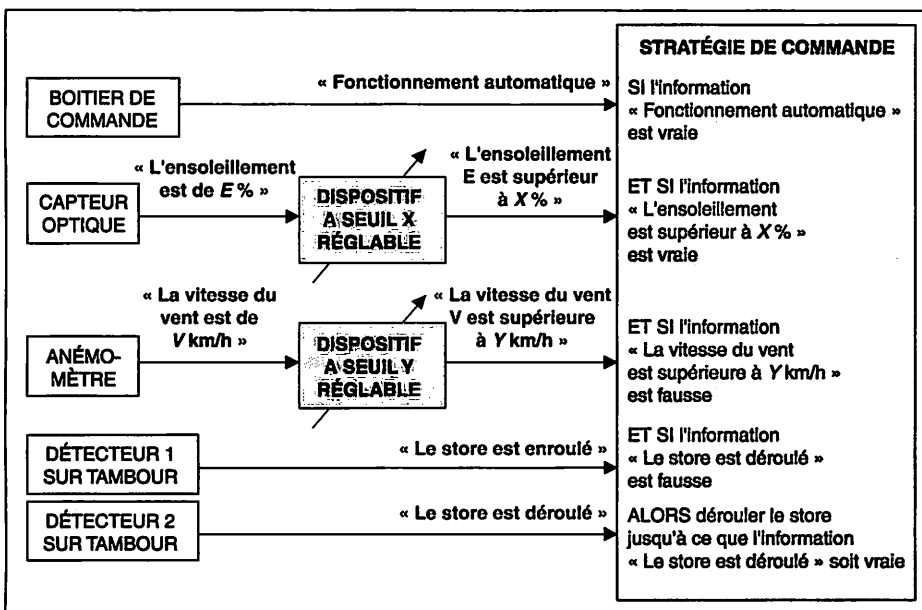


Figure 2.3. Informations utilisées par la commande automatique du store.

Dans le cas où la décision relève de la responsabilité humaine, si l'utilisateur du store souhaite communiquer à une autre personne par écrit (Figure 2.1) l'heure à laquelle il pourra mettre en service le store, plusieurs choix s'offrent à lui. L'heure peut être écrite sous la forme : de chiffres (12 h), de lettres (douze heures), de chiffres romains (XII heures) ou même d'un dessin. Ces différentes formes s'expriment avec des symboles et des nombres de symboles différents (deux symboles choisis parmi les chiffres de 0 à 9, cinq symboles choisis parmi les 26 lettres de l'alphabet, ...).

La grande majorité des systèmes de traitement automatique de l'information actuels, pour des raisons purement technologiques (utilisation de composants logiques et numériques), ne peuvent traiter qu'une information exprimée sous forme binaire. Les codes utilisés par les systèmes de traitement sont des codes à deux symboles. Généralement, pour les représenter, on utilise les deux premiers chiffres de notre système décimal, soit 0 et 1.

Contrairement à un système de traitement de l'information, l'homme est capable « naturellement » de reconnaître le type de codage employé. À la vue de symboles utilisés, (12 h, XII heures, ..), il est possible de leur attribuer une signification.

En revanche, comment donner un sens à cette information binaire qui prend la forme d'une suite de 0 et de 1 pour être traitée par une commande de système automatique ?

Que représente : 000100 ? Un nombre, une lettre, un dessin, ... ?

Que ce soit pour l'homme et surtout la machine, il n'est pas donc pas possible d'attribuer une signification à cette représentation sans savoir comment elle a été codée.

Coder une information, c'est donc lui associer un symbole ou une combinaison de symboles qui permet de la conserver, de la traiter ou de la communiquer.

II. Les codes binaires

Depuis l'antiquité, les hommes ont inventé des méthodes de plus en plus sophistiquées pour dénombrer, puis compter et enfin calculer. La numération moderne telle que nous la connaissons est assez récente au regard de l'histoire, la définition générale des systèmes de base m , avec m entier ≥ 2 ayant été présentée en 1654 à l'Académie des sciences par le mathématicien français Blaise Pascal. D'une manière simple, la base d'un système de numération est le nombre de chiffres qu'utilise ce système.

1. Représentation des nombres et systèmes de numération

a) Bases de numération (Figure 2.4)

Si on utilise une variable binaire, on ne dispose que de deux chiffres, ou « bits » (contraction de *Binary Digits*), qui sont 0 et 1. Il s'agit du système de numération dit « en base 2 ».

Si on utilise les dix chiffres du système décimal (0, 1, 2, 3, 4, 5, 6, 7, 8, 9), il s'agit d'une numération « en base 10 ».

Si on utilise les dix chiffres du système décimal (0, 1, 2, 3, 4, 5, 6, 7, 8, 9), plus les lettres A, B, C, D, E, F, il s'agit du système de numération « en base 16 », ou hexadécimal. Ce dernier système de numération est intéressant en informatique, car il permet de discriminer seize combinaisons avec quatre bits, sans combinaison inutilisée.

b) Expression des nombres

L'utilisation d'un seul caractère limite la discrimination à un nombre de combinaisons égal à la base. Pour augmenter la discrimination on utilise une association de plusieurs caractères, un nombre, selon une règle simple de « codage », soit :

$$N(B) = a_n \cdot B^n + a_{n-1} \cdot B^{n-1} + \dots + a_1 \cdot B^1 + a_0 \cdot B^0$$

avec $B =$ base et $a_i < B \quad (i = 1 \text{ à } n)$

EXEMPLE

$$923 (10) = 9 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0 = 900 + 20 + 3$$

Les caractères ou symboles a_n sont les chiffres.

Le nombre de combinaisons est égal à B^d , avec $B =$ base, $d =$ nombre de digits. Par exemple : 4 bits, en base 2, donnent $2^4 = 16$ combinaisons, deux chiffres décimaux en donnent $10^2 = 100$, soit la possibilité d'exprimer les nombres de 0 à 99.

Décimal	Binaire pur	Hexadécimal
00	0000	00
01	0001	01
02	0010	02
03	0011	03
04	0100	04
05	0101	05
06	0110	06
07	0111	07
08	1000	08
09	1001	09
10	1010	0A
11	1011	0B
12	1100	0C
13	1101	0D
14	1110	0E
15	1111	0F
16	10000	10
17	10001	11

Figure 2.4. Systèmes courants de numération.

c) Codage à l'aide d'une variable binaire

Toute information logique peut être codée à l'aide d'une variable binaire (ou logique).

EXEMPLE

Coder l'information logique « lampe allumée » :
soit « a » la variable logique associée à cette information,
alors : $a = 1 \Leftrightarrow$ lampe allumée
 $a = 0 \Leftrightarrow$ lampe éteinte.

d) Codage à l'aide d'un « mot »

Pour conserver, traiter et communiquer une information continue ou numérique, une variable binaire ne convient plus. Une solution consiste à utiliser un codage à l'aide d'un « mot ».

● Constitution d'un mot

Un **mot** est un groupement de plusieurs variables binaires appelés « bits » (Figure 2.5.). Compte tenu des composants technologiques de traitement utilisés, la longueur du mot est généralement (mais pas toujours) un multiple de 8 bits : soit 32, 64, 128. Il s'agit du **format** du mot.

Un groupe de 8 bits est appelé octet ou **BYTE** en anglais

La position de chaque bit dans le mot est appelée « rang ».

Dans le cas de la représentation d'un nombre entier positif en base 2 (voir paragraphe 2.1.) à l'aide d'un mot, on appelle « poids » d'un bit de rang N dans le mot la valeur 2^N . Le poids du bit de rang 5 est 2^5 soit 32.

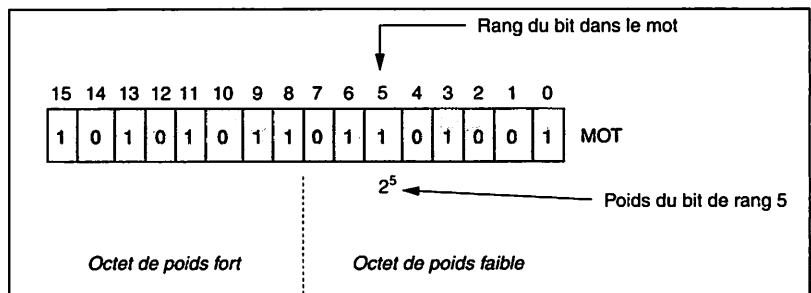


Figure 2.5. Exemple de mot de seize bits, soit deux octets.

On peut noter que le passage en base 16 est aisé puisque à chaque groupe de quatre bits correspond un symbole de 0 à F. Par exemple, le mot de 16 bits noté 1111 0010 1110 1000 en binaire correspond à F2E8 en hexadécimal. Cette représentation est souvent utilisée pour coder l'adresse mémoire à laquelle a été sauvegardé un mot particulier dans un système de traitement de l'information.

● **Les tailles des mémoires** informatiques sont traditionnellement indiquées en kB (ou Ko), MB, GB, etc. Dans ce cas, le « B » signifie Byte et non pas Bit, 1 kB se lisant « un kilo octet ». Un kilo-octet correspond à 1024 octets ($1 K = 2^{10}$) et non pas 1000 comme on pourrait l'imaginer. De même, un méga-octet ($1 M = 2^{20}$) correspond à 1 048 576 octets, un giga-octet ($1 G = 2^{30}$) à 1 073 741 824 octets et ainsi de suite.

● **Capacité de codage d'un mot** : un mot de N bits permet de coder 2^N éléments (ou possibilités).

● Les codes numériques et alphanumériques rencontrés dans les systèmes automatiques sont très nombreux :

- le code binaire pur ;
- le code binaire réfléchi (code Gray) ;
- le code binaire codé décimal (BCD) ;
- le code ASCII (*American Standard Code for Information Interchange*) et sa version étendue UNICODE ;
- les codes à barres (EAN, 5-1 entrelacé, etc.) ;

Le passage d'un code à un autre s'appelle transcodage et c'est une opération courante (voir FICHE MÉTHODE 2).

2. Le code binaire naturel ou code binaire pur

Il s'agit d'un code particulier, car il correspond précisément à la représentation d'un nombre entier et positif exprimé en base 2. En automatismes ce code pose des problèmes techniques particuliers (voir chapitre 3). Il est utilisé pour :

- repérer les emplacements de mémoires internes ainsi que ceux des entrées sorties (en pratique, on utilise plutôt la notation hexadécimale) ;
- effectuer des opérations logiques comme le masquage sur des mots en mémoire ;
- coder les digits du BCD ;
- coder des informations numériques provenant de capteurs à sortie numérique ;
- etc.



Il ne faut pas mélanger la notion de «code binaire» avec celle de «représentation d'un nombre binaire» utilisée pour la numération et les opérations arithmétiques (voir FICHE MÉTHODE 2).

Les codes binaires ne permettent généralement pas de réaliser des calculs : ils sont utilisés pour la transmission et la sauvegarde des informations. Par ailleurs, et afin de réaliser des calculs arithmétiques au sein des microprocesseurs, les nombres binaires sont représentés sous une forme normalisée très complexe (standard flottant IEEE 754), et non pas en binaire naturel.

La génération d'un code binaire naturel est très simple, sa **périodicité** étant remarquable. Par exemple, les sept premiers nombres codés sur quatre bits (trois suffiraient, B3 étant toujours nul) possèdent la représentation binaire suivante (Figure 2.6).

Valeur décimale	B3 (huit 0, huit 1, etc.)	B2 (quatre 0, quatre 1, etc.)	B1 (deux 0, deux 1, etc.)	B0 (un 0, un 1, etc.)
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0

Figure 2.6. Les sept premiers nombres codés sur quatre bits.

3. Le code GRAY (binaire réfléchi)

Le code GRAY permet de remédier à un défaut du code binaire lorsque l'on cherche à coder des états successifs d'informations provenant d'organes technologiques.

Un exemple typique est celui du codeur absolu, dont la fonction est de fournir un code binaire représentant la position angulaire de son axe (voir FICHE TECHNIQUE 7).

En prenant, par exemple, un codeur à 4 bits permettant de coder $2^4 = 16$ positions par tour, comprises entre 0000 et 1111. Ce codeur possède 4 détecteurs optiques associés à 4 pistes possédant des zones noires ou blanches formant un code binaire à quatre bits B0, B1, B2 et B3. Dans le cas d'un codage en binaire naturel, il risque de se produire un aléa lors du passage d'une position à une autre dans le

cas ou plusieurs bits changent simultanément d'état : par exemple lors du passage de la position 5 (0101) à la position 6 (0110), les deux capteurs associés à B0 et B1 sont supposés changer d'état simultanément, ce qui est impossible quelle que soit la technologie employée. En réalité, les capteurs vont changer d'état *successivement* dans un ordre inconnu à priori et le système va lire, pendant un instant court mais suffisant pour être pris en compte, une position erronée : soit B0 évolue le premier et, transitoirement, la position lue sera 0100 (4), soit B1 évolue le premier et, transitoirement, la position lue sera 0111 (7). *L'information fournie par le codeur sera temporairement fautive* (Figure 2.7).

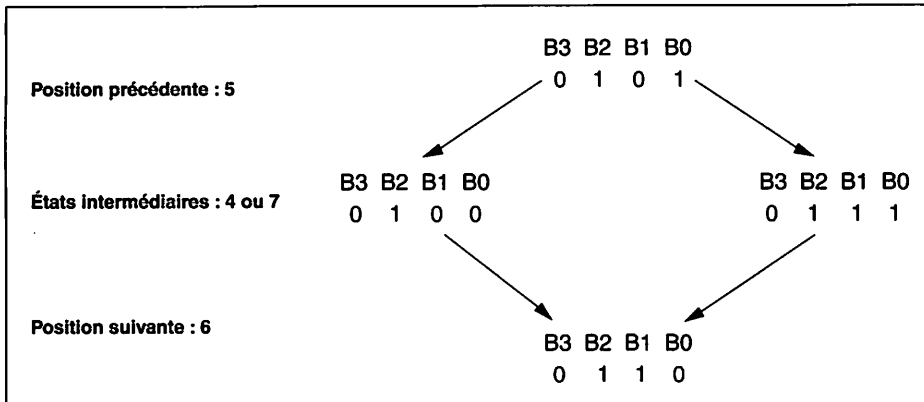


Figure 2.7. Aléas technologiques lors d'un changement de valeur d'un code binaire pur.

Dans toutes les situations où plusieurs bits sont supposés changer d'état simultanément, un aléa est probable. Les différentes zones de risque d'aléa sont représentées sur la figure 2.8.

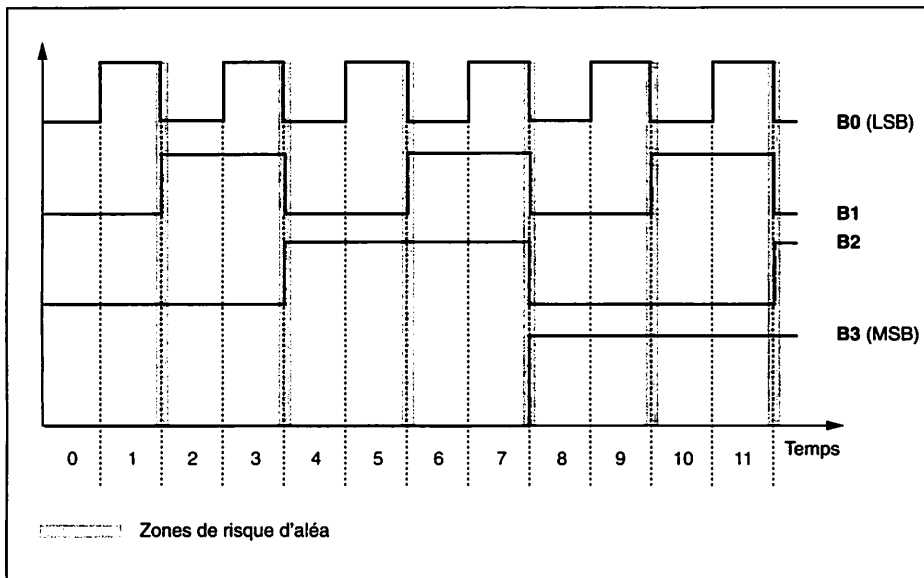


Figure 2.8. Zones de risque d'aléa lors d'un changement de valeur avec un code binaire pur à 4 bits.

Une solution élégante à ce problème consiste à utiliser un code binaire possédant la propriété suivante : entre deux nombres successifs du code, il ne se produit qu'un seul changement d'état de bit à la fois : c'est le code binaire réfléchi, dont le nom provient de la symétrie qui préside à sa construction et que l'on peut observer sur un disque de codeur (Figure 2.10).

On remarque bien que, dans la figure 2.9 le passage de la zone 5 à la zone 6 ne provoque qu'un seul changement de bit : code 0111 à 0101.

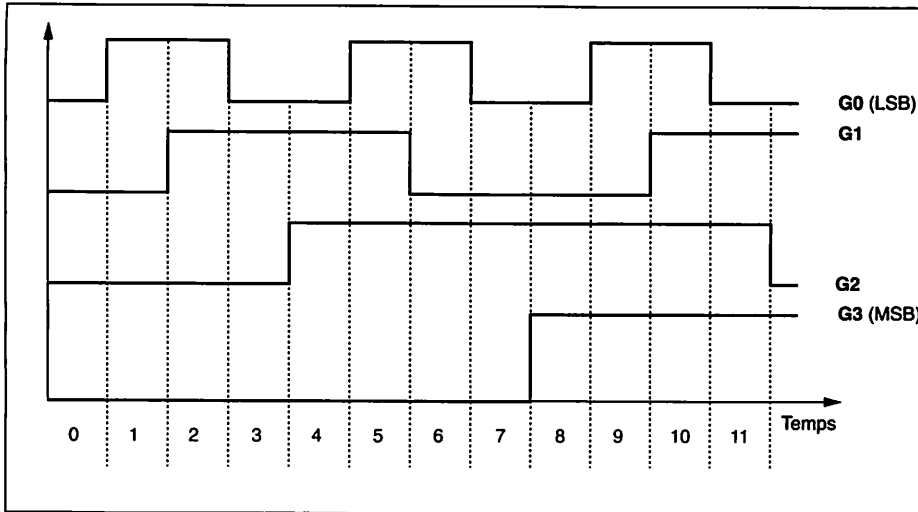


Figure 2.9. Disparition des zones de risque d'aléa lors d'un changement de valeur avec un code GRAY.

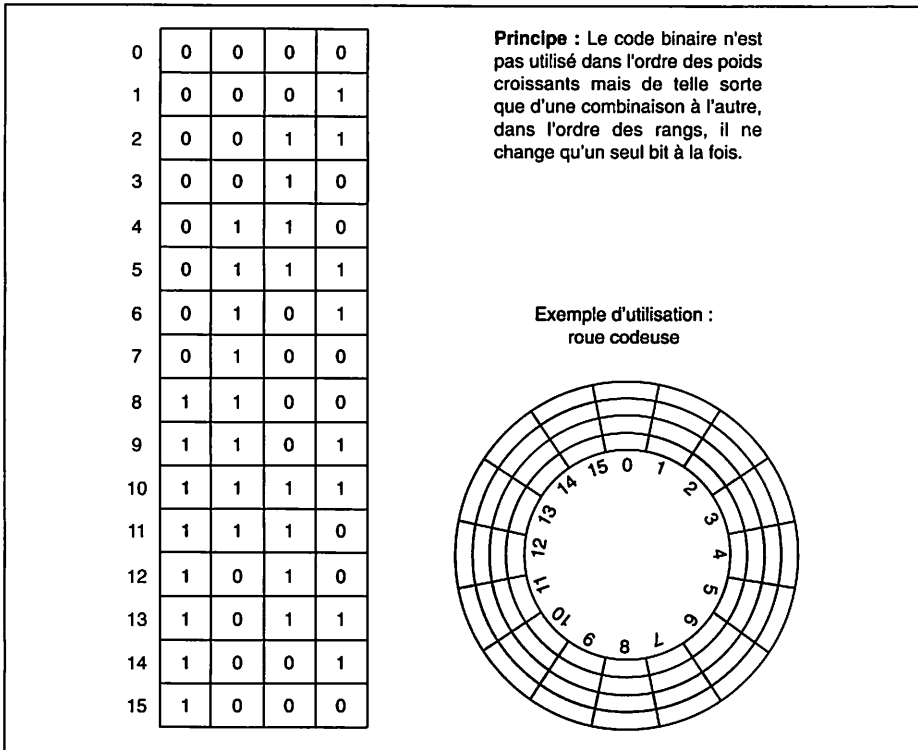


Figure 2.10. Codage GRAY sur 4 bits d'un disque de roue codeuse.

REMARQUE

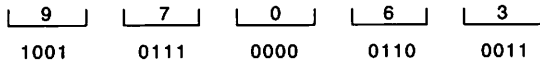
Certains codeurs absolus fournissent une sortie exploitable en binaire naturel. Ils possèdent une piste supplémentaire qui permet de masquer les zones de changements de valeur : les valeurs changent simultanément lors de la disparition du masquage.

4. Le code BCD (Binary Coded Decimal)

Appelé également DCB (décimal codé binaire), ce code est un bon intermédiaire entre le binaire et le décimal. Il est utilisé en particulier pour les entrées/sorties d'automates programmables telles que les roues codeuses et les afficheurs 7 segments. Le principe consiste à coder chaque chiffre décimal en binaire sur quatre bits de 0000 à 1001.

EXEMPLE

Le nombre décimal 97063, possédant cinq chiffres, se code en BCD sur $5 \times 4 = 20$ bits, 10010111000001100011 suivant la décomposition ci dessous :



Une application du codage BCD est l'afficheur à 7 segments : c'est un composant possédant quatre entrées logiques par chiffre, chaque chiffre étant codé en binaire. Lorsque l'on utilise N afficheurs, un nombre est représenté par la combinaison des N chiffres ce qui revient précisément à un codage BCD. Dans le cas de deux afficheurs, les valeurs correspondantes peuvent être comprises entre 00 et 99 (Figure 2.11).

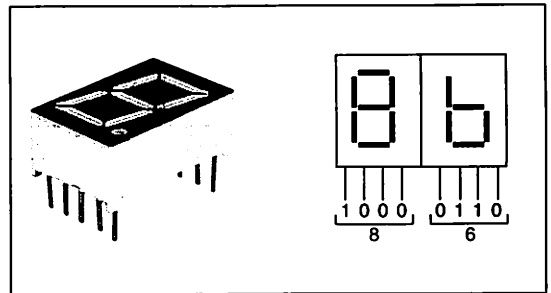


Figure 2.11. Codage BCD sur deux afficheurs à 7 segments.

Le code BCD présente deux inconvénients :

- il n'exploite pas toutes les possibilités du binaire, car il code seulement 10 chiffres sur 4 bits au lieu des 16 possibles (de 0000 à 1001 au lieu de 0000 à 1111) ;
- il faut le convertir pour effectuer des calculs arithmétiques, complexes sinon.

5. Les codes ASCII et UNICODE

Le code ASCII (American Standard Code for Information Interchange) est une norme quasi universelle pour les transmissions d'informations locales : ordinateur vers imprimante, automate programmable vers terminal d'exploitation, clavier vers processeur, etc. Le tableau (Figure 2.12.) fournit les codes ASCII exprimés en hexadécimal, le codage s'effectuant sur 2 caractères hexadécimaux. Le code le plus petit correspond à NULL (00 = 0000 0000 en binaire), le code le plus grand correspondant à DEL (7F = 0111 1111). Sept bits sont donc utilisés pour le codage, le huitième servant à la détection d'erreur (bit de parité).

Le code ASCII s'est rapidement révélé insuffisant, de nombreux pays utilisant des symboles non représentables au moyen du code ASCII (plus de 20000 idéogrammes Han, pour les Chinois et les Japonais). Un code plus complet, appelé UNICODE, a donc été défini, ce qui a donné lieu à la norme internationale IS 10646. Il possède 16 bits de largeur et permet de coder 65536 caractères différents. Il est accepté par les systèmes d'exploitation, applications et langages actuels tels que Windows, JAVA, etc. Toutefois, le problème n'est pas résolu de manière définitive, puisqu'on estime à 200 000 le nombre de symboles utilisés par l'ensemble des langues du monde.

Code Hexa	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11
Caractère ASCII	NULL	SOH	STX	ETX	EOT	ENQ	ACK	BELL	BS	HT	LF	VT	FF	CR	SO	SI	DLE	DC1

Code Hexa	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	40	41	42
Caractère ASCII	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US	SPACE	@	A	B

Code Hexa	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54
Caractère ASCII	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T

Code Hexa	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66
Caractère ASCII	U	V	W	X	Y	Z	[\]	^	-	'	a	b	c	d	e	f

Code Hexa	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78
Caractère ASCII	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x

Code Hexa	79	7A	7B	7C	7D	7E	7F
Caractère ASCII	y	z	{		}	~	DEL

Figure 2.12. Table des codes ASCII.

III. Le signal support de l'information

1. Le signal

La communication d'une information suppose un émetteur envoyant un message à un récepteur. Ce message qui contient l'information est transmis au moyen d'un signal codé en direction du récepteur qui est capable de le décoder. Le support de cette information est un signal qui peut prendre diverses formes physiques. Cette forme physique (onde, signal électrique, ...) implique une quantité d'énergie aussi petite que la technologie le permet, mais jamais nulle.

a) Nature et niveau d'un signal

Dans les systèmes automatiques, la transmission du signal s'effectue en utilisant souvent les formes de l'énergie suivantes : tensions continues (5, 12, 24 volts) ou alternatives en électricité, pression et débit d'un fluide hydraulique ou pneumatique, ondes radio ou lumineuse (infrarouges par exemple).

Historiquement, les composants électromécaniques dans les circuits ont été rapidement supplantés par des composants électroniques d'abord à tubes, puis à semi-conducteurs (transistor), fonctionnant sur un principe similaire (passant ou bloquant) et des circuits intégrés (microprocesseurs) qui ont permis l'explosion du nombre d'opérations et un abaissement des coûts. Cependant, de nombreux circuits de sécurité utilisent, de nos jours encore, les composant électromécaniques.

Dans la pratique, les états 1 et 0 correspondent à une valeur comprise dans une fourchette. Les fiches des constructeurs précisent ces valeurs.

REMARQUE

Toutefois, d'autres méthodes sont utilisées pour coder des 0 et des 1, en particulier dans les dispositifs de communication utilisant le réseau téléphonique (Internet ADSL), des fibres optiques (Câble) ou des ondes électromagnétiques aériennes (Téléphone portable, Télévision numérique, Réseaux locaux sans fil). Ces méthodes, complexes, consistent principalement à moduler un signal de manière à distinguer deux états.

Ces valeurs sont d'ailleurs différentes selon les technologies et selon qu'il s'agit d'une entrée ou une sortie de composant.

EXEMPLE

Circuit TTL (Figure 2.13).

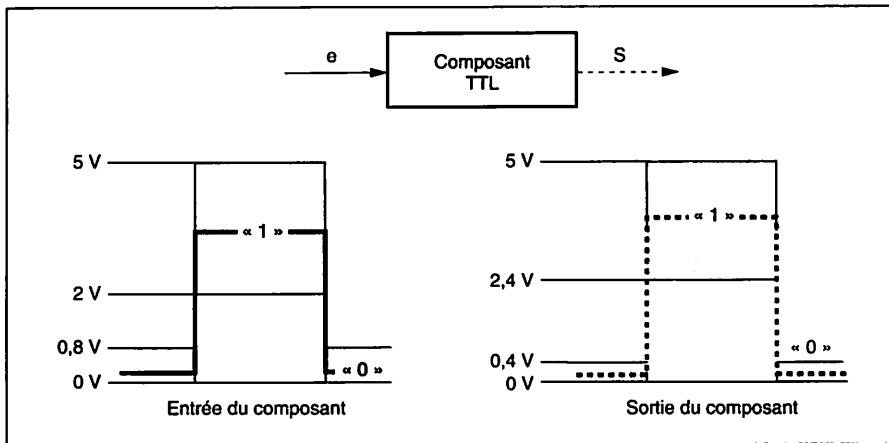


Figure 2.13. Allure typique des seuils logiques d'un composant de technologie TTL.

Hors ces plages le signal représente un état indéterminé. Cette situation est souvent révélatrice d'une anomalie (mauvaise utilisation, défaillance du composant), etc.

b) Logique positive, logique négative, logique complémentaire

La correspondance est appelée **logique positive** : la valeur 1 est attribuée au potentiel le plus élevé. C'est la forme la plus naturelle et la plus fréquente.

Vrai	1
Faux	0

On rencontre parfois une autre correspondance appelée **logique négative** : la valeur 0 est attribuée au potentiel le plus élevé.

Vrai	0
Faux	1

Une difficulté avec ces deux cas est l'ambiguïté de l'absence de tension qui peut correspondre à un contact ouvert (état 0) mais également à une absence de tension pour cause de panne.

Une autre technique est parfois utilisée, celle de la logique complémentaire qui consiste à utiliser deux sources de potentiel (une positive (+ V) notée 1 et une négative (- V) notée -1).

La figure 2.14 illustre les divers cas rencontrés.

	Logique naturelle	Logique positive	Logique négative	Logique complémentaire	Logique informatique
Fonction vraie	Vrai	1	0	1	True
Fonction fausse	Faux	0	1	-1	False

Figure 2.14. Logique positive et logique négative.

2. Typologie des signaux supports des informations

a) Les trois types de signaux

Les signaux peuvent être classés de différentes manières.

Pour l'automaticien, on distingue trois cas au sein de deux grandes familles :

● **Les signaux discrets ou digitaux**

Les signaux discrets sont binaires : ils ne peuvent prendre que deux valeurs 0 et 1. Suivant le nombre de variables binaires que véhicule le signal, une ou plusieurs, on parle de signal Tout Ou Rien ou de signal numérique.

– **Signal Tout Ou Rien (TOR)** : il ne contient qu'une variable binaire et ne peut prendre que les valeurs 0 ou 1. Il supporte des informations logiques du type « vrai ou faux » : « la porte du palettiseur est fermée », « un carton est arrivé en fin de convoyeur », etc. Les signaux TOR se rencontrent très fréquemment dans les automatismes : les détecteurs, les boutons poussoir, etc. fournissent des signaux de ce type.

– **Signal numérique** : il transporte plusieurs variables binaires qui permettent de réaliser un code. Plus le nombre de variables est important, plus la quantité d'information est grande. Il permet de transporter une information plus complète et discriminante, comme une quantité (de couches, de flacons remplis, etc.) ou un rang (ordre de rangement).

● **Les signaux continus ou analogiques**

Un signal analogique peut varier de manière continue. Il s'agit le plus souvent d'une tension électrique dont la valeur évolue selon une loi connue, ce qui permet d'en extraire l'information. Le cas typique est celui d'une grande partie des capteurs industriels (température, pression, vitesse, etc.) qui fournissent une tension normalisée et généralement proportionnelle à une grandeur physique que l'on désire connaître. Il est à noter que, les systèmes de traitement étant de nature discrète, la tendance actuelle est de convertir systématiquement les signaux analogiques en signaux numériques avant de les traiter.



Il ne faut pas confondre l'information et le signal qui la porte : les signaux analogiques, lorsqu'ils sont modulés, sont tout à fait capables de transporter des informations numériques. C'est le cas des signaux numériques qui transitent par le réseau INTERNET et qui sont transmis sur des lignes téléphoniques analogiques, y compris en mode ADSL.

Type de signal	SIGNAL DISCRET OU DIGITAL		SIGNAL ANALOGIQUE OU CONTINU
	Signal TOR ou binaire	Signal numérique	
Nombre d'états que le signal peut prendre	Deux états	Nombre fini d'états	Nombre infini d'états
Exemples d'informations pouvant être portée par le signal	Présence/absence Vrai/faux Fait/Non fait	Rang dans une liste Note d'un devoir Code binaire	État d'une grandeur physique Code numérique

Figure 2.15. Les trois types de signaux.

b) Information portée par un signal Tout Ou Rien

Ce type d'information est utilisé lorsque l'on désire connaître une possibilité parmi deux : présence ou non présence d'un objet, dépassement ou non d'un seuil de pression, de vitesse, etc. Le codage retenu pour une information Tout Ou Rien (TOR) est obtenu par une seule variable portée par un signal pouvant prendre seulement deux états, 0 ou 1.

Les informations TOR se rencontrent en plusieurs endroits dans la chaîne d'information, que ce soit en sortie d'un détecteur électronique ou encore en entrée

du système de traitement de l'information. Sur la figure 2.3, les informations « l'ensoleillement est supérieur à X% », « la vitesse du vent est supérieure à Y km/h », « le store est enroulé » et « le store est déroulé » sont portées par des signaux Tout Ou Rien

La figure 2.16 montre un détecteur photoélectrique fournissant l'information TOR « présence flacon sur tapis roulant » qui sera exploitée par la partie commande.

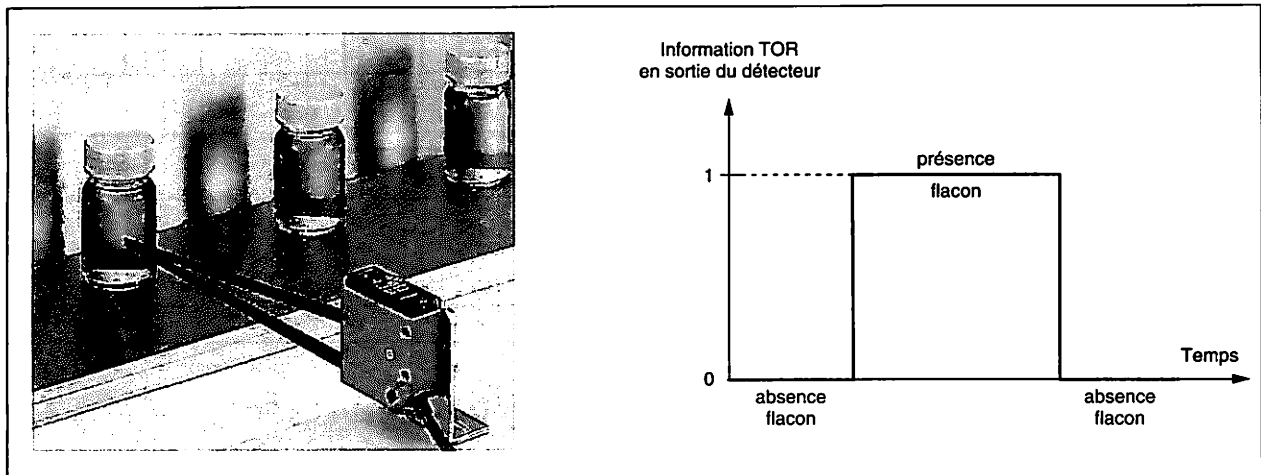


Figure 2.16. Détecteur TOR photo électrique de type proximité (doc. OMRON).

c) Information portée par un signal numérique

L'information numérique est associée souvent à un nombre décimal ($\in \mathbb{Q}$), son support est un ensemble de signaux TOR. L'information numérique discrète découle de la nécessité de discriminer une même information à l'aide d'un nombre restreint de variables. Une information numérique est le résultat d'un codage.

EXEMPLE

L'information de la dépose de la dernière couche de cartons dans le palettiseur est une information source logique (la dernière couche est présente) qui est utile pour l'évolution des tâches du palettiseur. Elle peut être traduite par une image numérique :

- directe, si s'allument tour à tour des voyants correspondant à chaque couche ;
- codée, si un afficheur indique une combinaison de chiffres à l'utilisateur, ou si un compteur indique un code binaire à la partie commande.

Une information numérique est le résultat d'un codage. Par exemple, pour coder numériquement un déplacement de 1 m avec une résolution de 0,01 mm il faudra 100 000 combinaisons binaires, nécessitant au minimum 17 bits de code (16 bits, ne permettant que 65 536 possibilités, sont insuffisants).

L'information numérique peut être élaborée de plusieurs manières : directement, par comptage ou bien par conversion analogique-numérique au moyen d'un CAN (Convertisseur Analogique Numérique).

EXEMPLE

Dans le cas d'une mesure de position angulaire, l'information numérique image de la position fournie par le capteur peut être obtenue :

- par comptage, le capteur fournissant un certain nombre d'impulsions par unité de temps : codeur incrémental ;
- après conversion de l'image analogique de la position par un CAN (Convertisseur Analogique Numérique) : resolver, etc ;
- directement, le capteur fournissant un code binaire sur N bits : codeur absolu.

Le codeur absolu représenté figure 2.17 donne la position de son axe sous la forme d'une information image numérique codée en GRAY sur 10 bits G0 à G9. Ce codeur permet de discriminer $2^{10} = 1024$ positions par tour, avec une résolution de $360/1024 = 0,35^\circ$.

(voir FICHE TECHNIQUE I)

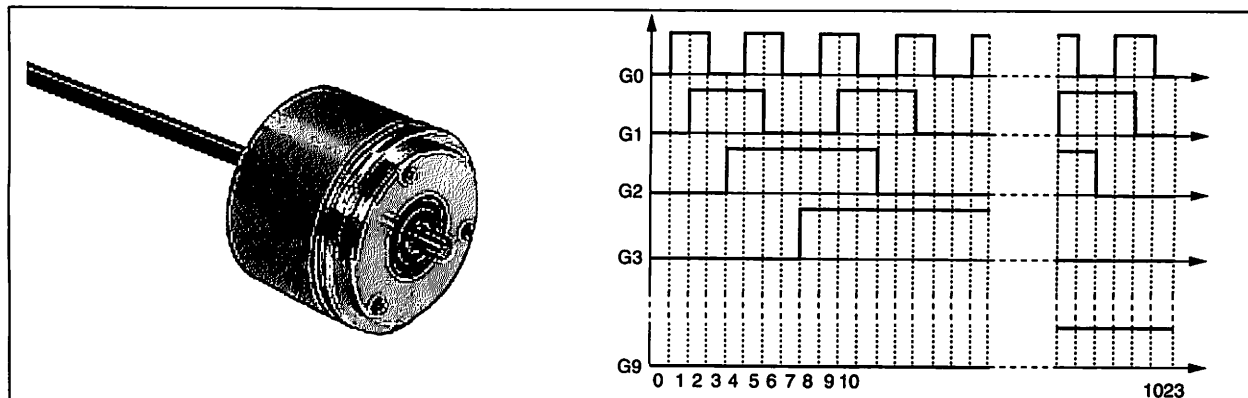


Figure 2.17. Codeur absolu à 10 voies (doc. Schneider).

d) Information portée par un signal analogique

Une telle information est utile lorsque l'on souhaite contrôler le comportement d'un système de manière continue en **commande proportionnelle**. Ainsi, la mesure d'une grandeur continue va consister en la transformation d'une information source telle que température, vitesse, pression, etc. en information image, le plus souvent sous forme d'un signal électrique.

EXEMPLE

L'information « l'ensoleillement est de 85 % » donnée par le capteur solaire du store Somfy, est une information portée par un signal continu ou analogique. Elle est ensuite traduite par l'électronique en un seuil qui est une information TOR.

Une information analogique est portée par un signal qui varie de manière continue, en relation avec l'évolution de la grandeur physique qui en est la source. D'une manière générale, les capteurs analogiques sont conçus pour être intégrés avec des automates programmables ou des systèmes d'acquisition de données, et fournissent un signal normalisé en sortie pleine échelle $\pm 50 \text{ mV}$, $\pm 1 \text{ V}$, $\pm 5 \text{ V}$, $\pm 10 \text{ V}$ pour une tension ou $4\text{-}20 \text{ mA}$, $0\text{-}20 \text{ mA}$ pour un courant, directement proportionnel au signal d'entrée.

La figure 2.18 illustre la variation continue de tension (information image analogique) proportionnellement à la distance (information source analogique) de l'objet, dans le cas d'un capteur optique à rayon laser.

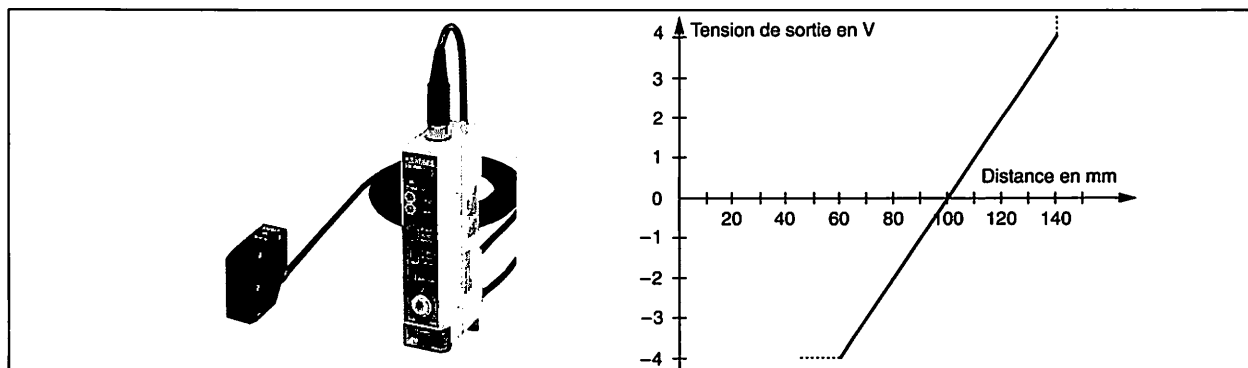


Figure 2.18. Capteur laser de déplacement à sortie analogique (doc. KEYENCE).

EXEMPLE

Système industriel de vision.

La figure 2.19 montre que les trois types d'information cohabitent au sein d'un système de vision industriel. Le contrôleur échange des informations avec les divers composants qui lui sont associés :

- Informations TOR

- entrée de synchronisation provenant d'un détecteur ;
- sortie vers un indicateur ou une sonnerie.

- Informations Numériques

- entrée provenant de la console de programmation ;
- liaison série RS 232 avec l'ordinateur.

- Informations Analogiques

- entrée provenant de la caméra CCD ;
- sortie en direction du moniteur.

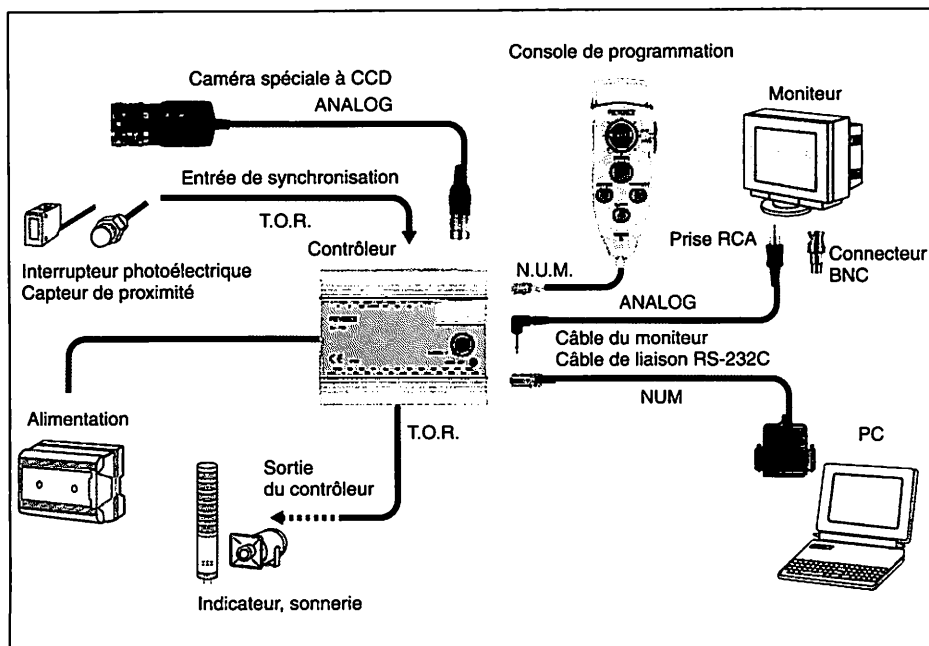


Figure 2.19. Différents types d'informations autour d'un système industriel de vision (Doc KEYENCE).

Résumé

- Pour fonctionner, un système automatique doit pouvoir acquérir, traiter et communiquer des informations qui sont portées par des signaux. Ces derniers peuvent être classés en trois types :

- les variables logiques permettant de distinguer une possibilité parmi deux (absence/présence, vrai/faux, etc.) ;
- les variables numériques fournissant une valeur ou un code (rang dans une liste, valeur entière, etc.) ;
- les signaux analogiques évoluant de manière continue (température, tension, etc.).

- Les systèmes de traitement de l'information étant réalisés autour de composants à deux états (transistors), leur langage naturel est le binaire. Un mot est constitué de N bits (8, 16, 32, 64, etc.) dont la position est repérée par leur rang, le rang le plus faible se situant à droite. Les groupes de 8 bits sont appelés octets, un kilo octet correspondant à 1024 octets. Un mot de N bits permet de coder 2^N combinaisons.

- Pour communiquer les informations, un codage est nécessaire. Les principaux codes numériques et alphanumériques sont :
 - le binaire réfléchi (code Gray) qui permet de supprimer les aléas technologiques lors du passage d'une valeur à la suivante car un seul bit change d'état;
 - le binaire codé décimal (BCD) utilisé pour les afficheurs 7 segments et les roues codeuses;
 - le code ASCII (*American Standard Code for Information Interchange*) et un code plus large, l'UNICODE qui permettent d'échanger des caractères alphanumériques entre n'importe quels systèmes et composants.

Exercices

EXERCICE 1. La position de l'élévateur du palettiseur (voir FICHE SYSTÈME 2) est traduite par un capteur incrémental. Indiquer quelle est la nature de l'information qui parvient à la partie commande.

EXERCICE 2. Citer, pour les systèmes décrits dans les fiches systèmes, dix exemples d'information Tout Ou Rien et identifier les objets qui produisent l'image informationnelle correspondante.

EXERCICE 3. En utilisant les propriétés du développement polynomial, effectuer les changements de base ci-dessous :

$$101101_{(2)} \rightarrow D1_{(10)}$$

$$A732_{(16)} \rightarrow D2_{(10)}$$

EXERCICE 4. En effectuant des divisions successives, opérer les changements de base ci-dessous :

$$831_{(10)} \rightarrow H_{(16)}$$

$$230_{(10)} \rightarrow B_{(2)}$$

EXERCICE 5. Après avoir pris connaissance de la FICHE MÉTHODE 2 « effectuer un transcodage binaire », opérer le transcodage ci-dessous :

$$10101_{(2)} \rightarrow G_{(\text{gray})}$$

EXERCICE 6. Dans un automate programmable, les nombres entiers positifs sont codés en binaire naturel sur 16 bits. Quel est le plus grand nombre entier représentable dans ce cas ?

EXERCICE 7. Calculer le nombre de bits d'informations contenus dans un disque dur d'une capacité de 140 Giga octets.

EXERCICE 8. Décoder la séquence ASCII suivante (sans bit de parité) :

1000011 0100111 1000011 1010011 1010100 0100000 1001010
1010101 1010011 1010100 1000101

EXERCICE 9. Parmi les mots suivants, lesquels n'ont pas de sens en notation hexadécimale ?

BAC BADGE BAIE BEC CAFE DADA DEFI ECHEC EFFACE FACADE FEE

EXERCICE 10. Pour un codeur absolu, déterminer le nombre de pistes nécessaires pour obtenir une résolution minimale de une minute d'arc (un degré = 60 minutes d'arc).

3

La chaîne d'acquisition

I. Structure d'une chaîne d'acquisition

1. Acquisition de données

a) La chaîne d'acquisition

Pour prendre des décisions, les systèmes de traitement de l'information ou commandes de systèmes automatiques ont besoin d'informations issues de l'environnement humain et matériel, d'information sur l'état du produit ou de l'énergie sur lequel il agit et sur son propre état. Ces informations sont supportées par un signal (principalement) électrique qui doit être adapté aux caractéristiques de la commande (typologie de signal, niveau de tension, fréquence d'acquisition, protocole).

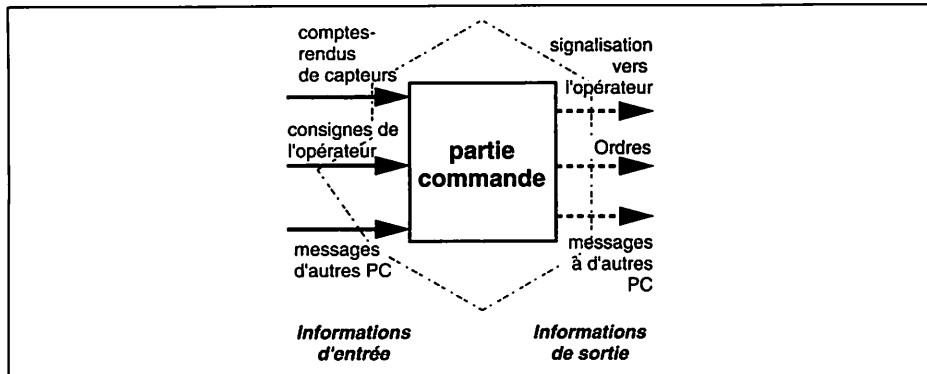


Figure 3.1. Catégories d'informations d'entrées utiles à la commande.

Dans le cas des comptes-rendus issus des capteurs et des consignes opérateurs, divers organes sont chargés de prélever la grandeur physique à mesurer (position, dimension, poids, vitesse, température, ...) ou la consigne pour la transformer en une grandeur exploitable par la commande : capteurs, interfaces homme - machine, systèmes numériques d'acquisition de données (caméra, microphone, scanner, codeurs, ...).

Fonctionnellement, il s'agit d'acquérir des grandeurs physiques pour les transformer en grandeurs utilisables directement à des fins de traitement.

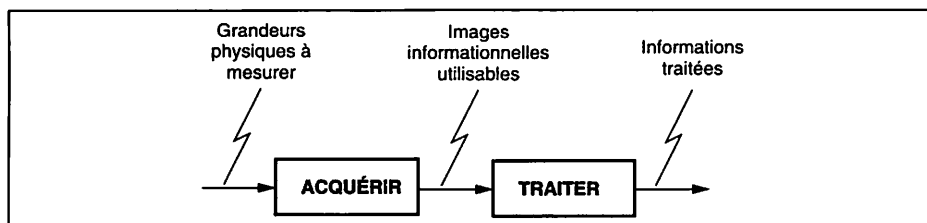


Figure 3.2. Fonction et situation de la chaîne d'acquisition.

b) La grandeur physique à mesurer

La grandeur physique à mesurer également appelée **mesurande**, n'est généralement pas directement exploitable par la commande du système automatique. C'est l'entrée ou stimulus du système d'acquisition.

Le mesurande peut garder une valeur stable ou peu rapidement variable (niveau de liquide dans une cuve). Cette valeur peut également varier à des fréquences plus ou moins importantes (détection d'un objet se déplaçant à grande vitesse, mesure de vibrations, ...).

On distingue habituellement quatre classes de fréquences :

TBF très basses fréquences (0 Hz à 250 Hz)

BF basses fréquences (250 Hz à 2 kHz)

MF moyennes fréquences (2 kHz à 10 kHz)

HF hautes fréquences (> 10 kHz)

L'étendue des valeurs courantes est définie par les valeurs extrêmes que peut prendre le mesurande de manière générale.

L'étendue des valeurs accidentelles est définie par les valeurs extrêmes que peut prendre le mesurande sous l'action de causes extérieures prévisibles ou non (phénomènes de faible durée).

2. Structure générale d'une chaîne d'acquisition de données

a) Les différentes fonctions assurées par la chaîne d'acquisition

La transformation d'un mesurande en une grandeur utilisable par une commande est réalisée grâce à un ensemble d'éléments assurant différentes fonctions permettant la saisie de l'information, sa transformation, sa mise en forme et sa transmission. Cet ensemble d'éléments ou de fonctions (Figure 3.3) est appelé chaîne d'acquisition.

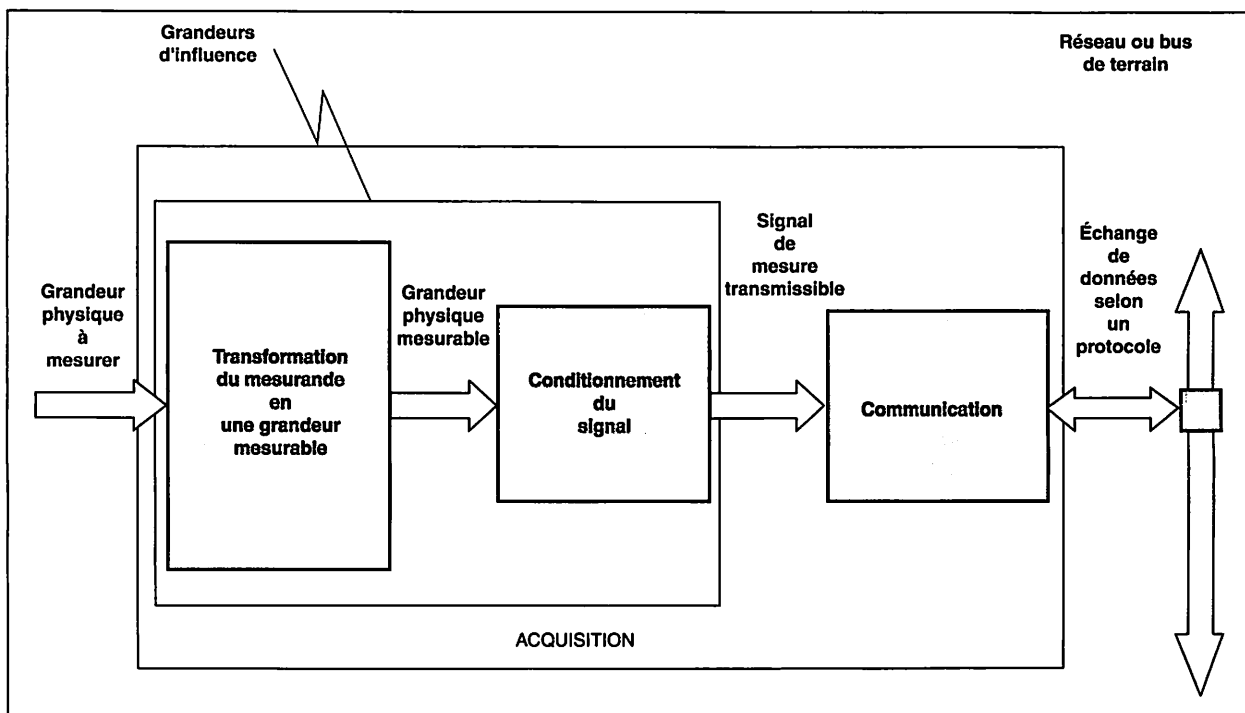


Figure 3.3. Exemple de structure d'une chaîne d'acquisition de données.

La chaîne d'acquisition inclut ainsi tout ou partie des fonctions suivantes :

Fonction	Description
<p>Transformer le mesurande en une grandeur mesurable <i>Elle consiste à transformer la grandeur à mesurer en une grandeur mesurable, généralement de nature électrique.</i></p>	<p>Cette fonction peut faire appel à de nombreuses sous-fonctions réalisées généralement par un corps d'épreuve et un élément de transduction. Le corps d'épreuve est un élément mécanique qui réagit sélectivement à la grandeur à mesurer. Il a pour rôle de transformer la grandeur à mesurer en une autre grandeur physique dite mesurable (membrane, ressort, barre de torsion, thermocouple, ...). Cette grandeur constitue la réaction du corps d'épreuve. L'élément de transduction est un élément sensible lié au corps d'épreuve. Il traduit les réactions du corps d'épreuve en une grandeur électrique constituant un signal de sortie.</p>
<p>Conditionner le signal <i>Elle consiste à mettre en forme le signal afin de l'adapter au système de traitement.</i></p>	<p>Le conditionnement consiste en général à :</p> <ul style="list-style-type: none"> - alimenter électriquement le capteur ; - mettre en forme et amplifier le signal de sortie ; - filtrer, corriger, traiter, convertir le signal (analogique/numérique, tension/fréquence, ...).
<p>Communiquer</p>	<p>Le développement des bus de terrains et des bus capteurs-actionneurs a conduit les constructeurs à proposer des solutions permettant à la chaîne d'acquisition de communiquer directement avec la commande en respectant un protocole donné.</p>
<p>Traiter</p>	<p>Bien que cette fonction soit généralement réalisée en aval du capteur par la partie commande ou par une électronique dédiée, elle est présente sur certains capteurs dits « intelligents », qui portent des fonctions avancées de traitement et de communication :</p> <ul style="list-style-type: none"> - auto configuration, auto étalonnage ; - auto surveillance et diagnostic ; - prise en compte des grandeurs d'influence ; - calculs ; - travail en réseau, ...

● La grandeur d'influence

Le signal de mesure doit être indépendant des grandeurs étrangères pouvant influencer sur la chaîne d'acquisition.

La plupart des grandeurs d'influence sont liées à l'environnement de cette chaîne d'acquisition :

- la température ;
- la pression environnante ;
- les vibrations mécaniques ou acoustiques, les chocs ;
- la position du capteur et sa fixation ;
- l'accélération et la pesanteur ;
- l'humidité, les projections d'eau, l'immersion, les ambiances corrosives ;
- les perturbations électromagnétiques ;
- les rayonnements nucléaires ;
- l'alimentation électrique du capteur, ...

b) Illustration de la fonction : transformer le mesurande en une grandeur mesurable

La figure 3.4 représente quelques corps d'épreuve typiques transformant un mesurande E en grandeur mesurable S :

- tube de bourdon et capsule anéroïde (ou barométrique), transformant une variation de pression ou de température en déplacement ;
- bilame transformant une variation de température en déplacement ;
- accéléromètre transformant une accélération en déplacement rectiligne ou angulaire ;

- potentiomètre transformant un déplacement angulaire en variation de résistance;
- jauge d'extensométrie transformant une déformation en variation de résistance.

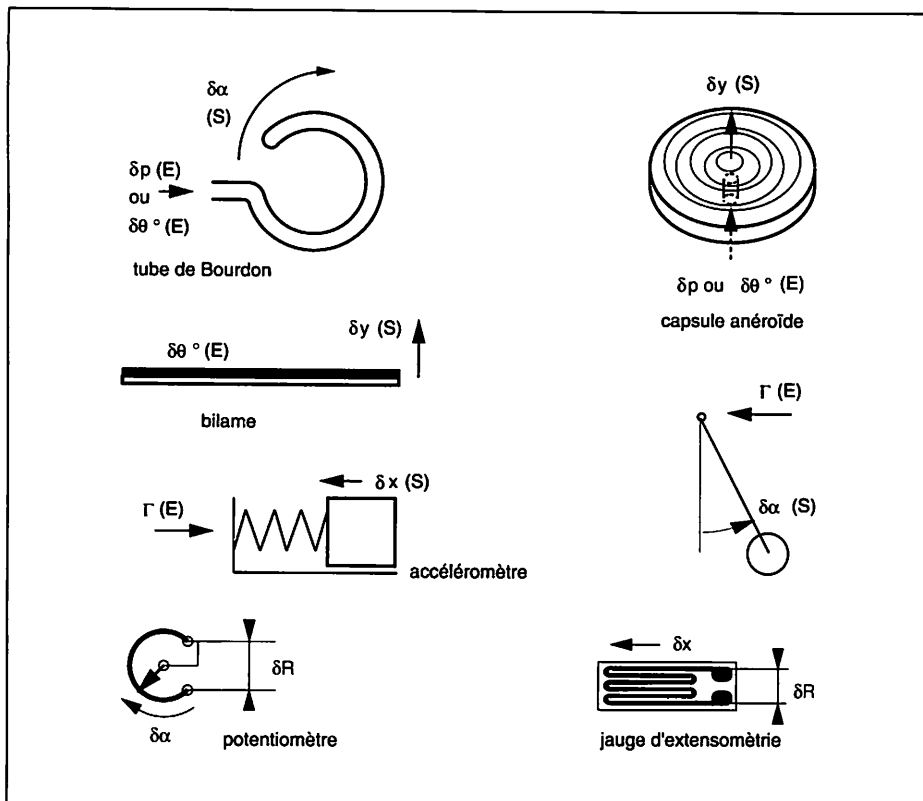


Figure 3.4. Corps d'épreuve produisant un déplacement ou une variation de résistance.

c) Les différents niveaux d'intégration

De nombreux éléments permettent de réaliser les diverses fonctions assurées par la chaîne d'acquisition de données : corps d'épreuve, transducteur, conditionneur, électronique de traitement et de communication, microcontrôleur, ...

Selon les applications et les contraintes technico-économiques exprimées dans le cahier des charges, ces éléments peuvent être intégrés partiellement ou totalement au sein de capteurs. Trois niveaux d'intégration différents sont illustrés figure 3.3 selon que le conditionneur et l'électronique permettant la communication soient intégrés ou non.

Parmi tous les vocables rencontrés suivant le niveau d'intégration, deux grandes catégories peuvent être identifiées : les détecteurs et les capteurs. Toutefois, ces termes recouvrent des sens différents selon les utilisateurs, ce qui est source de confusion. Par la suite, on conviendra d'utiliser les définitions suivantes :

<p>Capteur (du latin <i>captare</i>, chercher à prendre)</p>	<p>Dispositif qui délivre, à partir d'une grandeur physique, une autre grandeur, souvent électrique, fonction de la première et directement utilisable pour la mesure ou la commande. <i>La valeur portée par le signal électrique est généralement proportionnelle à la grandeur physique mesurée.</i></p>	<p>Le signal est délivré à la commande sous forme analogique ou numérique.</p>
<p>Détecteur (du latin <i>detegere</i>, découvrir)</p>	<p>Dispositif de mesure fournissant un signal de sortie tout ou rien.</p>	<p>Le signal est délivré à la commande sous forme logique.</p>

II. Les constituants d'acquisition de données

1. Les constituants d'acquisition de consignes fournies par un opérateur

Durant la phase d'exploitation d'un système, il est indispensable que les personnels d'exploitation, de maintenance, de réglage, de surveillance puissent dialoguer avec la commande afin de donner des consignes :

- de réglages lors de la mise en route d'une nouvelle production ;
- de modes de marche et d'arrêt permettant la conduite du système ;
- de pilotage de production (changement de production, gestion qualitative et quantitative de la production, ...);
- pour permettre la maintenance, le diagnostic et le dépannage en cas d'incident.

Cette acquisition peut s'effectuer à partir de nombreux constituants reliés à la partie commande tels que :

- pupitre de commande et ses boutons, sélecteurs, roues codeuses, etc. ;
- constituants séparés spécifiques : commande bimanuelle, pendant, commande d'arrêt d'urgence, pédale, etc. ;
- terminal d'exploitation muni de touches et d'un clavier, ou muni d'un écran tactile ;
- poste de supervision relié à un réseau de communication.

a) Constituants d'acquisition de consignes tout ou rien

Ces constituants sont toujours très employés, car simples d'utilisation, faciles à mettre en œuvre et d'un coût inférieur aux constituants d'acquisition numériques qui pourraient remplir les mêmes fonctions. On distingue essentiellement les constituants pour :

- Commandes manuelles : bouton poussoir, sélecteur à deux positions ou plus, commande bimanuelle, bouton coup-de-poing, levier, manette, boîtes à boutons, etc. Ces constituants sont montés sur un pupitre de commande qui peut être fixe, mobile (pendant) ou séparé (radiocommande).

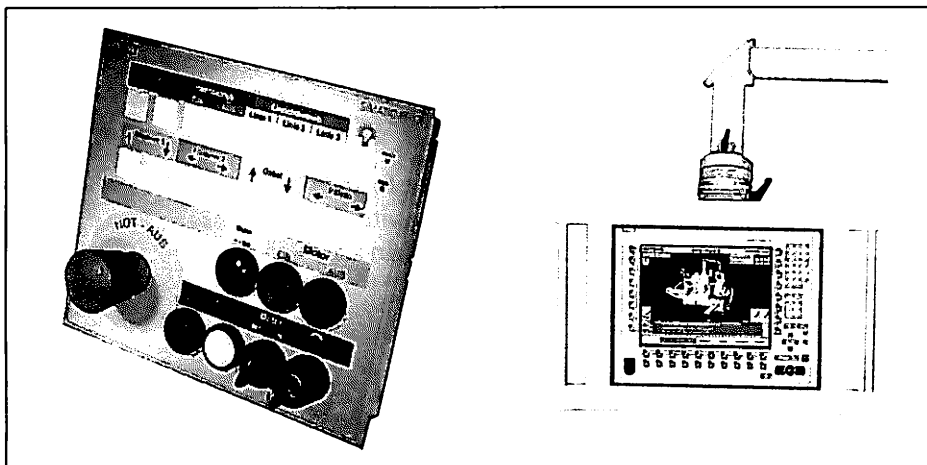


Figure 3.5. Exemples de pupitres : classique et « pendant » (doc. Siemens).

- Commandes au pied : interrupteur à pédale simple ou double, tapis sensible.

Certaines gammes de constituants d'acquisition de consignes tout ou rien peuvent être reliées à un bus de terrain.

b) Constituants d'acquisition de consignes numériques

L'acquisition de consignes numériques s'effectue généralement au moyen d'un clavier numérique ou alphanumérique. Les constructeurs proposent des pupitres programmables qui sont de plus munis de touches sensibles réalisant les mêmes fonctions que des boutons-poussoirs : dans ce cas, seules les fonctions de sécurité et de mise en énergie sont réalisées par des constituants séparés. Un terminal d'exploitation est relié à la partie commande par le biais d'une liaison série ou d'un bus de terrain.

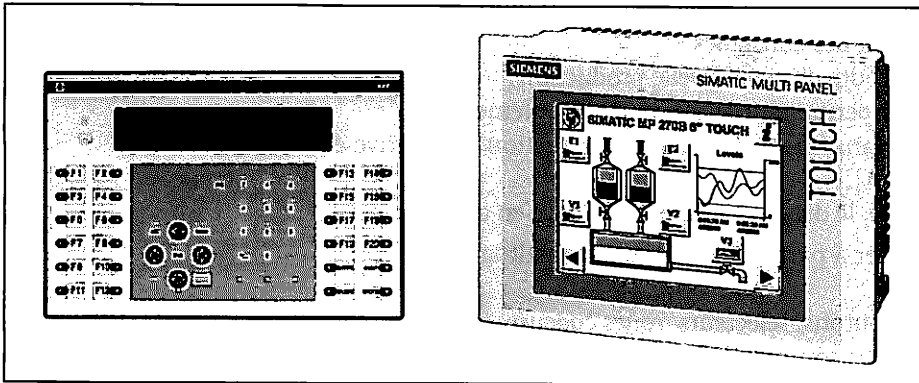


Figure 3.6. Exemples de terminaux d'exploitation alphanumériques et tactiles (doc. SCHNEIDER et Siemens).

L'utilisation de ce type de terminal programmable est économiquement justifiée dès qu'il y a plusieurs types de consignes (tout ou rien, numérique) et que le nombre de consignes devient important.

Toutefois, cette solution présente des inconvénients connus : nécessité d'employer un personnel d'exploitation suffisamment qualifié pour utiliser ces terminaux aux nombreuses fonctions et à l'ergonomie adaptée au procédé et au processus concerné. Pour gérer certaines situations critiques lors de l'exploitation de systèmes potentiellement dangereux, des constituants plus classiques sont encore utilisés : boutons, manettes, leviers et autres commandes rustiques mais efficaces en cas d'urgence.

● Cas particulier des roues codeuses

Lorsque l'acquisition de consignes numériques se limite à celle d'une valeur de consigne (nombre de produits à réaliser, température à atteindre, etc.) on peut utiliser une ou plusieurs roues codeuses reliées aux entrées tout ou rien de la partie commande et fournissant un code BCD qui devra être décodé.

c) Constituants d'acquisition de consignes analogiques

Il s'agit principalement de potentiomètres qui peuvent de présenter sous une forme circulaire, linéaire ou « joystick ». Ils permettent par exemple de régler une vitesse (cas des machines à commande numérique) ou de piloter le déplacement d'un axe numérique ou d'un robot.

2. Les détecteurs industriels

a) Structure générale

Deux grandes classes de détecteurs existent :

- les **interrupteurs**, ne possédant que deux états par construction : interrupteurs électromagnétiques ou pneumatiques, micro-rupteurs, détecteurs Reed, etc. Ils possèdent une structure fonctionnelle très rudimentaire, sans circuit de mise en forme puisqu'ils se limitent à commuter un signal déjà présent à leurs bornes.

– les détecteurs possédant une structure identique à celle d'un capteur, muni d'un système de détection de seuil, réglable ou non, de telle manière que le signal de sortie soit de type tout ou rien : détecteurs de proximité inductifs, capacitifs, photoélectriques, ultrasoniques, pressostats, vacuostats, etc.

Pour ce qui concerne la détection de présence ou non d'un objet, on distingue les détecteurs de proximité qui opèrent à distance et les détecteurs de présence qui opèrent par contact direct avec l'objet.

EXEMPLE

1. Dans le contact à pression utilisé sur le palettiseur, la pression d'air comprimé produit un déplacement de la membrane (corps d'épreuve). Ce déplacement assure la commutation d'un signal électrique par un contact électrique, permettant une modulation TOR.
2. Le capteur d'ensoleillement du store est constitué d'un phototransistor dont le signal de sortie analogique, proportionnel au flux lumineux reçu, est traité de façon à obtenir un « créneau » dès que le seuil est atteint (Figure 3.7).

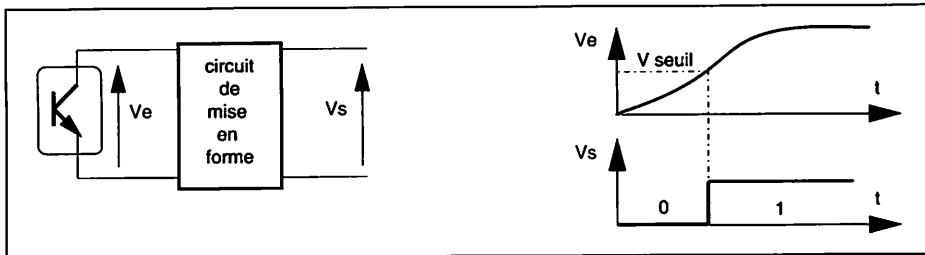


Figure 3.7. Mise en forme du signal du capteur Somfy.

Les détecteurs industriels les plus usités sont les détecteurs de présence. Ces détecteurs industriels disposent soit de sorties électriques, soit de sorties pneumatiques.

b) Caractéristiques principales des détecteurs de présence et de proximité

● La nature des sorties tout ou rien

Quelle que soit la technologie électrique employée (contacts secs ou transistors), il existe 2 types de circuits de sorties, dont les noms varient suivant les constructeurs :

- « normalement ouvert » (NO), encore appelé « à fermeture de circuit » (F) ou « travail » (T) ou « à établissement de circuit »;
- « normalement fermé » (NF), encore appelé « à ouverture de circuit » (O) ou « repos » (R) ou « à coupure de circuit ».

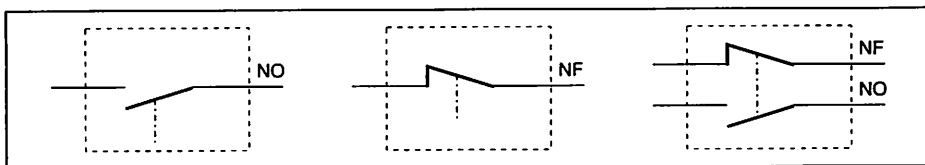


Figure 3.8. Contacts NO, NF et NO/NF.

● La fréquence maximale de commutation, exprimée en Hertz

C'est le nombre maximum de commutations possibles par seconde : de quelques hertz à plusieurs kHz selon le détecteur.

● La portée nominale S_n , exprimée en mm

Uniquement définie pour un détecteur de proximité, elle correspond à la distance à laquelle l'élément peut être détecté. La portée réelle est généralement inférieure à la portée nominale (30% à 100% de S_n) : elle dépend de nombreux paramètres, en particulier du matériau de la cible.

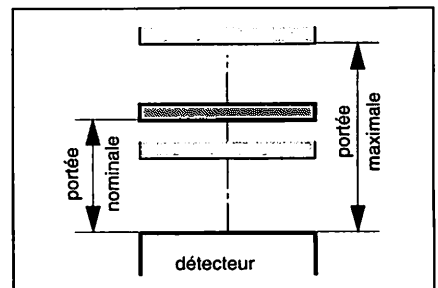


Figure 3.9. Portées d'un détecteur de proximité.

● La course différentielle ou l'hystérésis H, exprimée en mm

C'est la différence de course entre les points d'enclenchement (apparition d'un signal de sortie) et de déclenchement (disparition d'un signal de sortie) du détecteur. Dans le cas d'un détecteur de proximité elle est mesurée perpendiculairement à la face sensible et dans le cas d'un détecteur de présence selon l'axe de déplacement de l'organe de commande. En approche latérale la course différentielle est matérialisée par la zone située entre les limites d'enclenchement et de déclenchement (Figure 3.10).

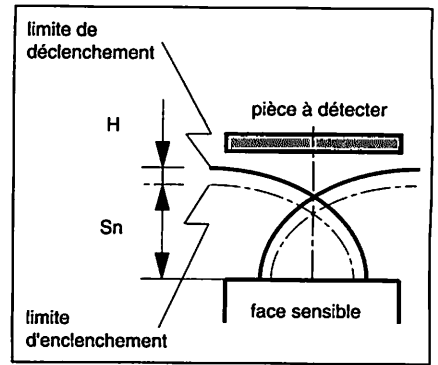


Figure 3.10. Zones d'enclenchement d'un détecteur inductif.

● La répétabilité, exprimée en mm

Elle correspond à la plage de tolérance de la position d'enclenchement lorsque l'élément à détecter est soumis à des détections répétées. Elle caractérise la capacité du détecteur à commuter en une même position.

c) Détecteurs de présence à action mécanique : détection par contact

Encore appelés interrupteurs de fin de course, interrupteurs de position, détecteurs de position ou, improprement, capteurs de fin de course. Ce sont des commutateurs commandés par le déplacement d'un organe de commande (corps d'épreuve). Lorsqu'ils sont actionnés, ils ouvrent ou ferment un ou plusieurs circuits électriques ou pneumatiques. Ce sont des détecteurs Tout Ou Rien (TOR).

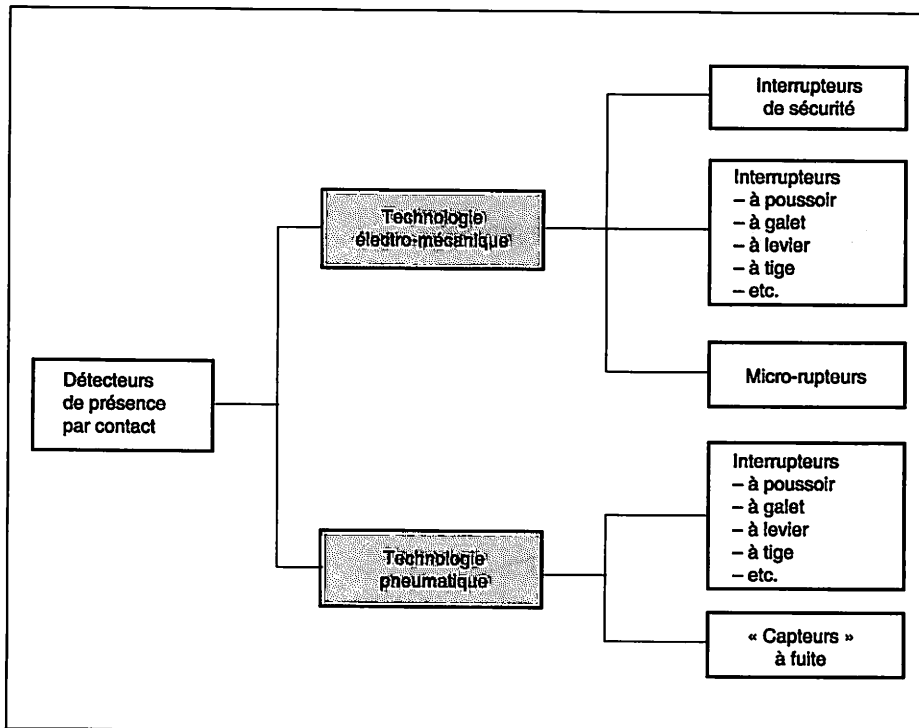


Figure 3.11. Typologie des détecteurs de présence par contact.

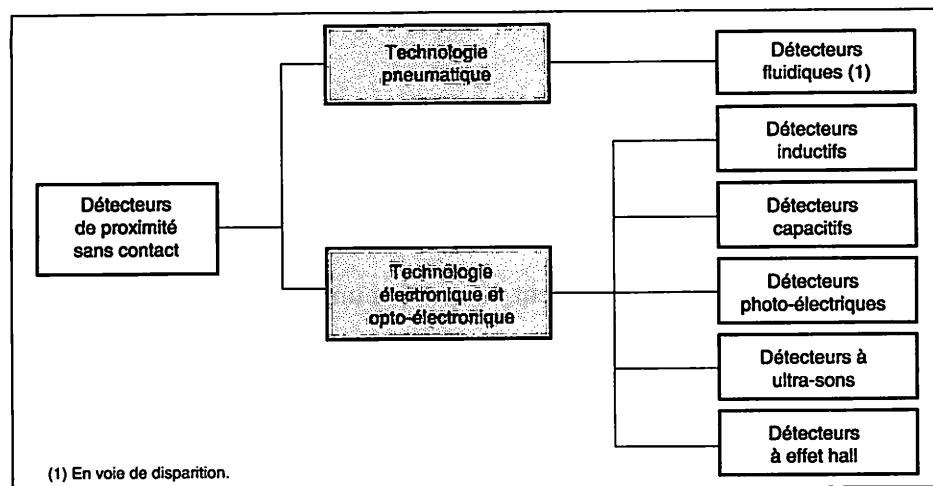
Les détecteurs pneumatiques sont peu à peu remplacés par des détecteurs à sortie électrique pour plusieurs raisons :

- La mise sur le marché de petits automates programmables à quelques entrées sorties et à faible coût. Dans ce cas de figure, les détecteurs pneumatiques doivent être interfacés, ce qui augmente le coût de l'application ;
- Le développement de détecteurs électroniques capables de travailler dans des ambiances explosives.

Pour plus de détails, voir FICHE TECHNIQUE 2.

d) Détecteurs de proximité sans contact

Si l'on excepte la technologie fluïdique, de moins en moins employée, les détecteurs de proximités sans contact sont conçus comme des capteurs utilisant un système à détection de seuil (permettant ainsi de générer un signal de sortie de type TOR). Compte tenu de leur coût, de leur durée de vie et de l'absence de contact direct avec les éléments à détecter, ces détecteurs de proximités sont de plus en plus utilisés.



REMARQUE

Les détecteurs à effet Hall possèdent une partie libre qui est fixée sur l'objet à détecter. Ils sont en contact avec cet objet, mais la détection s'effectue à distance.

Figure 3.12. Typologie des détecteurs de proximité sans contact.

Pour plus de détails, voir FICHE TECHNIQUE 2.

e) Détecteurs de position implantables sur vérins

Ces détecteurs existent dans toutes les technologies (Figure 3.13). Ils sont fixés directement sur le corps d'un vérin ce qui simplifie l'installation des moyens d'acquisition des informations de position des mobiles mus par vérin.

Il faut toutefois noter que l'information délivrée concerne la position de la tige du vérin et non la position de l'objet déplacé ou bridé. Cela peut parfois être la cause de problèmes liés à la sécurité et à la disponibilité des installations.

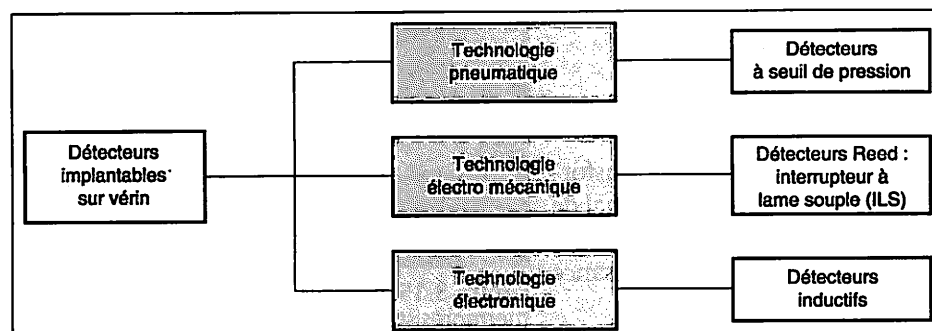


Figure 3.13. Typologie des détecteurs implantables sur vérin.

Pour plus de détails, voir FICHE TECHNIQUE 2.

f) Détecteurs de pression

La plupart des systèmes utilisant l'énergie pneumatique (ou hydraulique) sont conçus pour fonctionner avec une pression minimale dans le circuit de puissance. Ce seuil de pression peut être détecté par un détecteur appelé : pressostat.

De la même manière, certains systèmes automatiques utilisent le vide pour assurer certaines fonctions opératives, il peut ainsi être utile de mesurer une dépression (génération de vide, ventouses, ...) avec un détecteur appelé : vaccuostat.

● Les pressostats

Ils ont pour fonction de détecter un seuil de pression dans un circuit hydraulique, pneumatique, et plus généralement de tout corps liquide ou gazeux, le seuil étant généralement réglable. Il existe également des pressostats à double seuil, permettant la régulation tout ou rien de la pression du fluide. Le signal de sortie de ces détecteurs est du type TOR sur contacts secs à ouverture ou à fermeture. La plage de fonctionnement évolue typiquement entre 1 bar et plusieurs centaines de bars en hydraulique.

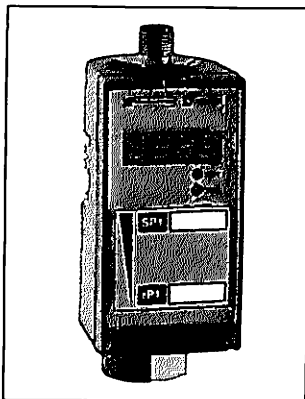


Figure 3.14. Pressostat électronique à double seuil (document SCHNEIDER).

EXEMPLE

Pressostat de contrôle de la pression d'air dans un circuit pneumatique 6 bars. Si la pression descend en dessous d'un seuil pré-réglé (aux alentours de 4 bars), le pressostat déclenche une procédure de sécurité.

● Les vacuostats

De la même manière que les pressostats, les vacuostats permettent de contrôler un seuil de dépression.

EXEMPLE

Vacuostat de compte rendu « dépression obtenue » dans un circuit pneumatique alimentant une ou des ventouses. Cette information permet de valider le fait que l'objet est correctement saisi par la ventouse.

3. Les capteurs industriels

a) Mode de fonctionnement des capteurs

Les capteurs industriels peuvent se présenter sous des formes très diverses, tant ils sont variés, nombreux et en constante évolution. Toutefois, certains points communs peuvent apparaître au niveau des fonctions internes réalisées par des constituants typiques (Figure 3.3) ainsi qu'au niveau du mode de fonctionnement du capteur.

Les capteurs fonctionnent selon deux principes de base suivant l'origine du signal électrique de sortie :

- les capteurs actifs (fonctionnant en *générateurs*);
- les capteurs passifs (fonctionnant en *modulateurs*).

● Les capteurs actifs

Une partie de l'énergie prélevée sur le mesurande est transformée directement en énergie électrique constituant le signal de sortie. Ce signal peut être : un courant, une tension, ou une quantité d'électricité (dans le cas d'un signal de faible puissance une amplification peut être nécessaire).

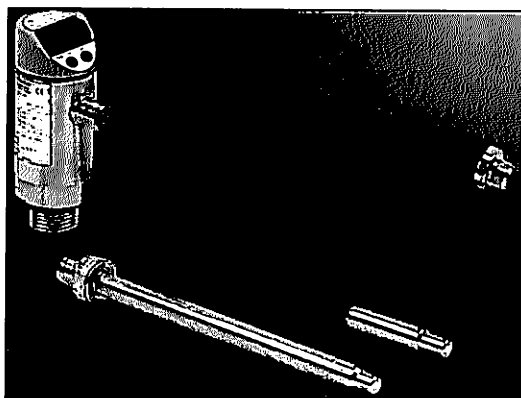
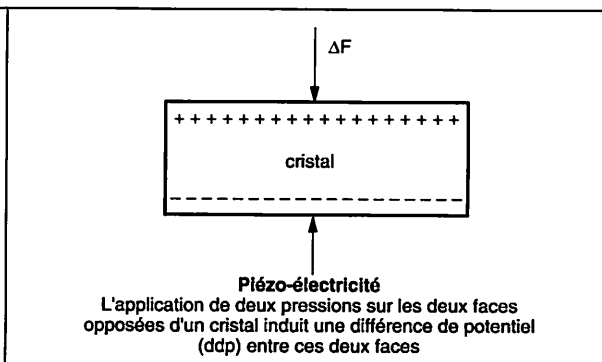
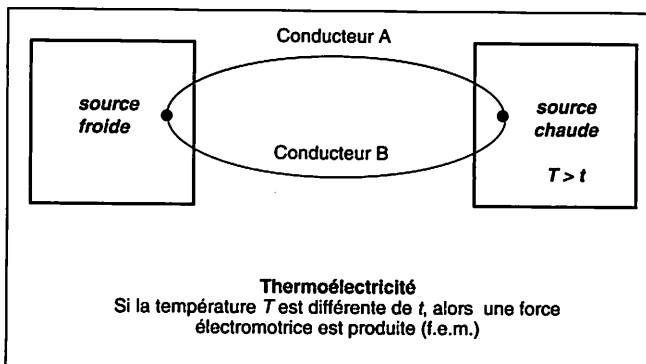


Figure 3.15. Capteur (actif) de température à thermorésistance platine (doc IFM electronic).



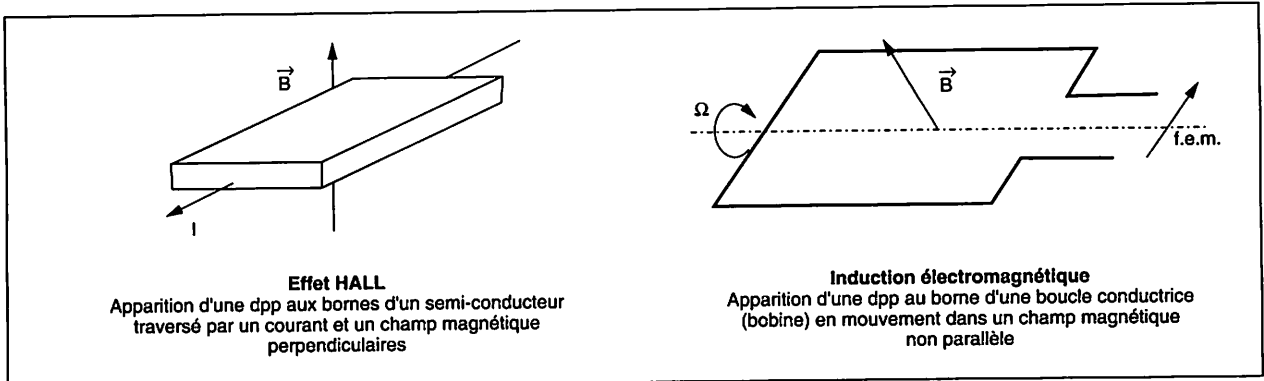


Figure 3.16. Capteurs actifs : principes physiques utilisés.

● **Les capteurs passifs**

C'est l'impédance du capteur qui est sensible aux variations du mesurande. Un module de préconditionnement électronique sera nécessaire pour mesurer les variations d'impédance.

Ces capteurs sont alimentés par une source d'énergie extérieure (tension continue ou alternative, courant).

Le signal de sortie est généralement une tension continue modulée en fréquence.

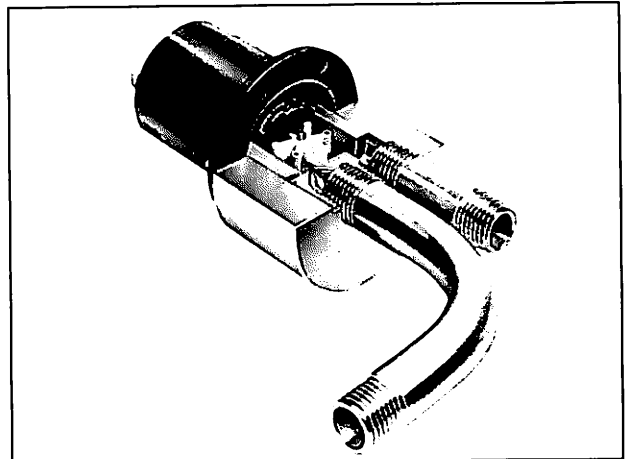


Figure 3.17. Débitmètre à turbine (doc KOBOLD).

EXEMPLES

Capteurs de pression à diaphragme, capteurs de force à jauges, accéléromètres sismiques, anémomètres à hélice, débitmètres à turbine, couplemètres, gyroscopes à gyroscope, etc.

Mesurande sensible	Caractéristique électrique utilisé	Types de matériaux
Température	Résistivité	Métaux : platine, nickel, cuivre, semi-conducteurs.
Très basse température	Constante diélectrique	Verres
Flux de rayonnement optique	Résistivité	Semi-conducteurs
Déformation	Résistivité Perméabilité magnétique	Alliage de nickel, silicium dopé Alliages ferromagnétiques
Position (aimant)	Résistivité	Matériaux magnéto-résistants: bismuth, antimoniure d'indium
Humidité	Résistivité Constante diélectrique	Chlorure de lithium Alumine, polymères
Niveau	Constante diélectrique	Liquides isolants

Figure 3.18. Capteurs passifs : principes physiques et matériaux.

b) Cas particulier des capteurs de vitesse et de position munis d'un élément mobile

Il s'agit de capteur dont le corps d'épreuve est un élément mobile, guidé en translation ou en rotation, lié mécaniquement au solide dont on désire mesurer la position ou la vitesse. Le corps d'épreuve est particulier, en ce sens qu'il ne réalise pas une conversion de la grandeur physique à mesurer mais un couplage avec le dispositif interne optique ou électromagnétique.

EXEMPLES

Codeurs optiques incrémentaux et absolus, règles optiques, tachy-codeurs, capteurs LVDT, capteurs inductifs à noyau plongeur, capteurs potentiométriques, synchromachines, resolver, règles inductosyn, génératrices tachymétriques, capteurs à effet hall, etc.

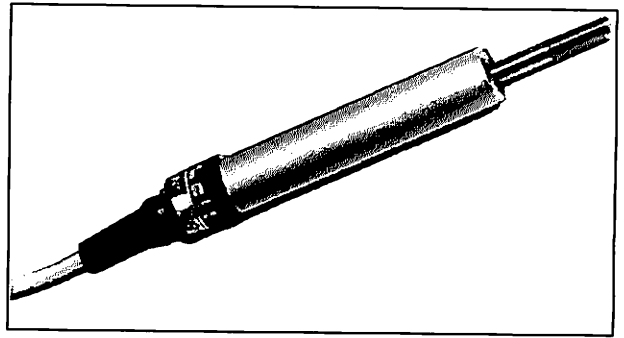


Figure 3.19. Capteur de déplacement rectiligne inductif LVDT avec sa tige mobile (doc Chauvin & Arnoux).

c) Typologie des principaux capteurs de grandeurs mécaniques et de température

Les figures 3.20 et 3.21 proposent une typologie prenant en compte la technologie, la structure et l'absence ou non de contact avec le corps physique à mesurer. Cette typologie est limitée aux capteurs de grandeurs mécaniques telles que position, vitesse, accélération, force, couple, pression, débit, ainsi qu'aux capteurs de température.

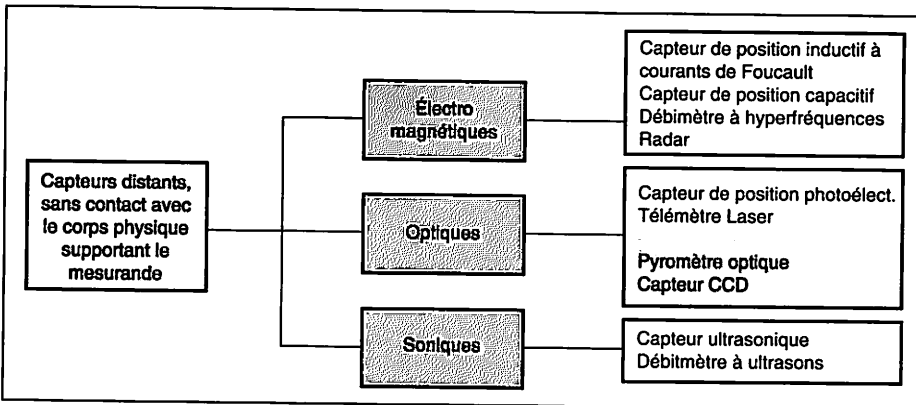


Figure 3.20. Typologie des capteurs distants, sans contact avec le corps physique supportant le mesurande (capteurs actifs en gris).

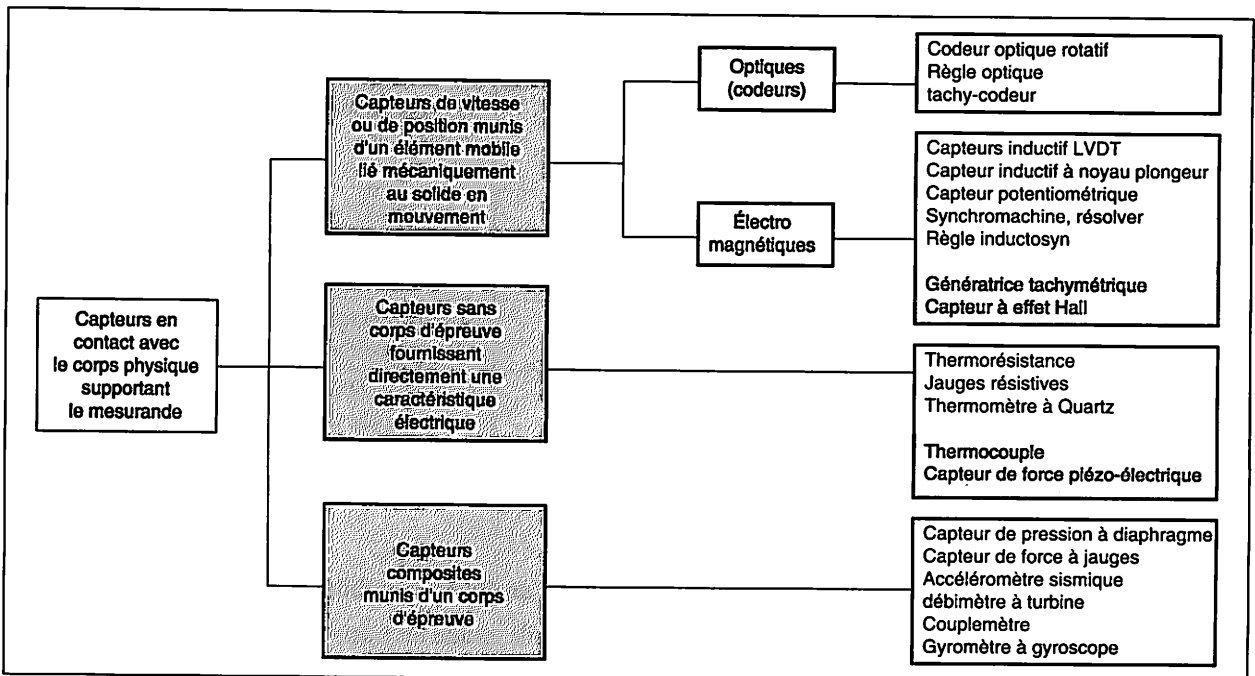


Figure 3.21. Typologie des capteurs en contact avec le corps physique supportant le mesurande (capteurs actifs en gris).

Pour plus de détails concernant les codeurs optiques, voir FICHE TECHNIQUE 1.

d) Principales caractéristiques métrologiques des capteurs (norme AFNOR NF X07001, Instruments de mesure, vocabulaire.)

● **Fidélité** : la fidélité est l'aptitude à donner pour une même valeur de la grandeur mesurée, des indications concordant entre elles. Les résultats du mesurage sont groupés autour de leur valeur moyenne.

● **Justesse** : la justesse est l'aptitude à donner des indications égales à la grandeur mesurée, les erreurs de fidélité n'étant pas prises en considération.

● **Précision** : c'est la qualité qui caractérise l'aptitude d'un capteur à donner des indications qui, individuellement, sont proches de la valeur vraie de la grandeur mesurée : un capteur précis est à la fois juste et fidèle. L'erreur de précision est généralement exprimée en pourcentage de l'étendue de mesure ou de la pleine échelle % P.E.

● **Sensibilité** : pour une valeur donnée de la grandeur à mesurer E_0 , la sensibilité s'exprime par le quotient de la variation de la grandeur de sortie S par la variation correspondante de la grandeur mesurée autour de la valeur E : $\left(\frac{\Delta S}{\Delta E}\right)_{E=E_0}$

● **Étendue de mesure** : c'est la différence algébrique entre les valeurs extrêmes pouvant être prises par la grandeur à mesurer. Les performances métrologiques du capteur ne sont garanties qu'à l'intérieur de l'étendue de mesure. On appelle portées les valeurs limites de la grandeur à mesurer correspondant à cette étendue de mesure, la portée minimale n'étant pas toujours nulle.

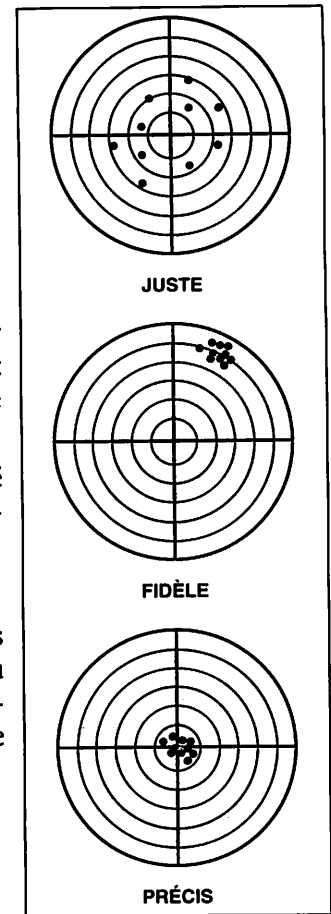


Figure 3.22. Illustration graphique des caractéristiques de fidélité, justesse et précision.

Résumé

● **La chaîne d'acquisition** fournit à la chaîne de traitement des informations provenant de l'extérieur du système, et qui peuvent être :

- des données physiques, dont l'acquisition s'effectue au moyen de **capteurs** qui sont des constituants fournissant un signal analogique ou numérique, image d'une grandeur physique et de **détecteurs** qui fournissent un signal tout ou rien ;
- des consignes fournies par un opérateur, dont l'acquisition s'effectue au moyen de constituants tels que boutons poussoirs, sélecteurs, claviers, pédales, etc ;
- des informations provenant d'un autre système.

● **Les capteurs industriels** sont généralement intégrés, c'est-à-dire qu'ils réalisent les fonctions de conversion de la grandeur physique à mesurer et d'adaptation du signal en sortie. Certains capteurs possèdent un élément mécanique appelé corps d'épreuve qui convertit la grandeur à mesurer en une autre grandeur mesurable. Ce corps d'épreuve peut également se présenter sous la forme d'une partie mobile en translation ou en rotation dans le cas particulier de capteurs de position et de vitesse. Les principales caractéristiques métrologiques des capteurs sont la fidélité, la justesse, la précision, la sensibilité et l'étendue de mesure.

● **Les détecteurs industriels** se présentent sous la forme d'interrupteurs mécaniques ou de capteurs tout ou rien. Ils agissent par contact (détecteurs de présence) ou à distance (détecteurs de proximité). Leurs principales caractéristiques sont le type des sorties (NO ou NF), la fréquence de commutation, la portée nominale, l'hystérésis et la répétabilité.

Exercices

EXERCICE 1. Combien de roues codeuses faut-il utiliser pour acquérir des valeurs numériques comprises entre 1 et 10, le codage étant effectué en BCD ?

EXERCICE 2. En utilisant la **FICHE MÉTHODE 6** et la **FICHE TECHNIQUE 2**, proposer un détecteur adapté dans le cas suivant : détection du passage de pièces en acier fraîchement peintes sur une chaîne de production. Distance de détection ≤ 10 mm.

EXERCICE 3. En utilisant la **FICHE MÉTHODE 6** et la **FICHE TECHNIQUE 2**, proposer un détecteur adapté dans le cas suivant : contrôle du remplissage de bidons opaques sur une chaîne de production. Distance de détection ≤ 10 mm.

EXERCICE 4. En utilisant la **FICHE MÉTHODE 6** et la **FICHE TECHNIQUE 2**, proposer un détecteur adapté dans le cas suivant : contrôle de passage de bouteilles en verre. Distance de détection = 50 mm.

EXERCICE 5. En utilisant la **FICHE MÉTHODE 6** et la **FICHE TECHNIQUE 2**, proposer un détecteur adapté dans le cas suivant : détection de la présence d'un obstacle quelconque devant un chariot mobile. Portée = 5 m.

EXERCICE 6. En utilisant la **FICHE MÉTHODE 6** et la **FICHE TECHNIQUE 2**, proposer un détecteur adapté dans le cas suivant : détection du passage de petites pièces en céramique en sortie de bol vibrant. Diamètre des pièces = 3 mm.

EXERCICE 7. En utilisant la **FICHE MÉTHODE 6** et la **FICHE TECHNIQUE 2**, proposer un détecteur adapté dans le cas suivant : tri de pièces en fonction de leur couleur rouge ou bleue.

EXERCICE 8. En utilisant la **FICHE TECHNIQUE 1**, déterminer le nombre minimal de points d'un codeur incrémental dans le cas suivant : chariot entraîné par une vis à billes de pas 10 mm, codeur monté sur la vis, résolution de la mesure désirée = 0.01 mm.

EXERCICE 9. Déterminer la fréquence maximale des signaux numériques fournis par le codeur incrémental de l'exercice 8. Commenter. Vitesse de rotation de la vis = 1000 tours par minute.

EXERCICE 10. Déterminer la taille minimale en bits du compteur rapide contenant l'information numérique binaire de la position du chariot. Course du chariot = 600 mm.

4

Descriptions des systèmes automatiques

I. Le cycle de vie d'un système automatique

Un projet de conception, réalisation et mise en exploitation d'un système automatique est toujours marqué par un certain nombre d'étapes caractéristiques de l'avancement de son développement (voir aussi chapitre 1 - III). Ces étapes peuvent être représentées sur un cycle en «V» (Figure 4.1) utilisé à l'origine en génie logiciel. À chaque étape du développement, une activité principale peut être identifiée (spécification, conception, réalisation, test, intégration, ...). Chaque activité est contrainte par des données d'entrée (cahier des charges fourni par l'activité située en amont) et des données de sortie ou résultats d'activité (cahier des charges ou spécification pour l'activité suivante).

Une bonne gestion de projet nécessite une identification rigoureuse de ces activités et des résultats intermédiaires attendus.

L'axe horizontal (de la gauche vers la droite) met en évidence la correspondance entre les activités de conception et les activités de tests ou d'intégration. Afin d'assurer la bonne réalisation des activités de tests et d'intégration, des dossiers de tests et des cahiers de recette sont produits au cours des activités de conception. De nombreux tests intermédiaires sont en effet indispensables avant l'intégration complète du système et sa mise en exploitation.

L'axe vertical (du haut vers le bas) traduit le degré d'abstraction : du plus général au plus particulier (pour la branche gauche du «V»), du plus fonctionnel au plus technologique, du plus intégré au plus élémentaire (pour la branche droite du «V»).

Selon les étapes de la vie du système, des modèles et des représentations différents peuvent être utilisés pour répondre à des besoins précis et variés (modèles fonctionnels, modèles comportementaux, équations, dessins mécaniques, schémas, algorithmes, programmes, ...).

Bien que l'utilisateur, qui doit comprendre ou modifier les modèles initiaux à des fins de conduite et/ou de maintenance, n'ait pas les mêmes objectifs que le concepteur (créateur des modèles), il ne peut, en règle générale, découvrir et appréhender le fonctionnement du système automatique qu'à partir des documents réalisés en phases de conception et de réalisation. Il est donc essentiel que les méthodes et outils utilisés permettent de véhiculer les informations élaborées tout au long du cycle de vie, de façon à faciliter la communication entre tous les intervenants.

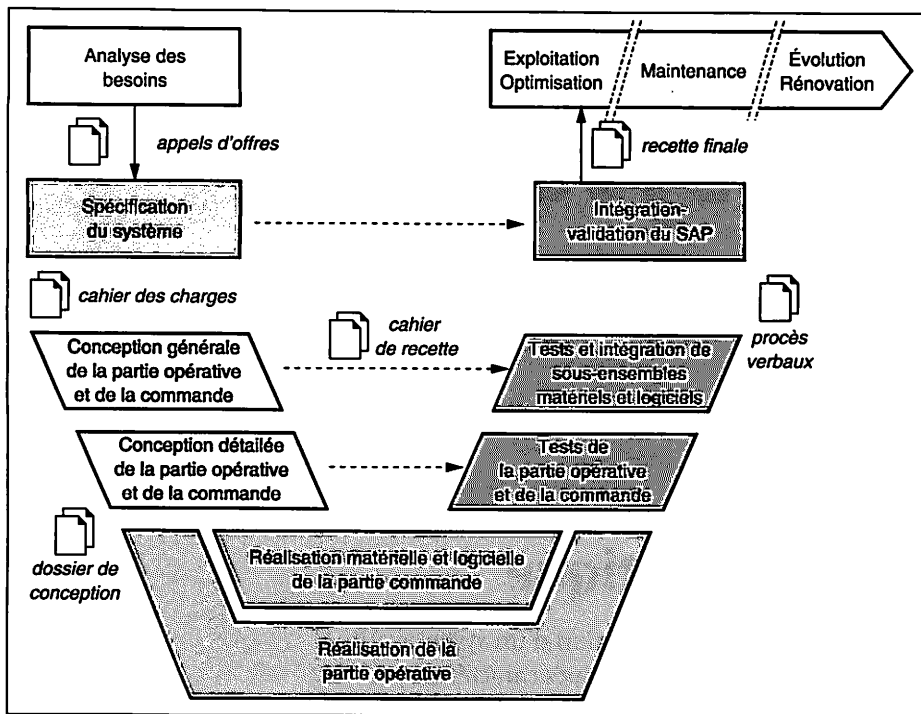


Figure 4.1. Les différentes étapes du cycle de vie d'un système automatique .

II. Description du comportement de la commande d'un système automatique

Le cahier des charges issu de la phase de spécification générale du système automatique, précise en particulier les fonctions qu'il doit assurer pour répondre aux besoins exprimés par le client. Une partie de ces fonctions seront réalisées par les exploitants, d'autres par le système automatique lui-même.

Pour concevoir la commande de ce système, l'automaticien va s'intéresser aux fonctions réalisées automatiquement (les fonctions assurées par le procédé). Après une première étape d'identification de ces fonctions, il est alors possible de commencer la description du comportement de la commande, puis des équipements associés. Au cours des différentes phases de conception, et donc de définitions progressives des équipements, les descriptions des fonctions et du comportement vont s'affiner.

1. Représentation des fonctions réalisées par le procédé

La représentation des fonctions réalisées par le procédé peut se faire à l'aide du diagramme FAST (Function Analysis System Technic). Il se présente sous forme d'un « arbre » de fonctions partant d'une fonction globale qui se décline de plus en plus finement en fonctions composantes, puis en fonctions élémentaires dont certaines sont des fonctions techniques (voir lexique).

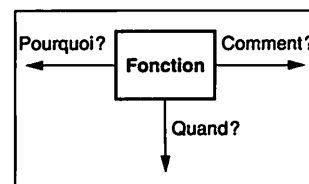
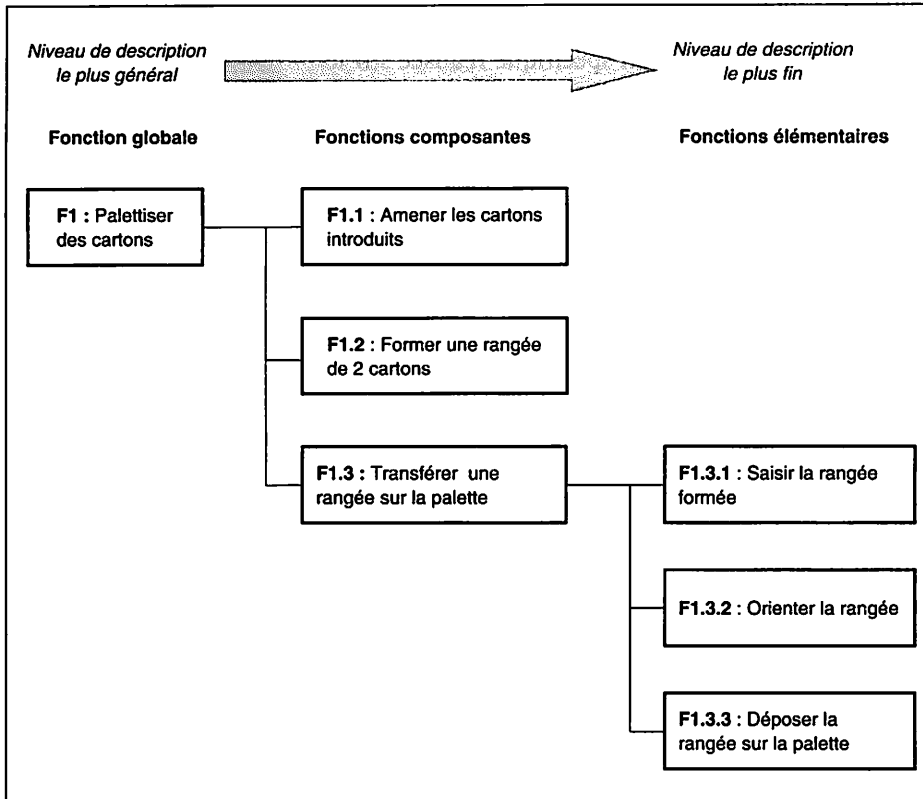


Figure 4.2. Principe de lecture d'un diagramme FAST.

Sa lecture (Figure 4.2) est fondée sur une technique interrogative : « Pourquoi cette fonction doit-elle être assurée ? Comment ? Quand ? ». La décomposition horizontale des fonctions (Figure 4.3) aboutit à représenter le passage du « pourquoi » (la fonction globale) au « comment » (les fonctions composantes), puis au « quoi » (les solutions techniques).

La décomposition verticale des fonctions précise le « quand » : les fonctions composantes peuvent être réalisées successivement ou simultanément.



REMARQUE

Dans le cas où l'objectif du concepteur est de représenter le comportement de la commande du système automatique, seules les fonctions réalisées automatiquement sont recensées dans le diagramme FAST. Les fonctions assurées par les exploitants (chargement, déchargement manuel par exemple) ne sont pas prises en compte.

Figure 4.3. Diagramme FAST partiel du palettiseur, fonctions réalisées par le procédé.

Il est indispensable que le développement d'un système automatique prenne en compte dès la conception tous les points de vue relatifs aux différentes spécialités des nombreux intervenants. Il en va de la performance globale du système et de sa conformité au cahier des charges.

Ces divers points de vue s'expriment simultanément ou successivement suivant le cas et à chaque niveau de description du système. La définition de ces niveaux de description et des frontières associées s'appuie principalement sur les critères suivants :

- un *critère de fonctionnement*, lié à l'expression des besoins concernant la conduite du système automatique (fonctions réalisées par le procédé) ;
- un *critère lié à la fourniture*, privilégiant un découpage par fournisseur ou par lot (système automatique, station ou cellule de production, poste de travail, chaîne d'action, capteurs/actionneurs).

a) Critère de fonctionnement

C'est une hiérarchie d'architecture fonctionnelle du système automatique. On distingue essentiellement les niveaux suivants, représentés figure 4.4.

● **Système automatique** : ensemble réalisant la totalité des opérations nécessaires pour apporter la valeur ajoutée au(x) produit(s).

● **Station** : sous-ensemble d'une machine permettant d'effectuer un ensemble d'opérations sur le produit maintenu en position. Les machines-transfert comportent en général plusieurs stations.

EXEMPLE

La machine de conditionnement comporte trois stations (trois arrêts de palette) : la station de chargement-déchargement des palettes, la station de remplissage et la station de bouchage.

Le palettiseur ne comporte qu'une station.

● **Poste** : partie d'une station associée à une fonction spécifique réalisée sur le produit (fonction : opérative, d'identification, de mesure, etc.).

EXEMPLE

La machine de conditionnement comporte trois postes (il y a pour cette machine identité entre postes et stations).

Le palettiseur comporte deux postes : le poste de groupage des cartons (fonctions « amener les cartons » et « former une rangée ») et le poste de palettisation (fonction « transférer une rangée sur la palette »).

● **Chaîne d'action** : ensemble de constituants d'un poste organisés pour réaliser une fonction élémentaire et, notamment, une fonction opérative (une fonction opérative est une fonction associée à une opération directe sur les matières d'œuvre; par opposition à d'autres fonctions, nécessaires pour le bon fonctionnement du système mais n'opérant pas directement sur le produit, par exemple : fonctions d'acquisition de données, de traitement, de gestion de l'énergie, de dialogue ou de communication, etc.).

Plusieurs chaînes d'action peuvent être associées : on parle alors de sous-ensemble fonctionnel.

Un sous-ensemble fonctionnel est un ensemble de chaînes d'actions appartenant au même poste et organisées pour réaliser une fonction particulière sur le produit (voir l'exemple ci-après).

EXEMPLE

Le poste de bouchage matérialisant la fonction « boucher un flacon » de la machine de conditionnement se compose de deux sous-ensembles fonctionnels : « indexage palette » et « saisie bouchon », et de la chaîne d'action « transfert ». Chaque sous-ensemble fonctionnel est lui-même constitué de deux chaînes d'action :

- « indexage palette » est formé des deux chaînes d'action : l'« arrêt palette » (vérin associé et son distributeur) et le « blocage palette » (vérin associé, etc.);
- « saisie bouchon » est formé des deux chaînes d'action « saisie par ventouse » et « montée ventouse »;
- « déplacement horizontal ventouse » : vérin associé et son distributeur;

Le poste de bouchage comporte donc cinq chaînes d'action.

Le poste de groupage des cartons du palettiseur comporte deux chaînes d'action : « faire avancer cartons » et « pousser cartons ».

b) Critère lié à la fourniture

Ce critère permet une classification très générale qui privilégie la constitution matérielle du système automatique.

On distingue en général trois niveaux : le composant, le constituant et l'équipement.

● **Composant** : c'est l'entité technologique élémentaire d'un constituant (exemples en figure 4.5). Sa mise en œuvre concerne essentiellement les constructeurs de constituants. Du point de vue des personnels de maintenance, ce sont des éléments non réparables (mais le plus souvent interchangeables).

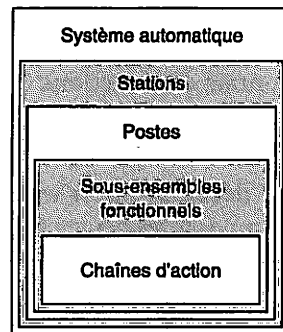


Figure 4.4. Structure fonctionnelle hiérarchisée d'un système automatique.

● **Constituant** (d'automatisme) : il désigne un ensemble de composants agencés par un fournisseur pour réaliser une fonction élémentaire. Le choix d'un constituant, sa mise en œuvre (installation, réglage et/ou paramétrage) et sa maintenance (interchangeabilité des composants réparables) sont du domaine des concepteurs, des constructeurs et des installateurs de machines (exemples : voir figure 4.5).

● **Équipement** (d'automatisme) : c'est l'ensemble de constituants et de composants agencés pour réaliser une fourniture complète ou partielle d'une machine. La hiérarchie ainsi introduite est une hiérarchie de fourniture (par des fournisseurs industriels ou des sous-traitants) dans la réalisation du système automatique.

Équipement	Constituant	Composant
Convoyeur	Modules de convoyage Motoréducteur Palettes	Bande de convoyage, éléments modulaires, palier, renvoi, ... Bobinage, tôle magnétique, roulement, arbre, engrenages, ...
Poste de bouchage	Préhenseur Unité de guidage linéaire Indexeur	Ventouse, venturi Rail de guidage, chariot, vérin, détecteurs Vérin, plaque-support, vis

Figures 4.5. Exemples de fournitures concernant la machine de conditionnement.

c) Quel critère pour décrire le comportement de la commande ?

Le critère de fonctionnement est le critère le plus pertinent pour la marche du système automatique, donc pour la description de son comportement. En effet, les besoins initiaux s'expriment souvent en terme d'exploitation (conduite, maintenance, réglage, sécurité, ...) et de performance de l'installation pour obtenir la valeur ajoutée souhaitée sur le flux de produits. Ce découpage fonctionnel est souvent privilégié, car il correspond à une réponse aux besoins de l'utilisateur final qu'est l'exploitant.

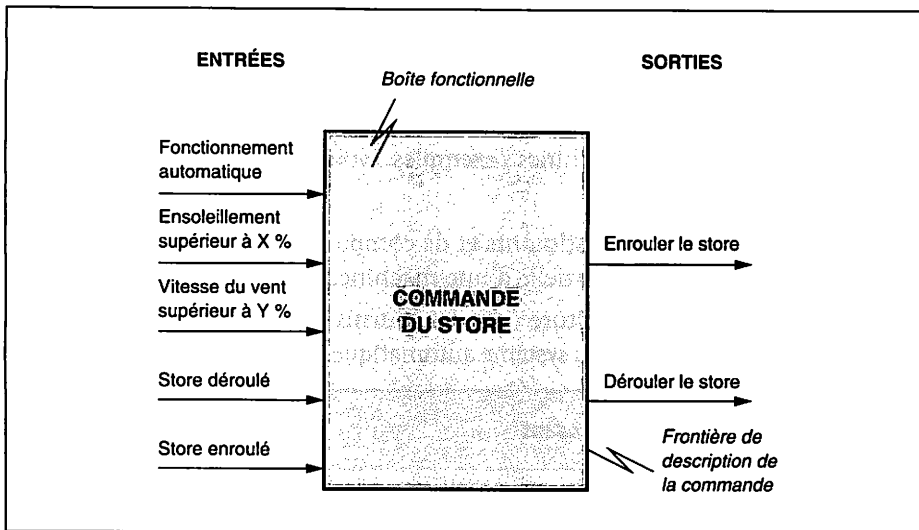
Tous les autres aspects sont décrits en lien avec cette décomposition, c'est ce qui permet de garantir une cohérence entre les diverses descriptions et de faciliter la communication, en évitant notamment les interprétations différentes. Si au cours des différentes phases du projet d'autres décompositions sont nécessaires (géographique, par fournisseur, par métier, etc.), elles doivent toutes se référer à la décomposition de référence privilégiant le critère du fonctionnement.

2. Représentation du comportement de la commande

a) Frontière de description du comportement de la commande

L'identification des propriétés de la commande, sa conception, ou la modification de sa structure et du programme nécessaire au fonctionnement de l'application, nécessitent l'utilisation de modèles de description permettant la spécification de son comportement.

L'utilisation d'un formalisme de type « boîte fonctionnelle » (Figure 4.6) permet, par le tracé d'une frontière, de préciser les interactions de la commande avec l'environnement et de séparer clairement les entrées des sorties. Cette description est purement fonctionnelle, elle est un préalable à toute modélisation comportementale.



REMARQUE

La modélisation du comportement passe obligatoirement par une phase d'isolement nécessaire pour identifier les frontières de ce que l'on s'attache à décrire.

Figure 4.6. Frontière de description du comportement de la commande du store.

b) Caractéristiques du modèle comportemental

● **L'abstraction**

On ne manipule pas les grandeurs physiques, mais des informations (éléments de connaissance liés à celles-ci).

EXEMPLE

Dans le cas de la figure 4.6 « Store enroulé » est une information d'entrée.

● **L'idéalisation**

Le signal (paragraphe 3, chapitre 2) délivré par le capteur du détecteur 1 sur tambour (tension électrique) est idéalisé par une variable booléenne, on ne s'intéresse pas aux phénomènes transitoires (observables avec un oscilloscope par exemple).

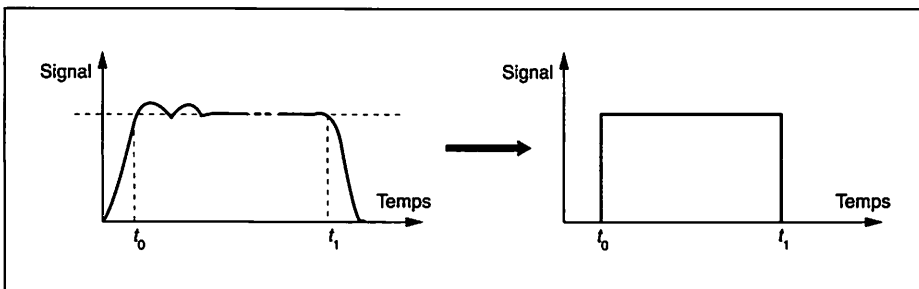


Figure 4.7. Illustration de l'idéalisation.

Cette « idéalisation » est souvent suffisante pour décrire le comportement tout en répondant au cahier des charges.

● **La simplification**

Seul un aspect de la commande est modélisé, sous un point de vue et avec une finesse de description (grossissement) choisis par le modélisateur et supposés pertinents vis-à-vis des objectifs poursuivis.

EXEMPLE

Les modes de marche, la gestion des sécurités des exploitants, le traitement du dialogue homme-machine, la surveillance des alimentations, la partie matérielle relative à la mise en énergie, ...

c) Les niveaux de décision

Il est possible d'élaborer une typologie des entrées selon leur importance (portée) décisionnelle : les entrées « consignes opérateur », telles que l'arrêt d'urgence, la mise en marche, l'arrêt, etc., n'ont pas la même importance dans la hiérarchie décisionnelle de la commande qu'une information issue d'un capteur fin de course permettant de passer à l'action suivante dans le cycle de fonctionnement.

La description du comportement d'une commande est très souvent structurée hiérarchiquement en fonction de cette hiérarchie décisionnelle. Les différentes *décisions* peuvent en effet être classifiées en plusieurs niveaux selon la portée (influence sur les niveaux inférieurs) ou l'importance de l'action déclenchée (par exemple, commande d'une action, commande d'un enchaînement d'action, commande de l'inhibition de l'ensemble des actions, etc.).

Chaque niveau de décision fait appel à un ensemble d'informations mémorisées, élaborées par d'autres niveaux de décision ou issues de la chaîne d'énergie (système opérant). Cette mise à disposition peut nécessiter un transfert des informations localement ou à longue distance.

Dans les industries manufacturières, la hiérarchie décisionnelle suivante est souvent prise en compte :

- la *surveillance* pour assurer la sécurité des exploitants et la non détérioration du système automatique;
- la *conduite* pour gérer le dialogue homme machine et pouvoir exploiter le système automatique;
- la *coordination* pour coordonner les fonctionnements des différentes zones de productions (voir la notion de coordination des tâches dans le chapitre 7);
- et la *commande* pour générer les ordres destinés aux actionneurs.

L'illustration (Figure 4.8) s'appuie sur l'exemple du palettiseur (voir FICHE SYSTÈME I) et l'application se fait en conception générale au niveau de la commande du procédé. Chaque commande de procédé peut être vue aussi comme le moyen de contrôler le processus concerné (paragraphe III.1).

En entrée/sortie de chaque boîte fonctionnelle, les flèches en pointillés représentent les échanges entre les différents niveaux de décision, les autres flèches représentent les échanges d'informations avec le procédé (comptes-rendus : tel que couche saisie, ordres : tel que saisir la rangée formée) et avec les exploitants (manu, auto, signaler palette terminée, ...).

Niveau de décision (du plus élevé au plus bas)	Fonctions assurées à ce niveau décisionnel	Boîtes fonctionnelles de caractérisation des entrées-sorties des commandes associées à chaque niveau de décision
Surveillance	Surveillance de temps de cycle, surveillance du bon fonctionnement, gestion des sécurités, ...	
Conduite	Gestion des modes de marche et d'arrêt du procédé de palettisation de cartons pour la conduite du système automatique	

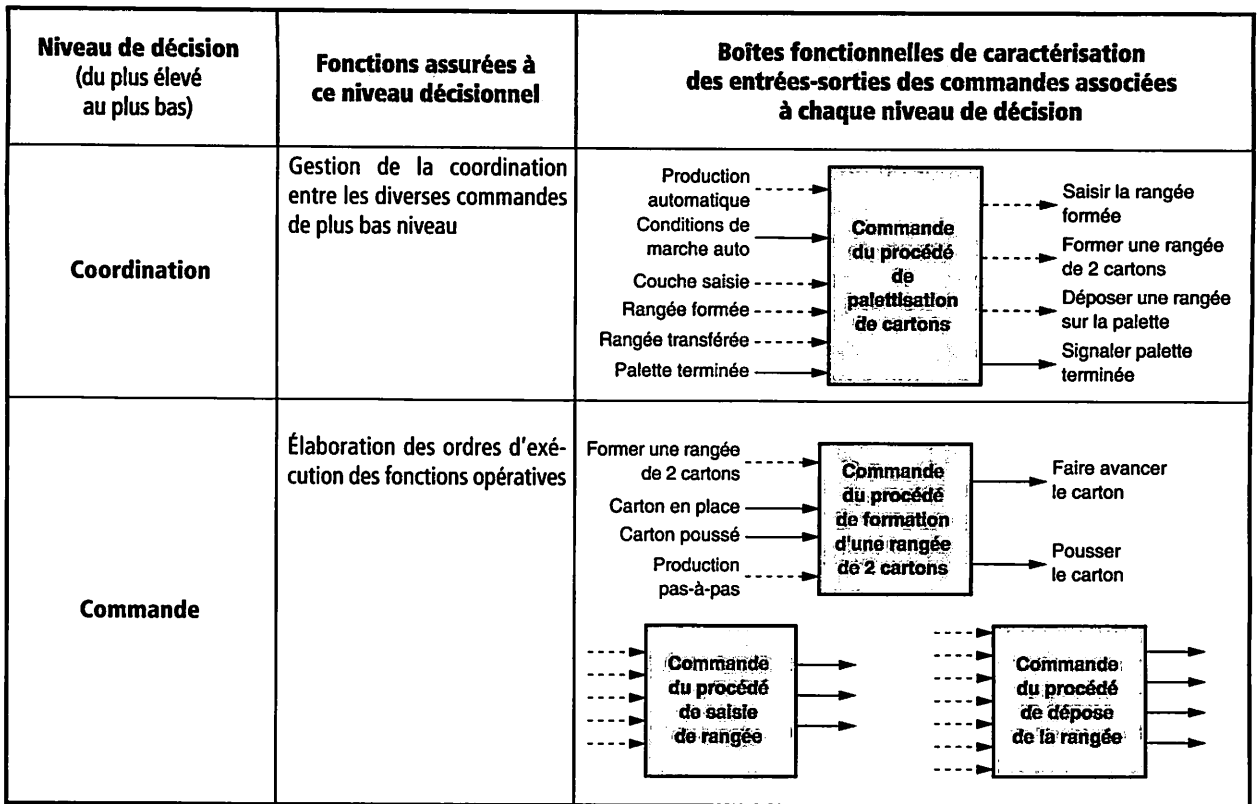


Figure 4.8. Exemple de hiérarchie décisionnelle couramment utilisée (analyse appliquée au palettiseur).

d) Typologie des modèles comportementaux

La définition de la frontière de description et la caractérisation des entrées et des sorties est un préalable à toute modélisation comportementale (paragraphe II.2 a) d'une commande de système automatique.

Ces entrées et ces sorties sont des variables de différentes natures (chapitre 2).

Pour les entrées :

- logiques (présence, seuil de pression, consignes de mode de marche et d'arrêt, ...);
- analogiques (mesures de températures, vitesses, distances, pressions, ...);
- numériques (acquisition de données de configuration telles que températures, durées, position, nombre de pièces, ...).

Pour les sorties :

- logiques (assignation, ordre d'effectuer un mouvement, d'allumer un voyant, ...);
- analogiques (températures, vitesses, distances, ...);
- numériques (affichage de températures, consignes numériques envoyées à des préactionneurs pour atteindre une position ou une vitesse, ...).

Il existe de nombreux outils de description du comportement, mais ils sont plus particulièrement adaptés à un type de variables (Figure 4.9) :

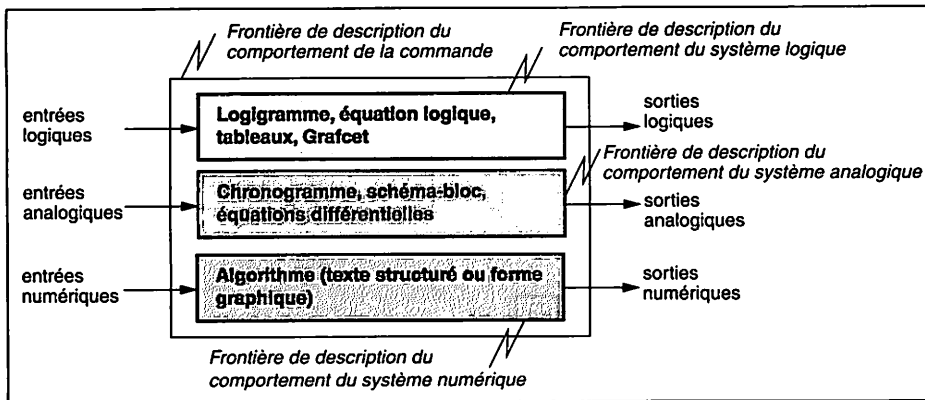
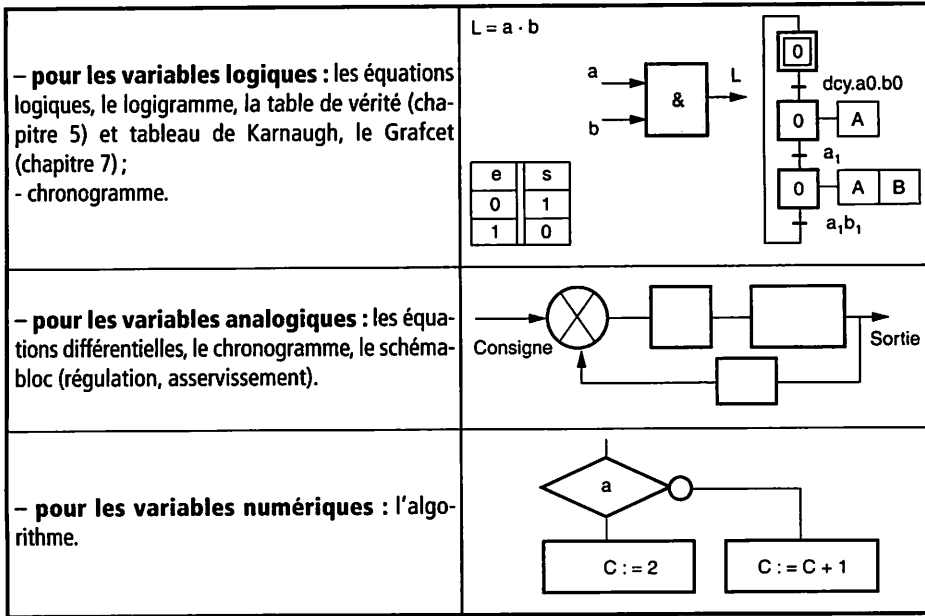


Figure 4.9. Adéquation du modèle comportemental utilisé avec la nature des entrées-sorties.

3. Représentation des structures

a) Structure de la commande

Il s'agit ici de toutes les descriptions qui permettent de définir la structure matérielle et logicielle d'une partie commande.

Les descriptions les plus couramment utilisées sont, par exemple :

- les schémas électriques de commande pour la définition des liens entre les constituants (mise en énergie, sécurités câblées, schémas de commande des actionneurs en sortie d'un automate programmable, parties commandes câblées, ...);
- les schémas de puissance électriques, pneumatiques et hydrauliques (pour la définition des liens entre constituants);
- les schémas d'implantation (des coupleurs et cartes montés dans un automate programmable, des éléments de dialogue sur le pupitre, des capteurs, des pré-actionneurs et des actionneurs sur la partie opérative, des constituants dans l'armoire électrique, des passages des câbles entre l'armoire de commande et les capteurs-actionneurs, ...);
- la structure logicielle de l'application développée dans un automate programmable (organigramme, graphes de structure, ...);
- l'architecture matérielle des réseaux et des bus de terrains utilisés;

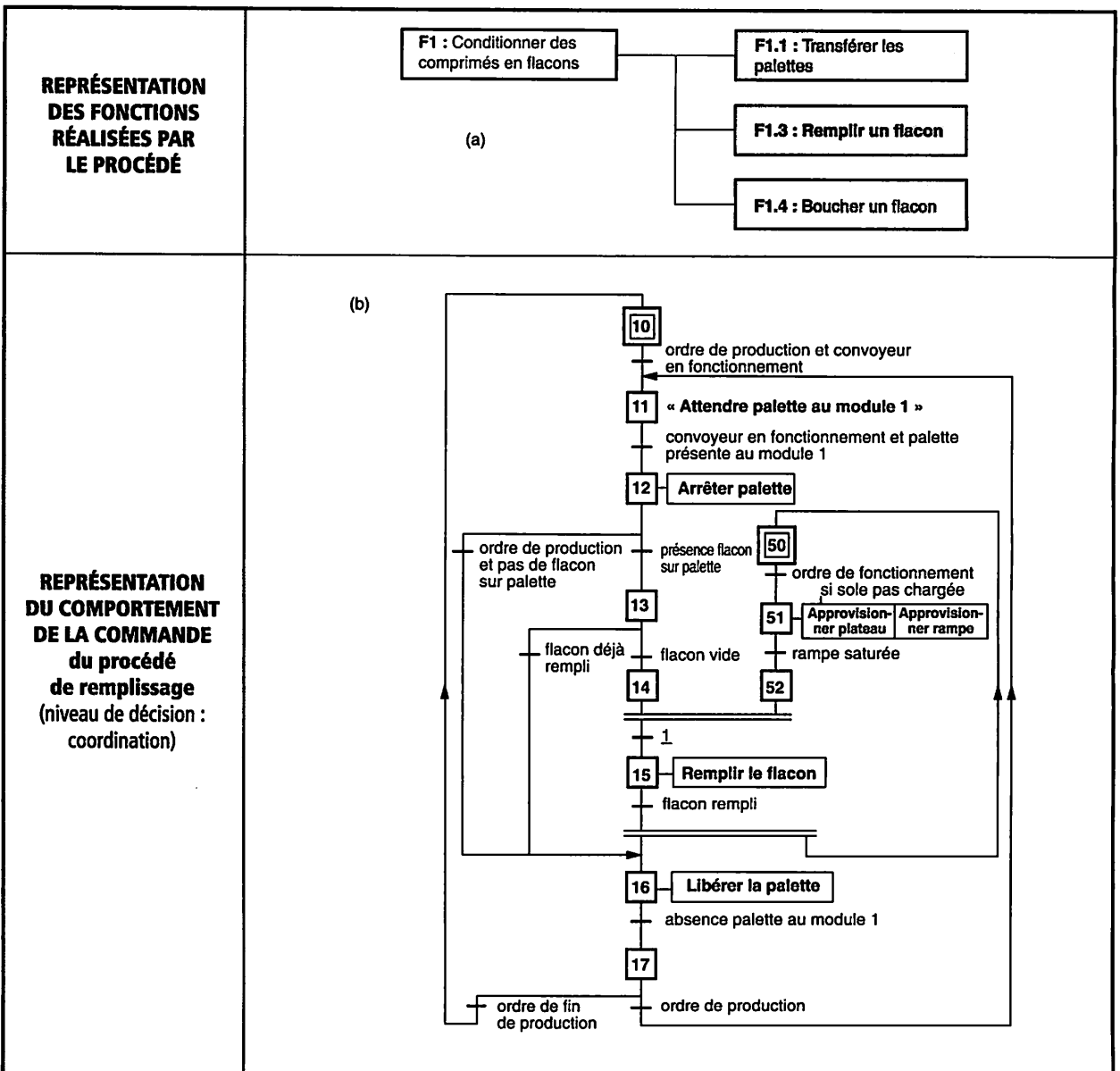
b) Schémas structurels de la partie opérative animée

Les automaticiens utilisent régulièrement des schémas structurels des sous-ensembles de la partie opératives faisant apparaître les actionneurs, les capteurs et parfois le produit. Ces schémas sont une aide à la compréhension du fonctionnement de ces sous-ensembles et à l'élaboration des modèles de comportement de la commande à concevoir.

La figure 4.10c illustre l'utilisation de ce type de représentation dans le cas du poste de remplissage de la machine de conditionnement (voir FICHE SYSTÈME 2).

La fonction procédé concernée est la fonction F1.3. La description du fonctionnement du sous-ensemble est réalisée à l'aide d'un grafcet. L'association de ces deux représentations au schéma structurel du poste de remplissage (Figures 4.10a, 4.10b, 4.10c), permet de leur donner du sens à celle-ci vis-à-vis de l'application.

Certains outils logiciels permettent d'animer ces schémas structurels à partir de la description comportementale fournie par le concepteur. Cette simulation peut être une aide à l'élaboration du modèle comportemental et à la vérification de la conformité au cahier des charges.



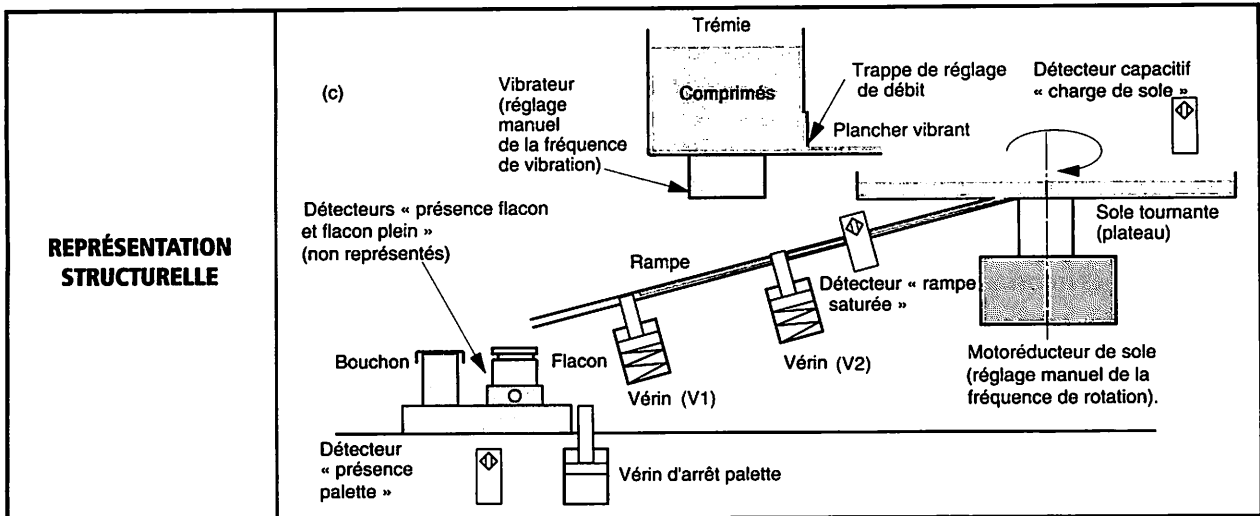


Figure 4.10. Utilisation de trois types de représentation pour la conception de la commande du poste de remplissage de la machine de conditionnement.

4. Représentation des informations

En entrée et en sortie d'une commande de système automatique, il est possible d'identifier trois catégories d'informations (chapitre 1, paragraphe 2).

En entrée :

- les comptes-rendus issus de la partie opérative ;
- les consignes opérateur ;
- les informations issues d'autres commandes ou systèmes de traitement de l'information.

En sortie :

- les ordres destinés à la partie opérative ;
- les signalisations opérateur ;
- les informations destinées à d'autres commandes ou systèmes de traitement.

Ces informations peuvent être logiques, analogiques ou numériques (chapitre 2).

Elles ont également d'autres caractéristiques :

- la rapidité de la variation des valeurs respectives des informations (température d'un four, détection du passage d'un bouchon vissé sur une bouteille entraînée par un convoyeur, ...);
- le débit d'informations (action sur un bouton poussoir « arrêt fin de journée », détection du passage de bouteilles accolées et entraînées par un convoyeur en vue de les compter, ...);
- les niveaux maximums et minimums des valeurs respectives associées aux informations ;
- ...

La réalisation de la conception de la commande et des choix technologiques associés imposent de connaître le sens de ces informations vis-à-vis de l'application (le nom peut être suffisant, par exemple : « présence carton ») et les caractéristiques respectives de celles-ci.

Recenser et caractériser les informations échangées entre la commande et l'environnement permet également :

- de leur donner du sens vis-à-vis de l'application pour faciliter la réutilisation des documents produits et le réinvestissement du travail effectué en conception pour d'autres applications similaires ;

- aux concepteurs de dimensionner un réseau après avoir notamment précisé les types de données échangées entre les parties commandes reliées en réseau, la fréquence d'échange, les contraintes de sécurité de transmission ;
- aux concepteurs de définir des bases de données qui pourront être utilisées par les systèmes de supervision pour l'aide au suivi de production et au pré-diagnostic de panne, ou encore par une partie commande nécessitant de nombreuses reconfigurations rapides ;
- la création de fichiers neutres facilitant l'échange entre des parties commandes ou des systèmes informatiques hétérogènes (non conçus à l'origine pour dialoguer ensemble).

À titre d'illustration (Figures 4.11 et 4.12), une simple analyse du pupitre opérateur du palettiseur et des éléments donnés **FICHE SYSTÈME I** permet d'identifier les catégories d'informations : consignes opérateur et signalisations opérateur ainsi que leur type (logique ou numérique).

Cette typologie a contraint les choix technologiques effectués pour réaliser le pupitre (Figure 4.12) : boutons poussoirs, clavier numérique, afficheur alpha-numérique (avec une capacité donnée),

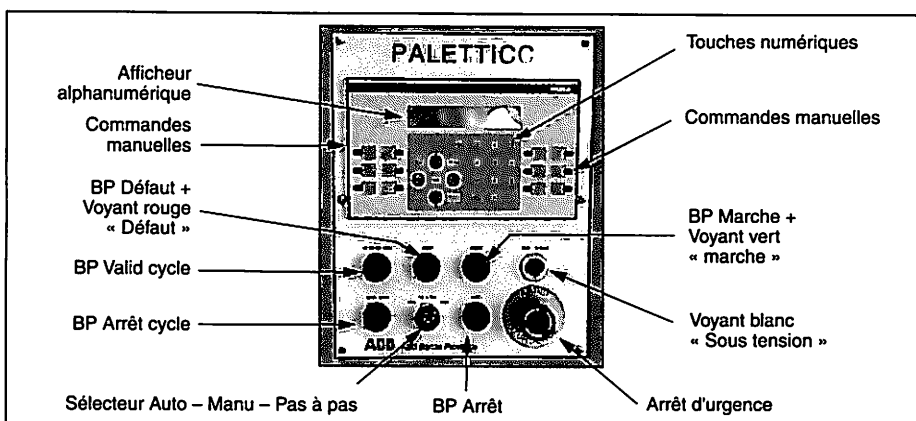


Figure 4.11. Organisation du pupitre du palettiseur (doc. Cybernetix).

Type d'information	Consigne opérateur		Signalisation opérateur	
	Information	Organe pupitre	Information	Organe pupitre
Logique	Marche	Bouton poussoir	Marche	Voyant vert
	Arrêt	Bouton poussoir	Sous tension	Voyant blanc
	Arrêt d'urgence	Bouton poussoir	Défaut	Voyant rouge
	Défaut	Bouton poussoir
	Arrêt cycle	Bouton poussoir		
	Valid cycle	Bouton poussoir		
Numériques	Manu (réservé aux personnes habilitées)	Sélecteur trois positions à clé		
	Commande convoyeur (action maintenue)	Touche F1		
	Départ séquence « prise d'origine » (action impulsionnelle)	Touche F9		
		
	Valeur de la hauteur carton	Clavier numérique	Message : « Hauteur carton ... mm »	Afficheur
	Nombre de rangs voulu	Clavier numérique	Message : « Nombre de rangs ... »	"
	"
			Message : « Nombre de rangs ... »	"
		

Figure 4.12. Identification et caractérisation de quelques informations échangées entre l'opérateur et la commande du palettiseur.

III. Approche descendante de la conception-réalisation de la commande

Au cours des différentes étapes du cycle de vie, de la conception générale à la réalisation, l'approche de conception de la commande est progressive. Elle est menée parallèlement à la définition des organes de la partie opérative. Ainsi, la frontière de description du comportement de la commande à décrire évolue et les entrées-sorties considérées sont caractérisées différemment selon la frontière concernée. Le point de vue dominant est celui du concepteur de la commande, même s'il est amené à intégrer d'autres points de vue dans son travail (conducteurs, régleurs, mainteneurs, responsables d'exploitation, ...). Dans l'approche progressive, trois grands objectifs de modélisation du comportement peuvent être identifiés :

- description du comportement de la commande du procédé;
- description du comportement de la commande d'organes de la partie opérative permettant d'agir sur le produit ou d'informer l'opérateur;
- description des ordres codés, élaborés par la commande programmée.

L'illustration s'appuie sur la fiche système 1 du palettiseur.

1. Commande du procédé

Étape du cycle de vie : conception générale.

Point de vue : du concepteur de la commande.

Objectif : décrire le comportement de la commande du procédé.

Il s'agit de décrire alors la gamme d'opérations (amener les cartons, ...) pour obtenir la valeur ajoutée, à partir d'informations sur l'état du produit et des opérations (cartons amenés, ...). Aucun effecteur, actionneur ou capteur n'est supposé connu. Ce travail d'analyse s'effectue au stade de la conception générale, ou lors de la description d'un système existant pour faire comprendre le procédé d'obtention du produit.

L'exemple, Figure 4.13, concerne la commande du procédé de formation d'une rangée de 2 cartons. Il s'agit donc de décrire la gamme d'opérations nécessaires (« faire avancer le carton », « pousser le carton ») pour former une rangée de 2 cartons, à partir des informations « carton en place », « carton poussé ». Les deux informations d'entrée associées à une flèche en pointillé sont issues de niveaux de décision supérieurs (conduite et coordination).

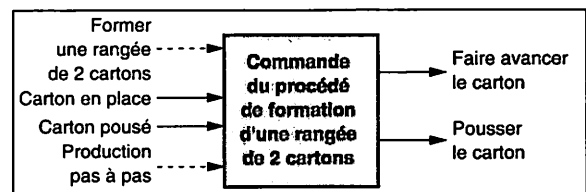


Figure 4.13. Commande d'un procédé.

REMARQUE

Cette commande peut être vue aussi comme une fonction de contrôle du processus (voir définition : chapitre 1 paragraphe II.2) de formation d'une rangée de 2 cartons. En effet, il s'agit de contrôler l'enchaînement et l'évolution temporelle ordonnés des procédés permettant de faire avancer le carton et de pousser le carton.

2. Commande des organes permettant d'agir sur la partie opérative et d'informer l'opérateur

Étapes du cycle de vie : conception générale et détaillée.

Point de vue : du concepteur de la commande.

Objectif : décrire le comportement de la commande des organes (mouvements des effecteurs, mouvements des actionneurs, pré-actionneurs) de la partie opérative permettant d'agir sur le produit ou d'informer l'opérateur (signalisation visuelle ou sonore destinée aux opérateurs).

Selon les choix techniques effectués sur la partie opérative, on peut distinguer trois niveaux de finesse dans cette analyse :

- *commande des mouvements des « effecteurs »* : les effets successifs attendus sur le produit et la PO (convoyage cartons...) sont décrits à partir d'informations sur l'état du produit et des effecteurs (cartons en butée...). L'architecture générale de la PO et les effecteurs sont connus, mais aucun actionneur ou capteur n'est supposé connu ;

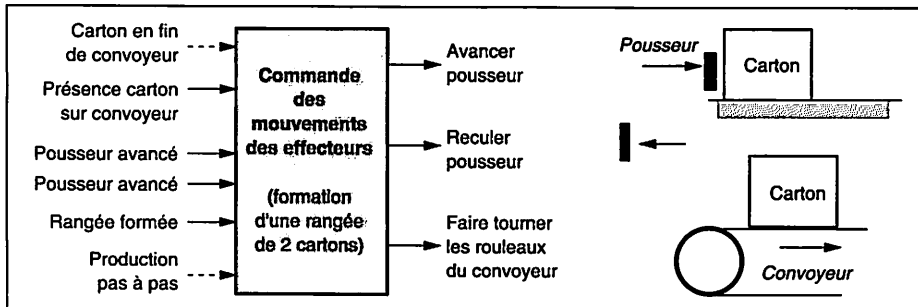


Figure 4.14. Commande des effecteurs permettant la formation d'une rangée de cartons.

- *commande des mouvements des « actionneurs »* : les comportements attendus des actionneurs (les actions : avance vérin V4, rotation moteur M6...) sont décrits à partir des informations logiques fournies par les capteurs (D11 actionné) et le pupitre. Les pré-actionneurs ne sont pas précisés, aucun élément technique n'est connu sur la partie commande ;

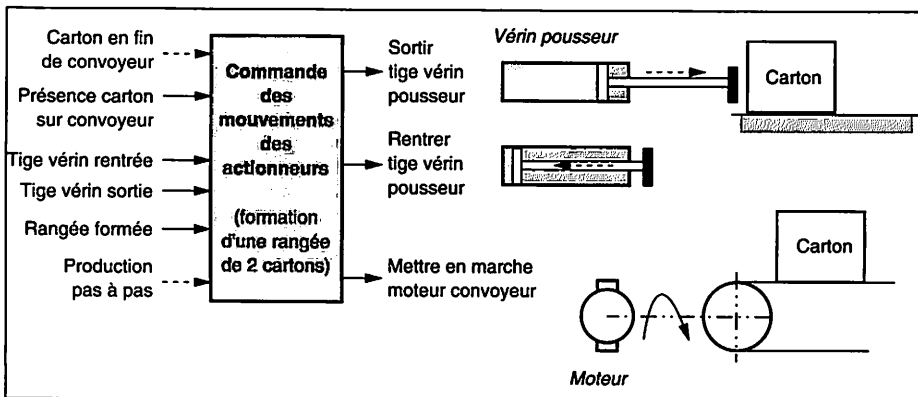


Figure 4.15. Commande des actionneurs permettant la formation d'une rangée de cartons.

- *commande des « pré-actionneurs »* : les comportements successifs des pré-actionneurs (EV4, KM6), donc des actionneurs (alimentation V4 sens +, alimentation V4 sens - réalisé grâce au retour par ressort de l'électrovanne, alimentation M6) sont décrits. A ce moment là, la partie opérative est entièrement connue.

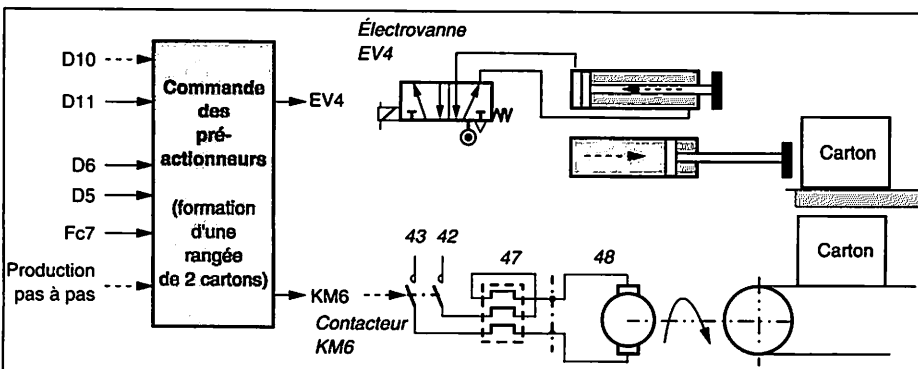


Figure 4.16. Commande des pré-actionneurs permettant la formation d'une rangée de cartons.

REMARQUE

Dans le cas de la figure 4.16, les repères utilisés pour les entrées et les sorties désignent les échanges informationnels entre la commande et les composants (capteurs, éléments du pupitre, pré-actionneurs) en utilisant les mnémoniques, plus faciles à employer que les repères technologiques. Ainsi le pilote du distributeur est repéré EV4, alors que le repérage technologique (sur schéma notamment) est 4YV1. Cette pratique, même si elle est commode, nécessite que la codification soit univoque.

3. Commande programmée

Étape du cycle de vie : réalisation.

Point de vue : du programmeur.

Objectif : décrire les ordres codés, élaborés par la commande programmée.

– Il s'agit de décrire comment sont élaborées les sorties de la commande programmable (%Q4.10; %Q2.3) à partir des entrées capteurs (%I3.3; %I3.4; %I1.14; %I1.13; %I3.5; %I3.12; %I3.13). C'est l'ensemble du traitement qui est alors pris en compte, y compris le traitement logique intégré dans les schémas câblés (mise en marche, sécurités, ...).

– La partie opérative n'est vue ici qu'à travers les entrées/sorties logiques de la partie commande, donc uniquement sous forme informationnelle. Souvent le concepteur est aussi le programmeur, car l'élaboration des modèles de comportement se fait directement dans des ateliers logiciels capables de générer automatiquement le code pour la commande programmée.

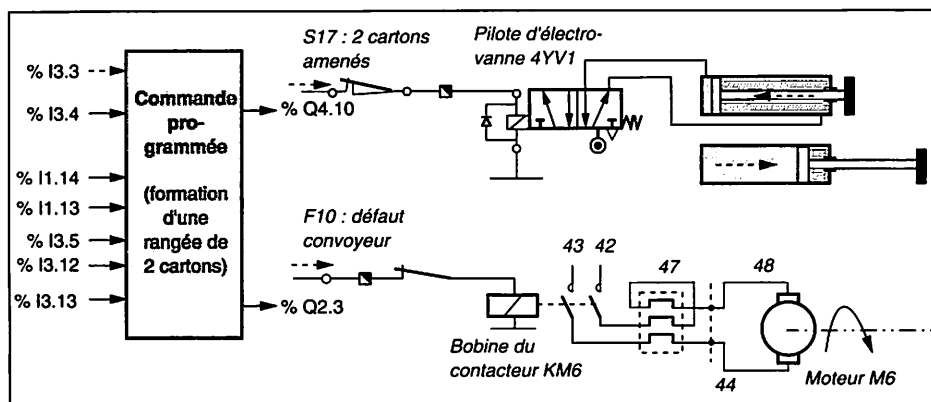


Figure 4.17. Élaboration des ordres codés, élaborés par la commande programmée.

Résumé

● Les différentes représentations

Chaque étape (activité) de spécification, conception du cycle de vie amène les intervenants à produire des modèles et des représentations (diagrammes, schémas, logigrammes, grafsets, algorigrammes, ...) pour constituer le dossier de conception nécessaire à la réalisation et aux tests du système. Une modélisation ou une représentation ne peut être pertinente que vis à vis des objectifs fixés au préalable. Donc, quoi que l'on représente au sujet du système, on ne fait qu'exprimer un point de vue, donc révéler un aspect de cette réalité (architecture matérielle générale du système, structure matérielle de la partie opérative, architecture fonctionnelle de la commande, sécurité, conduite, maintenance, ...).

Tous ces aspects s'expriment à partir de quatre types de représentation (Figure 4.18) :

- une *représentation des fonctions*, où une structure fonctionnelle relative au système est définie. Pour la conception d'une commande, la représentation des fonctions réalisées par le procédé (automatique) est privilégiée;
- une *représentation du comportement*, où les modes de marche et d'arrêt et le comportement de tout ou partie du système sont définis;
- une *représentation des structures*, où la partie matérielle est définie par des plans et des schémas (maquette numérique 3D, dessins d'ensemble, dessins de définition, schémas d'implantation, schémas d'architecture matérielle, ...);
- une *représentation des informations*, où les informations traitées et échangées dans le systèmes sont définies et caractérisées (type, format, fréquence d'échange, ...).

suite

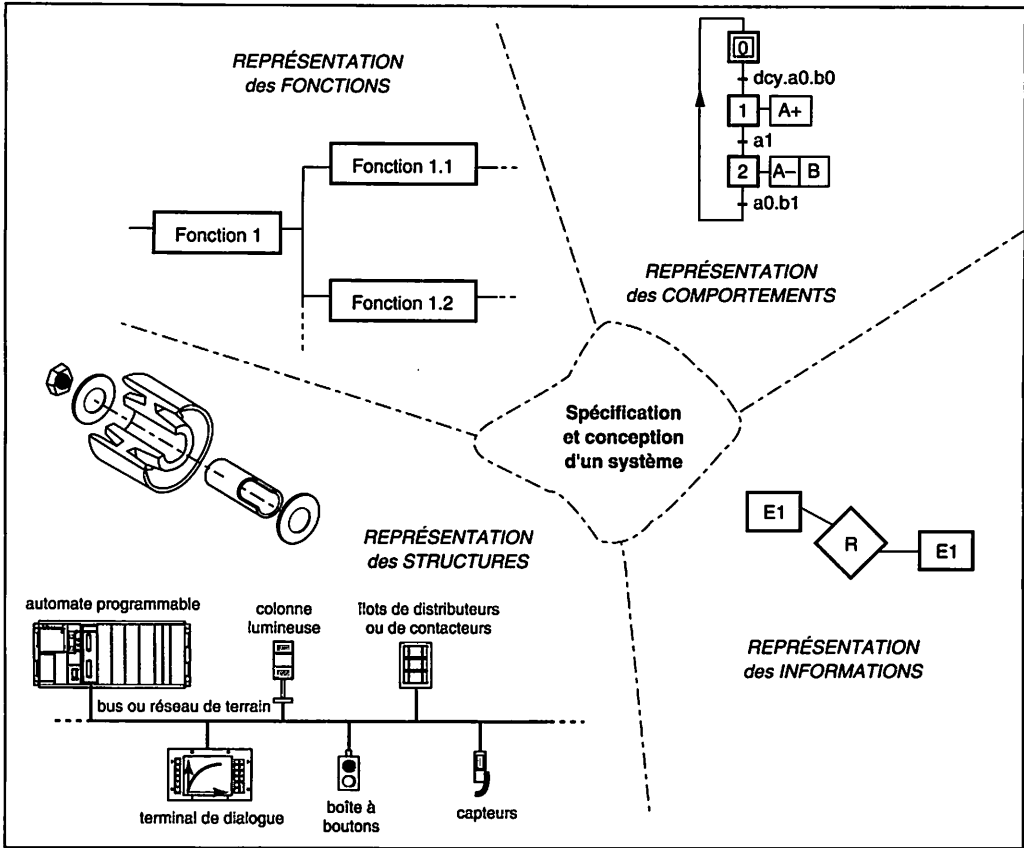


Figure 4.18. Illustration de la description d'un système automatique au travers de différents types de représentations.

● **Frontière d'isolement**

Quelque soit le type de représentation utilisé : fonctions, structures, comportements ou données, il est indispensable de définir les limites de ce que l'on s'attache à décrire (les données échangées entre deux commandes particulières, les fonctions assurées par le poste d'approvisionnement, l'architecture matérielle du bus de terrain utilisé, le comportement d'une commande donnée, ...).

La construction d'une représentation, qui apparaît comme étant à la base de toute activité d'étude d'un système complexe pour en révéler un aspect, nécessite donc de définir la **frontière de description** (isolement).

L'isolement de tout ou partie d'un système n'est pas une fin en soi, c'est un moyen pour satisfaire l'objectif de l'opération de représentation (ou de modélisation). Cette frontière (Figure 4.19) faisant apparaître un « intérieur » et un « extérieur », permet de caractériser l'élément isolé et de définir ses interactions avec l'environnement.

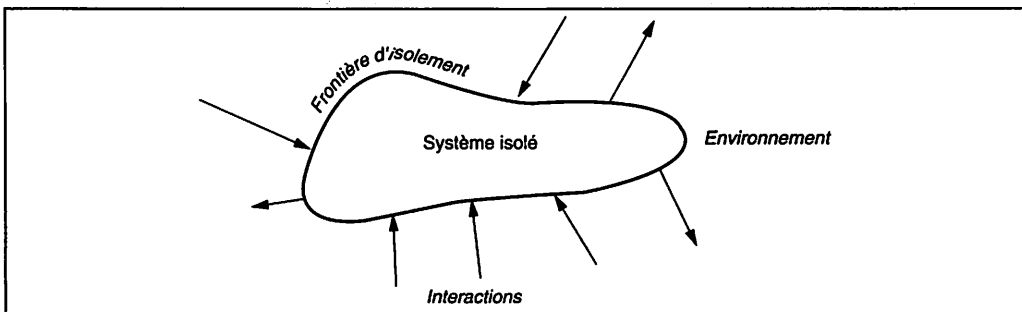


Figure 4.19. Frontière d'isolement.

Toute représentation ou modélisation doit ainsi préciser :

- les limites du système isolé;
- le point de vue dominant et l'objectif de modélisation;
- les relations pertinentes avec l'environnement (entrées, sorties).

● **Approche descendante de la conception réalisation de la commande**

Au cours des différentes étapes du cycle de vie, de la conception générale à la réalisation, l'approche de conception de la commande est progressive. Elle est menée parallèlement à la définition des organes de la partie opérative. Ainsi, la frontière de description du comportement de la commande à considérer évolue et les entrées-sorties considérées sont caractérisées différemment selon la frontière concernée. Le point de vue dominant est celui du concepteur de la commande, même s'il est amené à intégrer d'autres points de vue dans son travail (conducteurs, régleurs, mainteneurs, responsables d'exploitation, ...). Dans l'approche progressive, trois grands objectifs de modélisation du comportement peuvent être identifiés :

- description du comportement de la commande du procédé;
- description du comportement de la commande d'organes de la partie opérative permettant d'agir sur le produit ou d'informer l'opérateur;
- description des ordres codés, élaborés par la commande programmée.

Exercices

EXERCICE 1. Relever dans la FICHE SYSTÈME 2 concernant la machine de conditionnement, les paragraphes et les représentations caractéristiques de productions faites à l'issue des activités du cycle de vie suivantes : analyse du besoin, spécification du système, conception générale et détaillée, réalisation.

EXERCICE 2. Parmi toutes les représentations données dans la FICHE SYSTÈME 2, quelles sont celles qui relèvent de représentation du comportement, des structures et des données ?

EXERCICE 3. Quels sont les niveaux de décision utilisés dans la description du comportement de la commande de la machine de conditionnement (FS2) et les représentations associées ?

EXERCICE 4. Situer les représentations suivantes par rapport aux phases identifiées paragraphe III dans la démarche progressive de conception de la commande de la machine de conditionnement : FS2.9, FS2.10, FS2.11, FS2.21, FS2.24, FS2.25, FS2.30.

5

Comportement des systèmes logiques combinatoires

I. Systèmes logiques combinatoires

Une première analyse du comportement de la partie commande d'un store motorisé avec une commande utilisateur à boutons poussoirs (Figure 5.1) permet de constater que :

- l'action sur le bouton poussoir D_e permet d'enrouler le store ;
- l'action sur le bouton poussoir D_d permet de dérouler le store ;
- les actions simultanées sur D_d et D_e provoquent l'arrêt du store ;
- l'utilisateur doit maintenir le bouton poussoir appuyé tant que la manœuvre n'est pas terminée (store enroulé : S_e , store déroulé : S_d).



Figure 5.1. Store motorisé avec commande utilisateur à boutons poussoirs.

Les entrées - sorties de la commande sont celles indiquées sur la figure 5.2.

La valeur logique (vrai, faux) de chacun des ordres logiques « enrouler store » et « dérouler store » générés par la commande du store sera donc déduite des entrées logiques D_e , D_d , S_e et S_d (Figure 5.2).

Une représentation sous forme de table de vérité (Figure 5.3) permet de recenser diverses combinaisons d'entrées et de définir les valeurs logiques des sorties correspondantes.

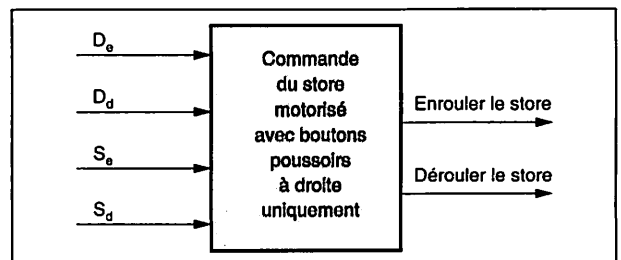


Figure 5.2. Bilan des entrées et des sorties de la commande du store motorisé.

Elle facilite également la conception de la commande en :

- formalisant le cahier des charges ;
- envisageant toutes les combinaisons d'entrées.

En effet, les quatre entrées amènent à étudier $2^4 = 16$ combinaisons possibles correspondant à des états du système fort variés :

- certaines combinaisons sont caractéristiques d'une utilisation courante du store, en action ou au repos : C0, C1, C2, C4, C6, C8, et C9 ;

- d'autres correspondent à une utilisation incorrecte des boutons poussoirs : C3, C5, C7, C10, C11 et C15. L'utilisateur appuie simultanément sur De et Dd, ou il demande l'enroulement «De» alors que le store est déjà enroulé «Se» ou inversement. Le concepteur doit impérativement prendre en compte tous ces cas afin d'éviter les commandes dangereuses.

- Enfin, on rencontre souvent des combinaisons « technologiquement » impossibles : C12, C13, C14, C15. Elles indiquent qu'un dysfonctionnement a lieu quelque part dans la chaîne d'information : défaut au niveau du capteur, du câblage, de l'acquisition, ou autres ($S_e = S_d = 1$ dans ce cas, signifiant que le store est à la fois enroulé et déroulé). Deux stratégies s'offrent au concepteur dans ce cas de dysfonctionnement :

+ soit il ne tient pas compte de ces combinaisons, les considérant de fait comme impossibles. Il peut alors affecter les sorties comme bon lui semble puisque le cas de figure n'est pas censé se produire ;

+ soit il prend en compte le défaut afin d'agir en conséquence, ce qui peut aller de la simple mise à 0 des sorties (stratégie adoptée dans notre exemple) jusqu'à la mise en œuvre d'une procédure d'alarme plus ou moins complexe.

		S_e	S_d	D_e	D_d	Enrouler le store	Dérouler le store	
Store en position intermédiaire	C0	0	0	0	0	0	0	Les actions incorrectes sur D_e et D_d ou bien l'apparition d'un défaut entraînent l'arrêt des commandes
	C1	0	0	0	1	0	1	
	C2	0	0	1	0	1	0	
	C3	0	0	1	1	0	0	
Store déroulé	C4	0	1	0	0	0	0	
	C5	0	1	0	1	0	0	
	C6	0	1	1	0	1	0	
	C7	0	1	1	1	0	0	
Store enroulé	C8	1	0	0	0	0	0	
	C9	1	0	0	1	0	1	
	C10	1	0	1	0	0	0	
	C11	1	0	1	1	0	0	
Informations contradictoires : Dysfonctionnement	C12	1	1	0	0	0	0	
	C13	1	1	0	1	0	0	
	C14	1	1	1	0	0	0	
	C15	1	1	1	1	0	0	

Figure 5.3. Table de vérité caractérisant le fonctionnement du store.

La table de vérité proprement dite se limite aux colonnes S_e , S_d , D_e , D_d , Enrouler le store, Dérouler le store.

Il est aisé de constater que quel que soit l'instant t_i considéré, une même combinaison d'entrées logiques provoque les mêmes effets en sortie de la commande (sorties logiques) : La commande considérée est un système logique combinatoire (ou système combinatoire).

Un système est combinatoire si et seulement si :

- toutes ses entrées et toutes ses sorties sont logiques,
- et la valeur logique de chaque sortie, à un instant t_i donné, peut être déduite directement de la valeur logique des entrées à cet instant t_i .

Le comportement d'un système combinatoire peut être décrit par un ensemble de fonctions logiques des variables d'entrée. Il est possible de simplifier certaines expressions de ces fonctions mathématiquement ou encore à l'aide de tableaux de Karnaugh (voir FICHE MÉTHODE 1 - « Déterminer les équations d'un circuit combinatoire »).



L'utilisation de tableaux de Karnaugh ou de tables de vérité se limite aux systèmes à faible nombre d'entrées, cinq ou six en général. Par exemple, pour décrire le fonctionnement du store en prenant en compte les boutons de droite et ceux de gauche, on doit théoriquement combiner 6 entrées. Ceci nécessite l'utilisation de tableaux de Karnaugh de taille importante ($2^6 = 64$ cases) et plus compliqués à utiliser.

REMARQUE

L'étude des systèmes combinatoires reste incontournable, car ceux-ci constituent le fondement des systèmes de traitement de l'information : codeurs, décodeurs, transcodeurs, unités arithmétiques et logiques, multiplexeurs, démultiplexeurs, comparateurs et bien d'autres composants réalisant des fonctions fondamentales. Les parties commandes de systèmes automatiques assurent également des fonctions relatives à la sécurité qui font appel à de nombreux traitements combinatoires.

II. Algèbre de Boole

C'est au milieu du XIX^e siècle que George Boole et Augustus De Morgan, mathématiciens et logiciens britanniques ont posé les fondements de la logique mathématique moderne appelée algèbre de Boole et de la logique des classes et des relations. L'algèbre binaire ou algèbre de Boole, traite de l'étude des variables à deux états (variables logiques) et des fonctions de ces variables (fonctions logiques). Les propositions logiques complexes du langage naturel peuvent toutes s'exprimer à partir d'expressions combinant des propositions logiques élémentaires avec les conjonctions de coordination ET, de disjonction OU, ainsi que la négation NON.

EXEMPLE

« Dérouler store » est vraie si, et seulement si « D_d est actionné » ET « D_e NON actionné » ET « S_d NON actionné ».

Il faudra attendre le XX^e siècle pour que le physicien américain Claude Shannon remarque en 1937 que la logique mathématique proposée par George Boole pouvait s'appliquer à l'étude des circuits électriques, les états logiques faux (0) et vrai (1) étant matérialisés respectivement par les états « ouvert » ou « fermé » du circuit.

1. Propriétés des opérateurs élémentaires « ET », « OU » et « NON »

L'algèbre de Boole définit les propriétés des opérateurs élémentaires. Les tableaux suivants décrivent ces propriétés et les illustrent avec leur correspondance électrique.

a) Opérateur logique « OU »

Appelée également disjonction ou union, elle s'applique à au moins deux variables logiques et son symbole est + (ex : $x + y$). Sa table de vérité pour deux variables x et y est donnée ci-contre (Figure 5.4).

x	y	$A = x + y$
0	0	0
0	1	1
1	0	1
1	1	1

Figure 5.4. Table de vérité de la fonction OU.

Propriétés de l'opérateur OU	Exemple	Illustration électrique
Commutativité	$x + y = y + x$	
Associativité	$(x + y) + z = x + (y + z)$	
Existence d'un élément neutre	$x + 0 = 0 + x = x$	
Existence d'un élément absorbant	$x + 1 = 1 + x = 1$	
Idempotence	$x + x = x$	
Complément	$x + \bar{x} = 1$	

Figure 5.5. Propriétés de l'opérateur logique « OU ».

b) Opérateur logique « NON »

Appelée également complémentation, la fonction « NON » est notée par une barre verticale située sur la (ou les) variable(s) à complémenter. \bar{x} (ou/x) se lit «x barre» ou «non x» (voir table de vérité, figure 5.6).

x	$B = \bar{x}$
0	1
1	0

Figure 5.6. Table de vérité de la fonction NON.

Propriété de l'opérateur NON	Exemple	Illustration électrique
Involution	$\bar{\bar{x}} = x$	Néant

Figure 5.7. Propriétés de l'opérateur logique « NON ».

c) Opérateur logique « ET »

Appelée également conjonction ou intersection (relation ensembliste), la fonction « ET » s'applique à au moins deux variables logiques. Son symbole est un point (ex : $x \cdot y$).

Sa table de vérité pour deux variables x et y est donnée figure 5.8.

x	y	$C = x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

Figure 5.8. Table de vérité de la fonction ET.

Propriétés de l'opérateur ET	Exemple	Illustration électrique
Commutativité	$x \cdot y = y \cdot x$	
Associativité	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	
Existence d'un élément neutre	$x \cdot 1 = 1 \cdot x = x$	
Existence d'un élément absorbant	$x \cdot 0 = 0 \cdot x = 0$	
Idempotence	$x \cdot x = x$	
Complément	$x \cdot \bar{x} = 0$	

Figure 5.9. Propriétés de l'opérateur logique « ET ».

d) Autres propriétés

Distributivité de l'opérateur ET par rapport à l'opérateur OU :

$$x \cdot (y + z) = x \cdot y + x \cdot z.$$

Distributivité de l'opérateur OU par rapport à l'opérateur ET :

$$(x + y) \cdot (x + z) = x + y \cdot z.$$



L'utilisation de symboles d'opérations logiques comme le « plus » pour le OU et le « point » pour le ET, utilisés par ailleurs en algèbre classique (multiplication et somme) peut être source de confusions. Par exemple :

- Idempotence de l'opérateur OU : $x + x = x$ (en algèbre usuelle : $x + x = 2x$)

- Idempotence de l'opérateur ET : $x \cdot x = x$ (en algèbre usuelle : $x \cdot x = x^2$)

Dans le cas où la confusion est possible, il est préférable de ne pas remplacer les fonctions logiques par leurs symboles :

- x OU $x = x$ plutôt que $x + x = x$

- x ET $x = x$ plutôt que $x \cdot x = x$.

e) Relations utiles

Les relations suivantes peuvent être déduites des propriétés décrites précédemment (voir FICHE MÉTHODE 1 : « Déterminer les équations d'un circuit combinatoire »). Elles permettent des simplifications rapides d'expressions logiques.

$x \cdot y + x \cdot \bar{y} = x$
$(x + y) \cdot (x + \bar{y}) = x$
$x + x \cdot y = x$
$x \cdot (x + y) = x$
$x + \bar{x} \cdot y = x + y$
$x \cdot (\bar{x} + y) = x \cdot y$

Figure 5.10. Simplifications d'expressions logiques.

2. Complémentation d'une expression logique.

Théorème de De Morgan

La fonction complémentaire d'une fonction logique f est la fonction \bar{f} (ou $/f$), des mêmes variables, qui prend la valeur complémentaire de celle de f pour chaque combinaison des valeurs des variables.

Le théorème de De Morgan s'exprime par deux relations :

1 – le complément d'une fonction OU entre deux variables logiques est égal à la fonction ET entre le complément des même variables logiques :

$$\overline{x + y} = \bar{x} \cdot \bar{y}$$

2 – le complément d'une fonction ET entre deux variables logiques est égal à la fonction OU entre le complément des même variables logiques :

$$\overline{x \cdot y} = \bar{x} + \bar{y}$$

REMARQUE

Ces deux relations se généralisent dans le cas où le nombre de variables est supérieur à 2.

EXEMPLE

$$\overline{x + y + z} = \bar{x} \cdot \bar{y} \cdot \bar{z}; \overline{x \cdot y \cdot z} = \bar{x} + \bar{y} + \bar{z}$$



Dans le cas où les variables logiques apparaissent dans une proposition, la complémentation peut entraîner des conclusions fausses. Ainsi la proposition « Si A alors B » ne permet pas d'affirmer que « Si NON A alors NON B ». Par exemple, le fait d'affirmer que tous les corbeaux sont noirs (« Si corbeau alors noir ») ne permet pas de déduire que tout ce qui n'est pas un corbeau ne peut être noir : la proposition « Si NON corbeau alors NON noir » est fautive (un morceau de charbon, donc « NON corbeau » est noir par exemple). Par contre, on peut inférer de la même proposition que tout ce qui n'est pas noir ne peut être un corbeau : La proposition « Si corbeau alors noir » étant vraie, la proposition « Si NON noir alors NON corbeau », est également vraie. Cette opération porte le nom de contraposition et la proposition obtenue s'appelle la contraposée de la proposition initiale.

3. Fonctions combinatoires de deux variables

Les fonctions suivantes (Figure 5.11), qui mettent en œuvre les propriétés présentées aux paragraphes 1 et 2 sont souvent utilisées dans les réalisations câblées et programmées des commandes et plus généralement des systèmes de traitement de l'information.

NON-ET			NON-OU			Inhibition			OU exclusif			Identité		
$A = \overline{x \cdot y}$			$B = \overline{x + y}$			$C = x \cdot \bar{y}$			$D = x \oplus y$ $D = x \cdot \bar{y} + \bar{x} \cdot y$			$E = x \odot y$ $E = x \cdot y + \bar{x} \cdot \bar{y}$		
x	y	A	x	y	B	x	y	C	x	y	D	x	y	E
0	0	1	0	0	1	0	0	0	0	0	0	0	0	1
0	1	1	0	1	0	0	1	0	0	1	1	0	1	0
1	0	1	1	0	0	1	0	1	1	0	1	1	0	0
1	1	0	1	1	0	1	1	0	1	1	0	1	1	1

Figure 5.11. Fonctions combinatoires de deux variables.

III. Représentation des fonctions combinatoires

Les fonctions logiques peuvent être définies à l'aide de :

- représentations fonctionnelles (algébrique, tabulée, symbolique graphique) ;
- représentations technologiques.

1. Représentations fonctionnelles

a) la forme algébrique

Il s'agit d'une équation logique littérale ou symbolique.

En prenant par exemple la fonction logique combinatoire :

$$S = e_1 \cdot \bar{e}_2 + \bar{e}_1 \cdot e_2 + e_1 \cdot e_2 \cdot \bar{e}_3.$$

Sous forme littérale : $S = [e_1 \text{ ET } (\text{NON } e_2)] \text{ OU } [(\text{NON } e_1) \text{ ET } e_2] \text{ OU } [e_1 \text{ ET } e_2 \text{ ET } (\text{NON } e_3)]$

Sous forme symbolique : $S = e_1 \cdot \bar{e}_2 + \bar{e}_1 \cdot e_2 + e_1 \cdot e_2 \cdot \bar{e}_3.$

b) La forme tabulée

Table de vérité et tableau de Karnaugh, donnant les valeurs de la fonction logique pour les 2^n combinaisons des n variables ;

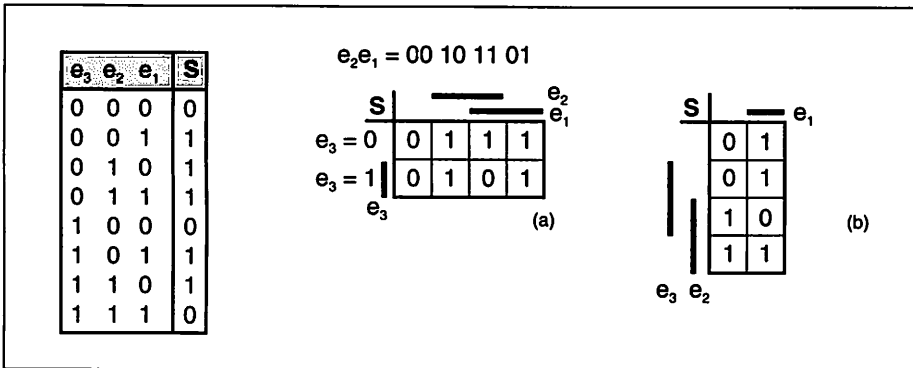


Figure 5.12. Représentations fonctionnelles tabulées de la fonction logique S : table de vérité et tableaux de Karnaugh.

Dans les tableaux de Karnaugh les combinaisons d'entrées sont ordonnées de manière à ne changer qu'une seule valeur logique d'entrée à la fois (par exemple pour e_1 et e_2 : 00, 01, 11, 10 au lieu de 00, 01, 10, 11) en passant d'une colonne à la suivante (ou d'une ligne à la suivante). Pour plus de détails, voir **FICHE MÉTHODE** I «Déterminer les équations d'un circuit combinatoire». Les valeurs logiques 1 et 0 peuvent être remplacées par respectivement un trait épais ou aucun trait afin de ne pas alourdir la représentation.

c) La forme symbolique graphique ou logigramme

Il s'agit de la traduction graphique de l'équation logique et dont les éléments sont les opérateurs de base (représentation normalisée, Figure 5.13).

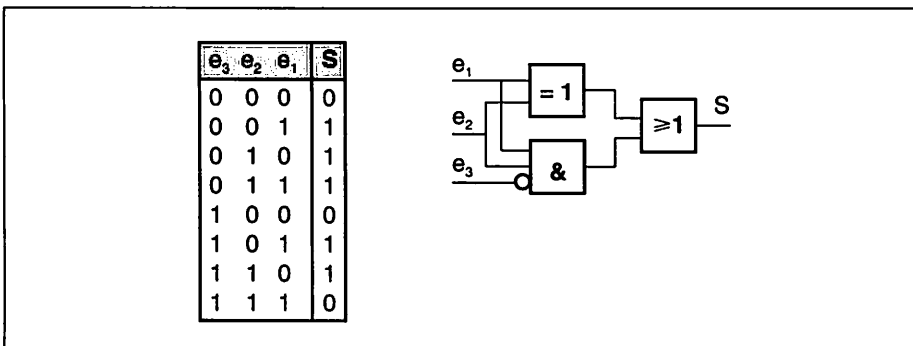


Figure 5.13. Représentations fonctionnelles de la fonction logique S : logigramme.

2. Représentations technologiques

Les formes de représentation des fonctions logiques varient selon les technologies.

a) En technologie électrique câblée

La représentation technologique est celle du schéma électrique à contacts (NF C03-20. Voir FICHE MÉTHODE 5 – Elaborer un schéma de commande ou de puissance), avec les principes de représentation logique suivants :

- une variable se représente par un contact NO, son complément par un contact NF;
- un opérateur ET se représente par deux contacts en série;
- un opérateur OU se représente par deux contacts en parallèle;
- la fonction de sortie se symbolise par un relais ou un voyant.

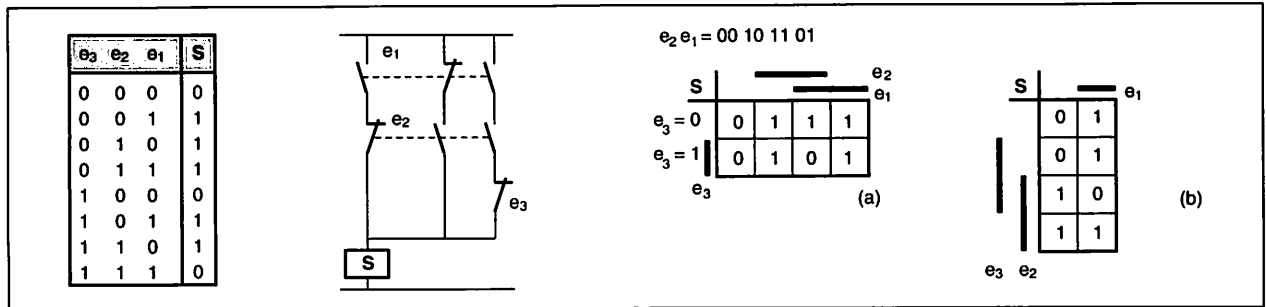


Figure 5.14. Représentation technologique de la fonction logique S : schéma électrique.

b) En technologie programmée

Le langage graphique normalisé (CEI 61131-3) est le « LD » (Ladder Diagram) ou langage à réseaux de contacts, utilisant une représentation particulière des contacts NO et NF, ainsi que des relais de sorties.

On rencontre également le langage FBD (Fonctional Block Diagram) défini par la norme CEI 61131-3. Il correspond à la forme graphique du logigramme.

c) En technologie électronique

La représentation normalisée est le logigramme, mais d'anciennes représentations d'origine américaine survivent encore dans de nombreux ouvrages, mêmes récents, et surtout dans la quasi-totalité des documentations de constructeurs.

REMARQUE

Le mot anglais « Ladder » signifie « échelle », d'où le terme parfois rencontré de « diagramme en échelle ».

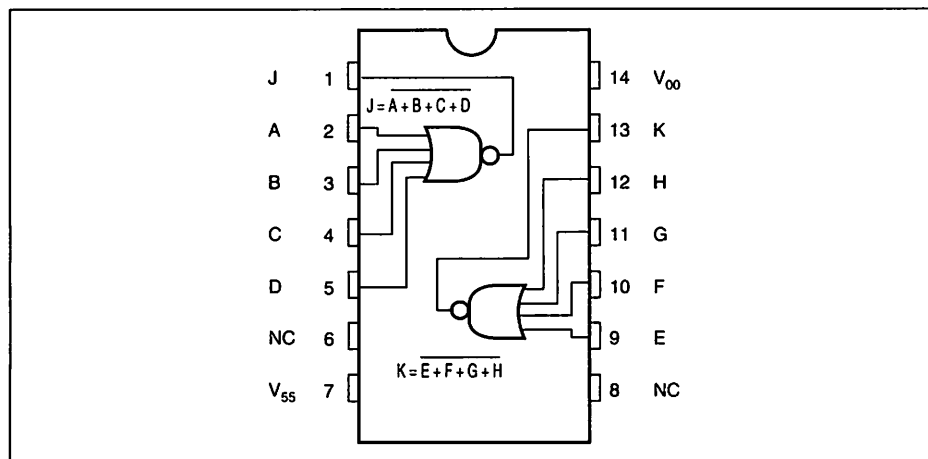


Figure 5.15. Représentation technologique d'un circuit HCF4002B réalisant deux fonctions NON-OU à quatre entrées chacune (document SGS-THOMSON MICROELECTRONICS).

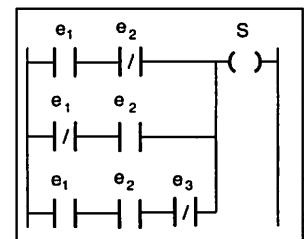


Figure 5.16. Représentation technologique de la fonction logique S : schéma en réseau de contacts (LADDER).

d) En technologie pneumatique

La représentation technologique est le logigramme.

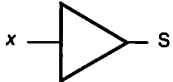
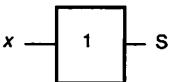

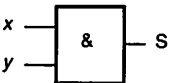

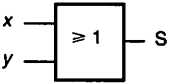
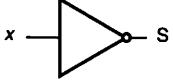
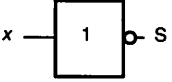

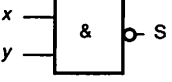

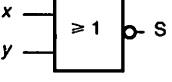

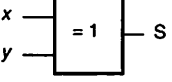
Circuit logique	Norme anglo-saxonne	Norme française
OUI $S = x$		
ET (AND) $S = x \cdot y$		
OU INCLUSIF (OR) $S = x + y$		
NON $S = \bar{x}$		
NON-ET (NAND) $S = \bar{x} + \bar{y}$		
NON-OU (NOR) $S = \bar{x} \cdot \bar{y}$		
OU EXCLUSIF (XOR) $S = \bar{x} \cdot y + x \cdot \bar{y}$		

Figure 5.17. Représentations normalisées des circuits logiques, logigramme.

IV. Exemples de systèmes combinatoires

1. Présentations générales des applications typiques

Les applications des fonctions logiques dans les systèmes de traitement de l'information sont très nombreuses :

- adressage de circuits logiques électroniques ;
- fonctions de codage, décodage, transcodage d'informations ;
- fonctions de commande de système (ou partie de système) combinatoire ;
- réceptivités et actions conditionnelles dans le modèle GRAFCET, etc.

La figure 5.18 met en évidence les fonctions réalisées par les principaux circuits combinatoires :

Type	Entrées	Sorties	Exemple
CODEUR n bits	2^n	n	
DÉCODEUR n bits	n	2^n	
TRANSCODEUR	n	p	
MULTIPLIXEUR 2^n entrées	$2^n + n + 1$ (données + adresses + validation)	1	
DÉMULTIPLIXEUR 2^n sorties	$1 + n + 1$ (données + adresses + validation)	2^n	
COMPARATEUR n bits	2_n	3 (<, >, =)	

Figure 5.18. Boîtes fonctionnelles de systèmes combinatoires usuels.

2. Décodeur d'adresses

● Fonction d'adressage d'un circuit logique. Décodeur

Un décodeur (n entrées, 2^n sorties) est un circuit dont une seule des sorties est activée pour chaque combinaison des variables d'entrée, appelées adresses dans ce cas. Le décodeur CA de la figure 5.19 permet de sélectionner une de ses sorties parmi 256 (2^8). Dans cet exemple, le circuit C doit être sélectionné en mettant à 1 son entrée «*sélect*», lorsque le mot de sortie affiché sur le bus d'adresses est \$6E, soit 01101110 en binaire. Les variables logiques associées à chaque voie du bus sont b_0 à b_7 . La proposition logique associée au besoin précédent est alors :

- SI (NON b_7) ET (b_6) ET (b_5) ET (NON b_4) ET (b_3) ET (b_2) ET (b_1) ET (NON b_0)
ALORS «*sélect*»;

- ou encore : $Sélect = \overline{b_7}.b_6.b_5.\overline{b_4}.b_3.b_2.b_1.\overline{b_0}$.

Le circuit décodeur CA, réalisé à partir de composants électroniques, permet de sélectionner 255 autres sorties qui ne sont pas représentées sur la figure 5.19.

● Application principale des décodeurs

Recherche d'un emplacement de mémoire. Le bus d'adresses d'un ordinateur contient l'adresse de l'emplacement de mémoire dans lequel on désire écrire une information binaire, le décodeur permettant de sélectionner cet emplacement.

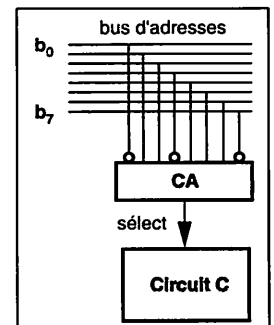


Figure 5.19. Sélection d'une adresse à partir d'un bus à huit voies.

3. Multiplexeurs et démultiplexeurs

a) Multiplexeur

La fonction d'un multiplexeur est de regrouper en série sur une voie unique les signaux parallèles provenant de n voies. La boîte fonctionnelle fait donc apparaître 2^n entrées de données (8 dans ce cas), n entrées de sélection (trois ici), une entrée de validation et une sortie unique.

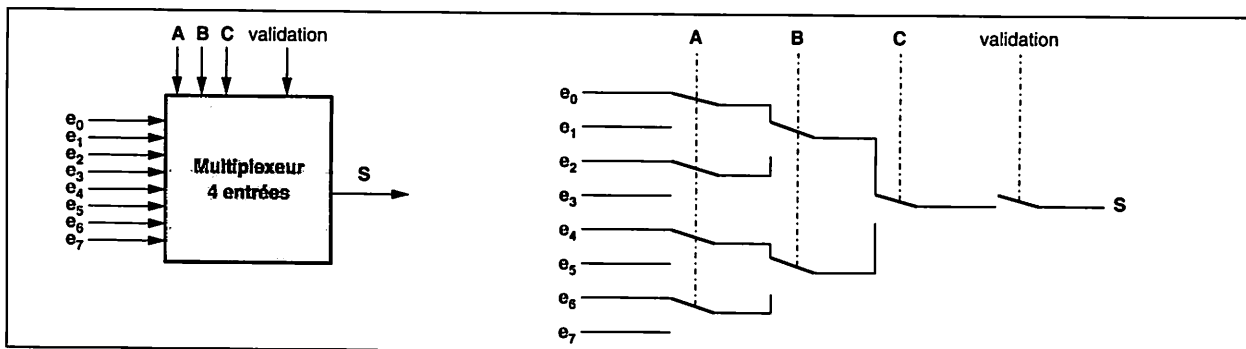


Figure 5.20. Principe du multiplexage.

Le schéma électrique équivalent montre clairement comment le sélecteur d'adresses ABC aiguille l'une des entrées sur la sortie unique. Le cas représenté correspond à la sélection de l'adresse 000 (tous les interrupteurs au repos) : l'entrée e_0 est envoyée sur la sortie S .

On remarque évidemment que, contrairement aux entrées A , B et C , l'état des entrées de données $e_7, \dots, e_3, e_2, e_1, e_0$ n'a aucune influence sur la sortie sélectionnée, puisque la fonction du multiplexeur est simplement d'aiguiller. L'écriture d'une table de vérité n'est pas nécessaire dans ce cas simple et le lecteur pourra vérifier aisément (en s'aidant de la figure 5.20 au besoin) que l'équation de la sortie est :

$$S = (e_0.\bar{A}.\bar{B}.\bar{C} + e_1.A.\bar{B}.\bar{C} + e_2.\bar{A}.B.\bar{C} + e_3.A.B.\bar{C} + e_4.\bar{A}.\bar{B}.C + e_5.A.\bar{B}.C + e_6.\bar{A}.B.C + e_7.A.B.C) \text{ validation.}$$

b) Démultiplexeur

La fonction d'un démultiplexeur permet de regrouper en série sur une voie unique les signaux parallèles provenant de n voies. La boîte fonctionnelle fait donc apparaître $2n$ sorties de données (8 dans ce cas), n entrées de sélection (A , B , C dans ce cas), une entrée de validation et une entrée E unique. Le principe du démultiplexeur est le symétrique de celui du multiplexeur.

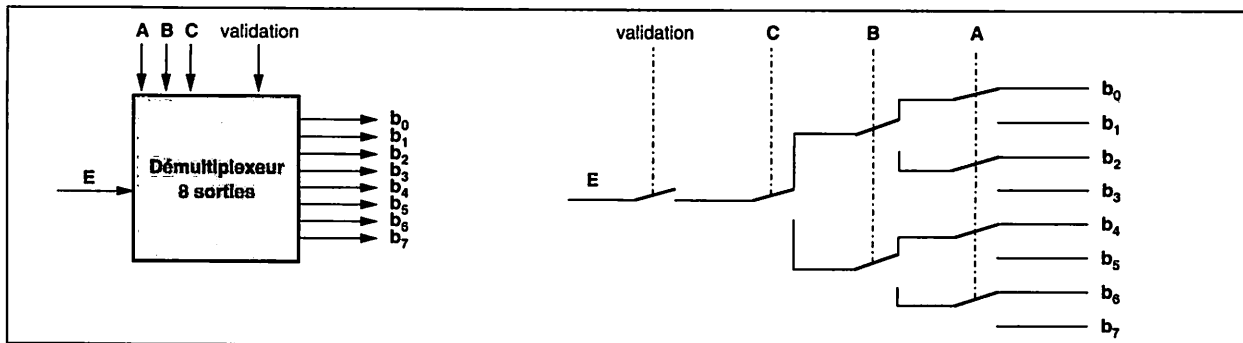


Figure 5.21. Principe du démultiplexage.

Les circuits de multiplexage et de démultiplexage possèdent de nombreuses applications : conversion parallèle-série et l'inverse, réalisation de fonctions logiques, transmission d'informations, etc.

L'application la plus connue concerne la transmission d'informations. On utilise souvent des lignes à fort débit pour transmettre « simultanément » les informations de plusieurs lignes à faible débit. Si la fréquence de multiplexage est suffisamment grande, le récepteur fonctionne comme si la transmission était continue.

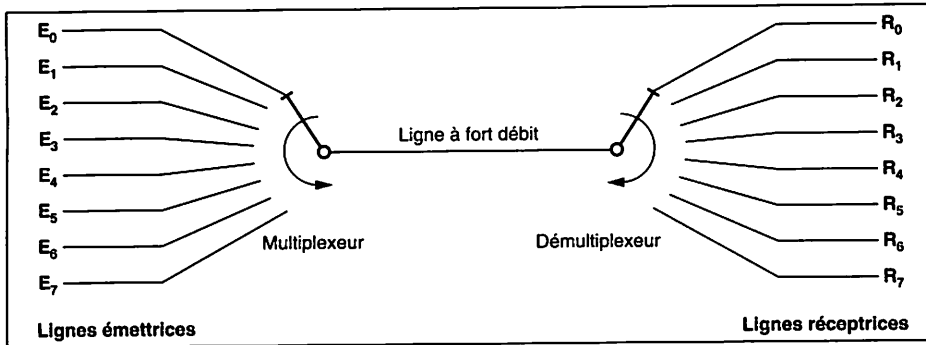


Figure 5.22. Transmission « simultanée » de plusieurs informations par multiplexage.

Pour assurer la transmission en continu, on fait évoluer l'adressage du multiplexeur par un compteur binaire cadencé par une horloge H.



Lors d'une transmission multiplexée d'informations, les circuits de multiplexage sont cadencés par une horloge et l'on peut considérer qu'il s'agit en fait d'un circuit séquentiel. Rigoureusement parlant, le multiplexeur tel qu'il a été défini ici est bel et bien un circuit combinatoire. C'est l'ensemble multiplexeur + compteur binaire + gestion du compteur qui est séquentiel.

4. Transcodeurs

a) Transcodeur code Gray-binaire naturel

Voir FICHE MÉTHODE I « Déterminer les équations d'un circuit combinatoire ».

b) Transcodeur binaire naturel – code Gray

En inversant la table de vérité du décodeur Gray vers binaire (FICHE MÉTHODE I), et en appliquant la méthode de simplification par tableaux de Karnaugh, on obtient les fonctions de transcodage permettant de passer du binaire au code Gray :

$$G_0 = B_0 \overline{B_1} + \overline{B_0} B_1 = B_0 \oplus B_1$$

$$G_1 = B_1 \overline{B_2} + \overline{B_1} B_2 = B_1 \oplus B_2$$

$$G_2 = B_2 \overline{B_3} + \overline{B_2} B_3 = B_2 \oplus B_3$$

$$G_3 = B_3$$

c) Commande d'un afficheur 7 segments : transcodeur BCD – 7 segments

Voir FICHE MÉTHODE I « Déterminer les équations d'un circuit combinatoire ».

- Un système est combinatoire si, et seulement si :
 - toutes ses entrées et toutes ses sorties sont logiques ;
 - la valeur logique de chaque sortie, à un instant t_i donné, peut être déduite directement de la valeur logique des entrées à cet instant t_i .
- Les fonctions (ou opérateurs) combinatoires élémentaires ET, OU et NON, permettent de décrire toutes les fonctions logiques. Il est de même pour chacune des fonctions NON-ET (NAND) et NON-OU (NOR).
- Toute fonction logique à n variables peut être représentée :
 - fonctionnellement sous l'une des formes équivalentes suivantes :
 - *algébrique* (équation logique ou langage littéral structuré) ;
 - *tabulée* (table de vérité ou tableau de Karnaugh donnant les valeurs de la fonction logique pour les 2^n combinaisons des n variables) ;
 - symbolique *graphique* (logigramme).
 - ou technologiquement sous l'une des formes suivantes :
 - schéma électrique ;
 - diagramme Ladder ;
 - logigramme.
- Les propriétés des fonctions logiques sont définies par l'algèbre de Boole. Le théorème de De Morgan s'exprime par les deux relations fondamentales : $\overline{x + y} = \overline{x} \cdot \overline{y}$ et $\overline{x \cdot y} = \overline{x} + \overline{y}$.
- Les applications qui font appel à la logique combinatoire sont très nombreuses. Tous les systèmes de traitement de l'information utilisent sur des composants électroniques programmables ou câblés qui s'appuient sur des circuits à base de portes logiques ET, OU, NON.
- Les équations logiques décrivant le fonctionnement de ces systèmes sont obtenues et simplifiées en utilisant des méthodes algébriques faisant appel aux propriétés de l'algèbre de Boole, ou en utilisant des méthodes graphiques faisant appel aux tables de vérité et aux tableaux de Karnaugh.

Exercices

EXERCICE 1. Vérifier les deux relations du théorème de De Morgan au moyen d'un tableau de Karnaugh.

EXERCICE 2. En utilisant les relations de l'algèbre de Boole, montrer que

$$\overline{a \oplus b} = \overline{a} \cdot \overline{b} + a \cdot b.$$

EXERCICE 3. Dresser le schéma Ladder ou à contacts correspondant à l'expression logique : $S = a\overline{b} \cdot (c \cdot d + e) + (e + \overline{f\overline{a}})$.

EXERCICE 4. Sachant que $\overline{a \oplus b} = \overline{a} \cdot b + a\overline{b}$, démontrer la propriété suivante $S = a \oplus b \oplus c = (a \oplus b) \oplus c = a \oplus (b \oplus c)$.

EXERCICE 5. Représenter à l'aide d'un tableau de Karnaugh l'expression logique suivante en vue d'une simplification : $L = abc + \overline{a}bc + a\overline{b}c + \overline{a}\overline{b}c$.

EXERCICE 6. Reproduire et compléter le tableau suivant dans le cadre d'une synthèse :

Associativité (opérateur OU)	$(a + b) + c =$
Associativité (opérateur ET)	$(a \cdot b) \cdot c =$
Commutativité (opérateur OU)	$a + b =$
Commutativité (opérateur ET)	$a \cdot b =$
Distributivité du ET par rapport au OU	$a(b + c) =$
Distributivité du OU par rapport au ET	$(a + b)(a + c) =$
Élément neutre du OU	$a + 0 =$
Élément neutre du ET	$a \cdot 1 =$
Élément absorbant du OU	$a + 1 =$
Élément absorbant du ET	$a \cdot 0 =$
Identités remarquables	$a \cdot a =$
	$a + a =$
	$a \cdot \bar{a} =$
	$\underline{\underline{a}} + \bar{a} =$
	$\bar{a} =$
	$a + a\bar{b} =$
	$a + ab =$
Théorème de De Morgan	$\overline{(a + b)} =$
	$\overline{(a \cdot b)} =$

Figure 5.23. Tableau de synthèse.

EXERCICE 7. En utilisant la table de vérité représentée figure 5.3, proposer un tableau de Karnaugh décrivant le comportement de la commande du store motorisé et donner l'expression logique de chaque sortie (commande par les boutons poussoirs de droite uniquement, avec mise à zéro des deux commandes en cas de dysfonctionnement).

6

Comportement des systèmes logiques séquentiels

I. Le concept d'état dans les systèmes séquentiels

1. Exemple 1

Il existe de nombreux systèmes industriels et « grand public » qui font appel à la logique séquentielle. Pour ne citer que deux exemples caractéristiques : la commande d'ouverture ou de fermeture de portières d'automobile à partir d'un seul bouton poussoir sur un boîtier de télécommande, l'allumage ou l'extinction des lampes à partir de différents boutons poussoirs. Dans ce dernier cas, une impulsion sur un bouton poussoir provoque l'allumage (si la lampe était éteinte), une seconde impulsion provoque l'extinction (si la lampe était allumée).

Afin de faciliter l'analyse qui permettra d'aborder le caractère séquentiel du système et le concept d'état, le système considéré se limite à un bouton poussoir bpt et une lampe L. Après avoir isolé le système étudié (la commande de la lampe, Figure 6.1), il est aisé de constater que les entrées et les sorties de ce système sont logiques.

Les chronogrammes figure 6.3 représentent un scénario possible d'évolution de la sortie L en fonction de l'évolution de l'entrée bpt.

Ils montrent que pour un même état (0) de l'entrée bpt considérée à deux dates différentes, la sortie L peut prendre deux états différents (1 ou 0). La valeur logique de la sortie, à une date donnée, ne peut donc pas être déduite directement de la valeur logique de l'entrée à cette même date. Le système logique considéré n'est pas un système logique combinatoire (chapitre 5) mais un système logique séquentiel.

Le même constat peut être fait dans le cas où l'entrée est à l'état 1, la sortie L peut prendre deux états différents selon la date considérée (voir également la table de vérité, Figure 6.2).

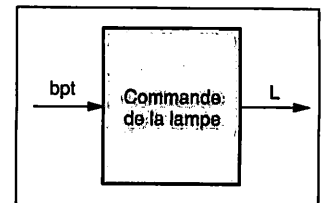


Figure 6.1. Boîte fonctionnelle de caractérisation des entrées/sorties de la commande de la lampe

bpt	L
0	0 ou 1
1	0 ou 1

Figure 6.2. Table de vérité de la variable L.

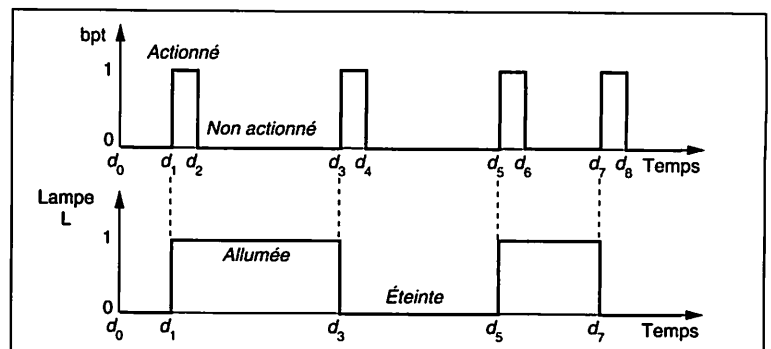


Figure 6.3. États successifs du bouton poussoir de commande du télérupteur et états correspondants de la lampe.

Un système logique est séquentiel si et seulement si :

- toutes ses entrées et toutes ses sorties sont logiques ;
- la valeur logique de chaque sortie, à une date donnée, dépend non seulement des valeurs logiques des entrées à cette même date, mais également de la valeur antérieure des entrées.

a) Notion d'état

La connaissance de toute l'histoire de l'évolution des entrées logiques depuis d_0 peut permettre le calcul de la valeur logique de la sortie quelque soit la date donnée, mais cette technique reste lourde à mettre en œuvre.

Chaque impulsion sur le bouton poussoir bpt de commande du télérupteur provoque un changement d'état de la sortie L, la sortie conservant cet état jusqu'à l'impulsion suivante. Le système a donc gardé la *mémoire* de l'impulsion sur bpt afin de pouvoir maintenir la sortie L à l'état 1.

Ainsi un système séquentiel est capable d'évolutions internes dont les diverses phases (en nombre fini), sont appelées **états**.

Définition : On appelle **état interne** d'un système, l'ensemble de l'information prenant en compte la mémorisation des évolutions passées et permettant les évolutions futures.

Un état interne est représenté par une combinaison de variables binaires encore appelées variables internes ou secondaires. Plusieurs solutions sont possibles pour le représenter. L'une d'entre elles, s'appuyant sur les notions vues dans le chapitre 5, est développée dans la suite de ce paragraphe, l'autre, développée dans le chapitre 7, s'appuie sur le langage GRAFCET.

Pour définir les variables logiques d'états, il est possible de passer par une représentation tabulaire particulière inspirée de la table de vérité (Figure 6.4) et nommée **matrice des phases primitives** qui précise les **états stables** (cerclés).

Un état stable est une situation permanente qui ne peut évoluer que par une action sur les variables d'entrées.

Cette matrice est remplie en commençant par définir l'état initial 1 et en précisant un état stable par ligne. On indique également les états transitoires ou instables (non cerclés) qui permettent le passage d'un état stable à l'autre.

L'état initial 1 où Bpt n'est pas actionné, est un état stable, il est donc cerclé. Le changement d'état (passage de l'état non actionné à l'état actionné) est caractérisé par un état instable 2 non cerclé. L'état stable 2 cerclé caractérise l'état où bpt est actionné. La même démarche est utilisée pour les états suivants.

REMARQUE

Autrement dit, dans un système logique séquentiel, à une même combinaison logique d'entrées et selon la date considérée, peut correspondre un ou plusieurs états logiques de la sortie considérée.

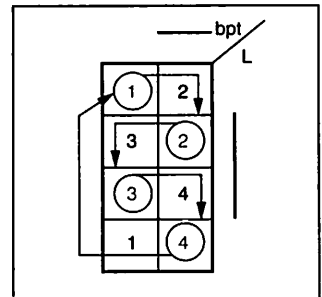


Figure 6.4. Matrice des phases primitives de L.

b) Codage des états

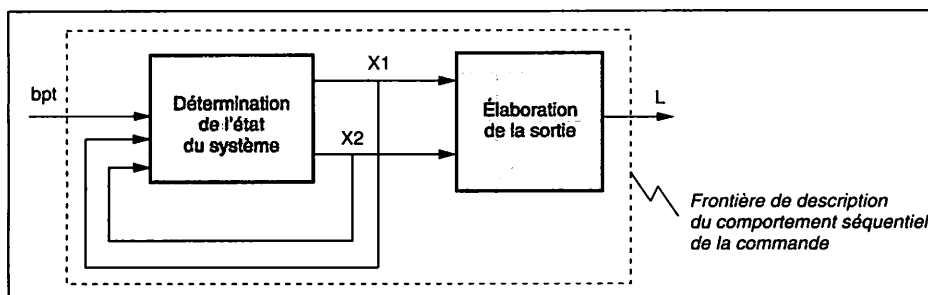


Figure 6.5. Décomposition de la commande séquentielle de la lampe en deux sous-systèmes logiques.

Cette opération consiste à affecter à chaque état une combinaison de variables, n variables pouvant discriminer 2ⁿ situations et permet la représentation du système en deux parties (Figure 6.5). La matrice figure 6.4 ne peut pas être simplifiée (pas de superposition de ligne possible). Les indéterminations correspondent aux lignes 1 et 3 et aux lignes 2 et 4, deux variables secondaires supplémentaires notées X1 et X2 sont donc nécessaires pour coder les 4 (lignes) états (Figure 6.6).

c) Détermination de l'état du système

La détermination de l'état du système s'effectue à partir de l'établissement des matrices relatives aux variables secondaires X2 et X1 et à la sortie L. Le codage employé sera celui du codage réfléchi (un seul changement) selon par exemple la répartition figure 6.6.

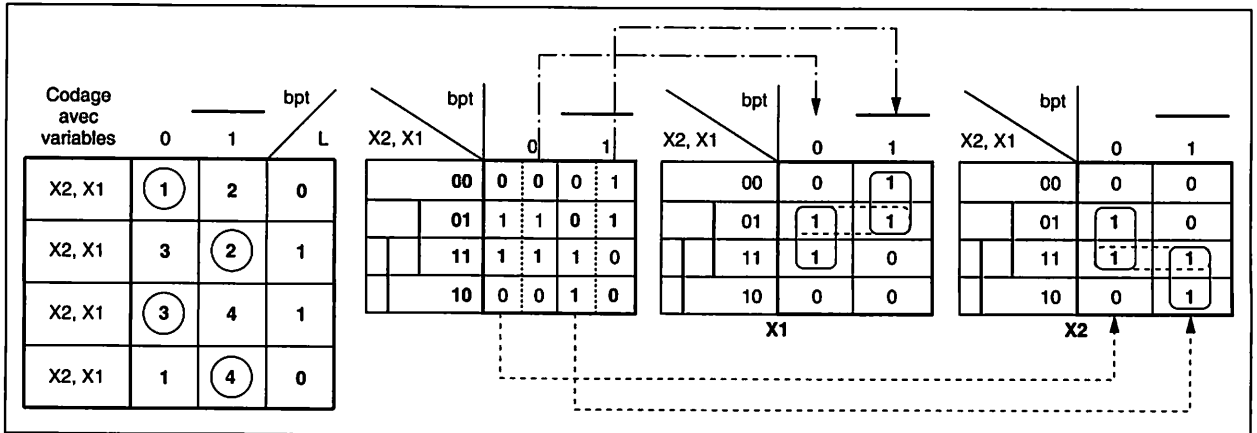


Figure 6.6. Représentation technologique de la fonction logique S.

Il s'agit de placer dans les cases correspondant aux états stables les combinaisons des variables secondaires correspondantes (par exemple 11 pour l'état ③). Puis, il faut placer dans les cases correspondant aux états instables, la combinaison des variables correspondant à l'état stable suivant, de même nom (par exemple 11 pour 3). Enfin, il suffit de retenir pour X2, les variables des colonnes de gauche et pour X1, les variables de colonnes de droite.

Les équations obtenues à partir des tableaux figure 6.6 sont donc les suivantes :

$$X2 = \overline{bpt} \cdot X1 + X2 \cdot X1 + bpt \cdot X2 \quad X1 = bpt \cdot \overline{X2} + X1 \cdot \overline{X2} + \overline{bpt} \cdot X1$$

Les états internes du système sont ainsi déterminés par la combinaison des variables X2 et X1 définies au-dessus.

d) Élaboration de la sortie

Pour établir le tableau relatif à la sortie, on affecte la valeur correspondante de la sortie pour les états stables à partir de la matrice des phases primitives. Pour les états transitoires (instables), lorsque le transitoire est entre deux états stables pour lesquels l'état de sortie ne change pas, on écrit pour ce transitoire ce même état de sortie.

Si ce transitoire est entre deux états stables pour lesquels les états de sortie sont différents, on donnera à l'état de sortie du transitoire une valeur indéterminée (∅), permettant les regroupements les plus intéressants (Figure 6.7).

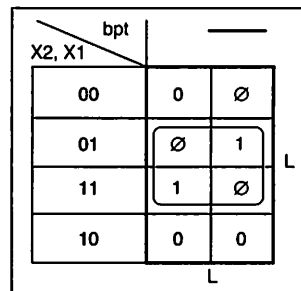


Figure 6.7. Utilisation d'états indifférents dans les regroupements.

REMARQUE

La prise en compte du groupement en pointillé, superflue dans un premier temps permet d'éviter les aléas de commutation.

REMARQUE

Dans cet exemple, la sortie ne dépend que de la variable interne X1. Dans le cas général. Les sorties du système séquentiel sont entièrement déterminées par une combinaison de l'état des entrées et de l'état interne du système.

e) Exemple d'une réalisation

Les expressions logiques des variables X1, X2 et L peuvent être exploitées pour passer à une réalisation câblée ou programmée. Cependant, la réalisation la plus courante dans ce cas consiste à utiliser un système électromécanique appelé télérupteur (Figure 6.8 à droite). Une impulsion sur l'un des boutons poussoirs bpt ou bpt1 de commande du télérupteur, permet d'allumer la lumière L (Figure 6.8 à gauche), l'impulsion suivante permet de l'éteindre. C'est un mécanisme interne au télérupteur qui assure le verrouillage de celui-ci dans une configuration donnée après chaque impulsion sur l'un des boutons poussoirs.

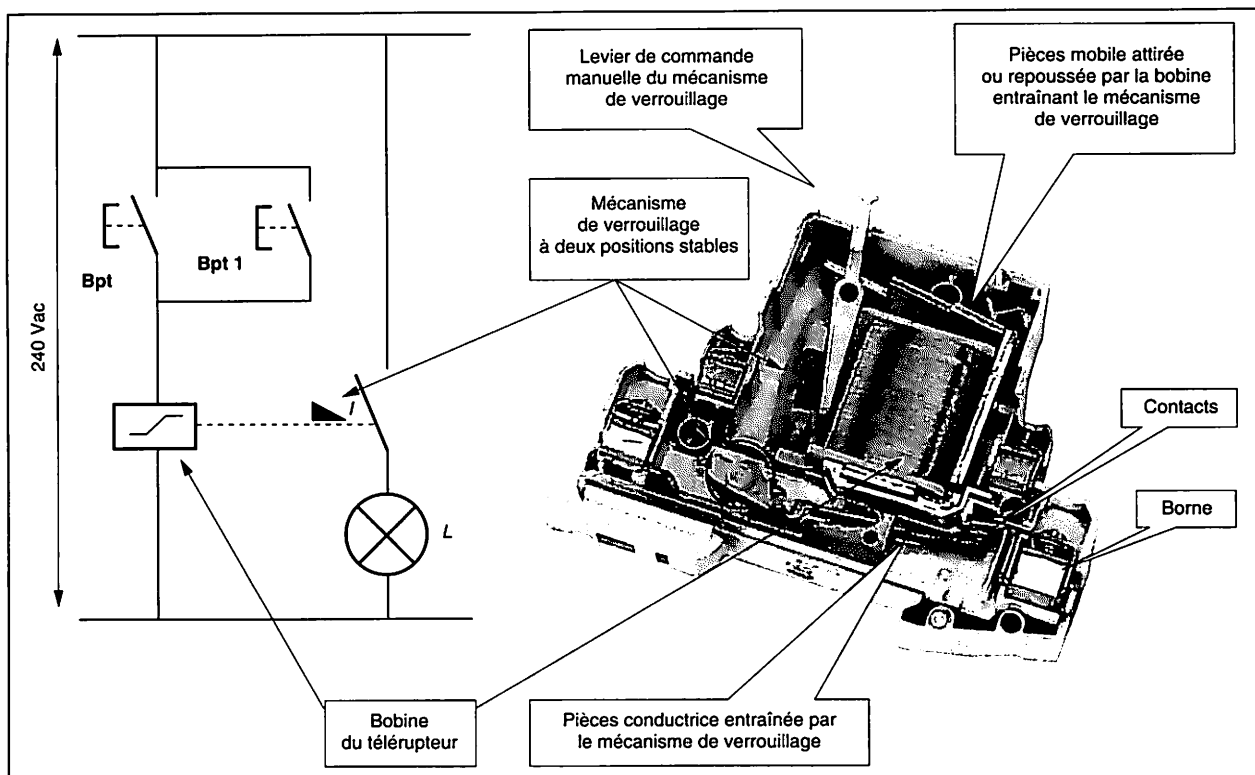


Figure 6.8. Télérupteur et sa commande électrique.

2. Exemple 2

Un opérateur dispose de deux boutons poussoirs « marche » et « arrêt » pour mettre en marche ou arrêter un moto-réducteur de convoyeur à bande.

- Une action sur le bouton poussoir « marche » permet de faire tourner les rouleaux du convoyeur ①.
- Ce convoyeur reste en marche après relâchement de ce bouton poussoir ②.
- Une action sur le bouton poussoir « arrêt » provoque l'arrêt du convoyeur ③, y compris dans le cas où, par mégarde, le bouton poussoir « marche » est simultanément actionné ④.

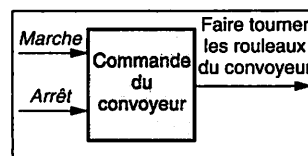


Figure 6.9. Entrées et sortie de la commande d'un convoyeur.

Les chronogrammes, Figure 6.10, permettent de préciser le comportement attendu de la sortie « Faire tourner les rouleaux du convoyeur » en fonction de l'évolution des entrées. Ces chronogrammes ne sont pas représentatifs de tous les scénarios possibles d'évolution des entrées, ils traduisent le cahier des charges notamment pour le fonctionnement le plus courant (dit normal) entre les dates d_0 et d_4 . Le tableau, Figure 6.11, aide à faire le parallèle entre les évolutions représentées dans ces chronogrammes et le cahier des charges.

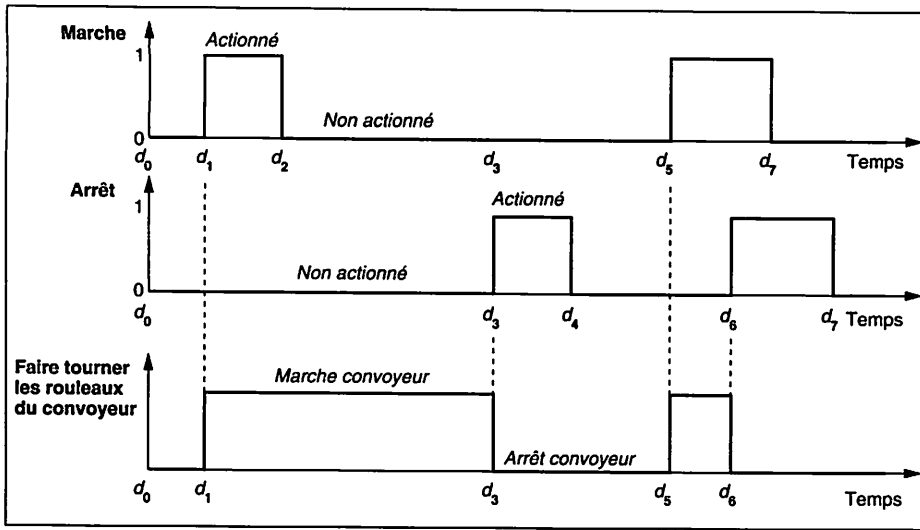


Figure 6.10. Chronogrammes des entrées et de la sortie de la commande du convoyeur. Pour faciliter la lecture, la sortie « Faire tourner les rouleaux du convoyeur » sera repérée : FTRC.

Cahier des charges	Temps t	Évolution des entrées	Évolution de la sortie
Point ①	$d_1 \leq t < d_2$	marche : état 1 arrêt : état 0	FTRC : état 1.
Point ②	$d_2 \leq t < d_3$	marche : état 0 arrêt : état 0	FTRC : état 1. <i>La sortie est maintenue à l'état 1.</i>
Point ③	$d_3 \leq t < d_4$	marche : état 0 arrêt : état 1	FTRC : état 0.
Point ④	$d_6 \leq t < d_7$	marche : état 1 arrêt : état 1	FTRC : état 0. <i>La priorité est donnée à l'arrêt.</i>

Figure 6.11. Parallèle entre les évolutions représentées dans les chronogrammes et le cahier des charges.

Une représentation à l'aide de la matrice des phases permet un recensement exhaustif de tous les scénarios d'évolution des entrées. Ainsi, par exemple, le cas d'un appui sur marche et arrêt est pris en compte (état 5).

a) Détermination de l'état du système

Il est aisé, également, de constater que pour une même combinaison d'entrées (lignes des états 1 et 3), la sortie FTRC peut prendre deux valeurs différentes.

Ces 5 lignes nécessitent 3 variables pour être différenciées, sauf à vérifier s'il est possible de réduire le nombre de lignes par fusionnement. Pour fusionner des lignes, il faut vérifier que :

- les phases stables correspondent à des combinaisons différentes des valeurs des variables d'entrée (colonnes différentes) ;
- toutes les transitions effectuées à partir des états stables conduisent aux mêmes états que dans la matrice primitive, donc que les lignes sont superposables.

Pour la matrice proposée, on peut fusionner les lignes 1, 4 et 5 et les lignes 2 et 3 permettant la réduction de la matrice à deux lignes.

Une seule variable X est donc nécessaire pour permettre la différenciation des états (Figure 6.13). Le système, ne connaissant que les événements d'entrées à une date donnée, utilise X pour savoir, à une date donnée, si le dernier événement est la mise à 1 de « marche » ou d'« arrêt ». X mémorise donc le dernier appui sur « marche ».

		Marche		
		Arrêt		
①	4		2	0
3		5	②	1
③	4	-	2	1
1	④	5		0
	4	⑤	2	0

Figure 6.12. Matrices des phases de FTRC.

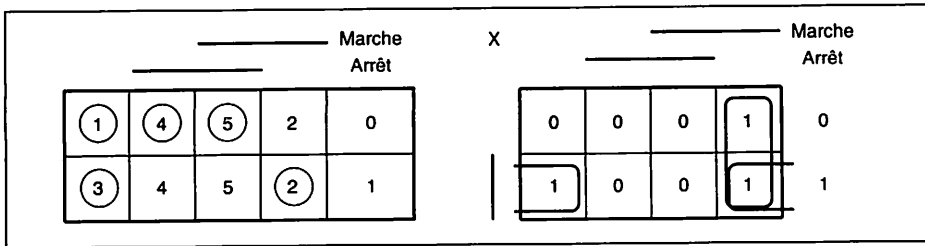


Figure 6.13. Matrice des phases de FTRC et tableau permettant de définir la variable secondaire X.

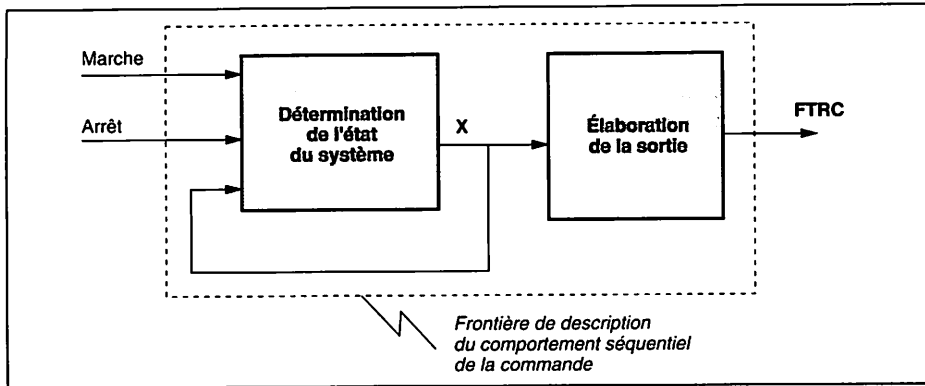


Figure 6.14. Décomposition de la commande séquentielle en deux sous-systèmes logiques.

L'équation de X est établie à partir du tableau X figure 6.13 est :

$$X = \text{marche} \cdot \overline{\text{arrêt}} + X \cdot \overline{\text{arrêt}} \text{ qui peut s'écrire : } X = (\text{marche} + X) \cdot \overline{\text{arrêt}}$$

Il s'agit d'un résultat tout à fait classique, où l'auto-maintien est réalisé par l'utilisation de la même variable (ici X) des deux côtés du signe d'égalité.

b) Élaboration de la variable de sortie

Le regroupement utilisant les états indéterminés permet d'écrire : $FTRC = X$.

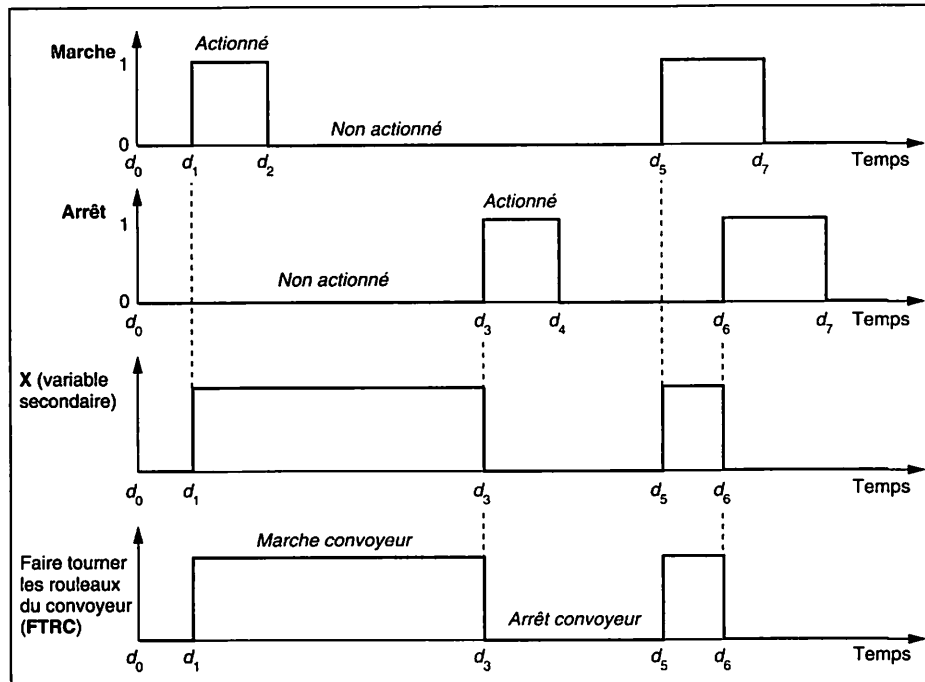


Figure 6.16. Chronogrammes des entrées, de la variable interne et de la sortie de la commande du convoyeur.

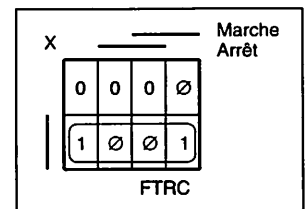


Figure 6.15. Tableau de FTRC.

REMARQUE

L'identité entre FTRC et X fait qu'on peut utiliser la variable de sortie comme variable secondaire. Cette facilité est très souvent employée, notamment dans les cas de commande de moteur par contacteur.

Trouver et établir les équations des diverses variables n'est pas un problème simple dès lors que le nombre d'entrées et la complexité de la matrice des phases primitives augmentent. Une autre solution consiste à utiliser un outil de description du comportement des systèmes séquentiels plus adapté comme le Grafcet (chapitre 7).

c) Exemple d'une réalisation

Un exemple de réalisation câblée est donné paragraphe II.b. Figure 6.21.

II. Les fonctions logiques séquentielles

1. Fonction mémoire

La fonction mémoire est représentée symboliquement de la manière suivante :

Le symbole général d'une mémoire est un rectangle divisé en deux parties par un trait pointillé (Figure 6.17). La partie du haut est représentative de l'état 1 (état actif) de la mémoire, la partie du bas est représentative de l'état 0 (état inactif).

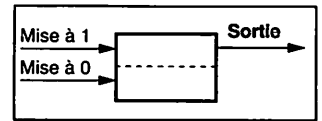


Figure 6.17. Symbole général de la fonction.

Les deux entrées : mise à 1 et mise à 0 de la mémoire permettent respectivement de l'activer et de la désactiver.

a) Mémorisation par auto-maintien. État de la mémoire

Le modèle logique de fonctionnement d'un moteur est donné par la proposition : « Si appui sur marche OU (NON appui sur arrêt ET moteur en marche) ALORS Faire tourner moteur. En associant respectivement les variables logiques marche, arrêt, m et M , aux propositions précédentes ($m \equiv$ moteur en marche, $M \equiv$ Faire tourner moteur), la fonction logique f représentant ce modèle est :

$M = f(\text{marche}, \text{arrêt}, m) = \text{marche OU (NON arrêt ET } m)$ peut encore s'écrire symboliquement :

$$M = \text{marche} + \overline{\text{arrêt}} \cdot m.$$

L'équation logique $M = \text{marche} + \overline{\text{arrêt}} \cdot m$ est le modèle de fonctionnement d'un système à deux entrées (marche et arrêt) et une seule sortie M (Figure 6.18) : son chronogramme montre qu'il s'agit bien d'une mémoire, dite par auto-maintien.

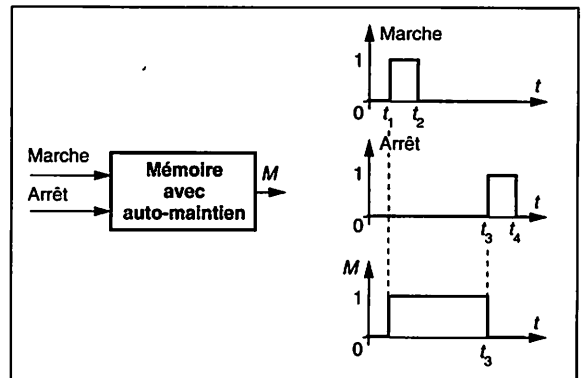


Figure 6.18. Représentations d'une mémoire par auto-maintien.

Quelle signification donner alors à la variable m (pour *moteur en marche*) introduite dans l'équation logique et dans les schémas logiques de la figure 6.19 ?

Par opposition aux entrées/sorties, qui sont des variables **externes** au système, m est une **variable interne** : elle est élaborée par le système lui-même pour construire la sortie. La proposition logique *moteur en marche* correspond à l'état du moteur : elle est vraie si le moteur a été mis en marche, par *appui sur marche*, et qu'il a démarré.

Comment l'élaborer ? On pourrait imaginer installer sur l'arbre moteur un capteur détectant qu'il tourne effectivement (capteur de vitesse, ou de position, par exemple), ou encore intégrer un capteur de courant sur un bobinage.

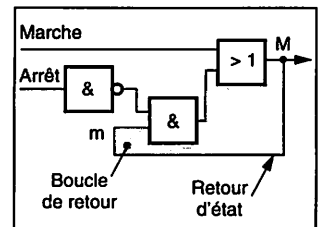


Figure 6.19. Construction d'une mémoire à auto-maintien.

Mais une telle solution serait en contradiction avec la définition du système, à deux entrées seulement ! Une solution consiste à retenir que le moteur est en marche lorsque la commande de mise en marche est vraie, c'est-à-dire lorsque la sortie M est vraie : la variable interne m est alors l'état réel de la sortie M , ce qui se représente par une liaison appelée retour d'état sur le circuit logique, qui présente alors une boucle (Figure 6.19).

L'équation d'activation de la mémoire montre que l'entrée marche est prise en compte quelque soit l'état de l'entrée arrêt : on dit que la marche est prioritaire sur l'arrêt.

Cette mémoire est dite à **marche prioritaire**.

Le concept d'état se retrouve dans le circuit électrique du contacteur auto-entenu (identifié en conséquence KM) de la figure 6.20a : « m » est matérialisé par un contact de sortie du relais KM, et son état logique matérialise bien l'état de la commande KM (qui est associée à d'autres contacts électriques de sortie du relais).

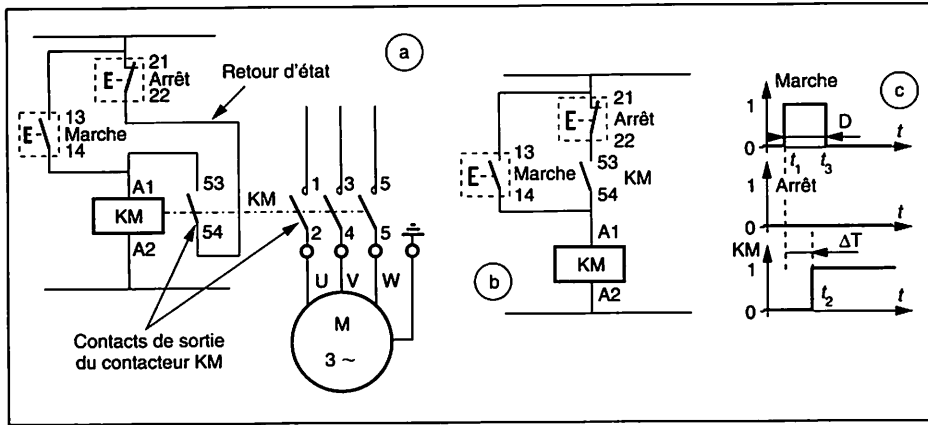


Figure 6.20. Auto-maintenance d'un contacteur utilisé comme mémoire.

b) Mémoires par auto-maintenance. Mémoire prioritaire à l'arrêt. Relais maître

L'équation logique $M = \text{marche} + \overline{\text{arrêt}} \cdot m$ apparaît comme un OU logique entre deux termes :

- le premier, ici *marche*, correspond à la partie activation de la mémoire;
- le second, qui s'exprime à partir de l'état m de la mémoire, est la partie auto-maintenance du modèle de fonctionnement de la mémoire.

Cette propriété se généralise à toute mémoire par auto-maintenance, de modèle général :

$$M = \text{équation d'activation} + \text{équation d'auto-maintenance} \\ = f + g \cdot m;$$

où f et g sont des fonctions combinatoires des entrées (Figure 6.21a).

La figure 6.21b présente le modèle logique d'une autre mémoire par auto-maintenance :

$$M = (\text{marche} + m) \cdot \overline{\text{arrêt}};$$

[activation : $\text{marche} \cdot \overline{\text{arrêt}}$; auto-maintenance : $\overline{\text{arrêt}} \cdot m$].

Son équation d'activation montre que l'entrée marche n'est prise en compte que si l'entrée arrêt n'est pas vraie : on dit que l'arrêt inhibe la mise en marche, ou encore que l'arrêt est prioritaire sur la marche.

Cette mémoire est dite à **arrêt prioritaire**.

REMARQUE

- dans les schémas électriques le contact de retour d'état porte par convention le même nom que le relais et la boucle d'état n'est pas représentée par souci de lisibilité des schémas (Figure 6.20b). Mais il est clair qu'elle sera matérialisée physiquement lors du câblage. Il est également à noter que, compte tenu du temps de réponse ΔT du contacteur au passage à 1 de l'entrée marche, celle-ci doit être de durée D suffisante pour le contact KM établisse le signal d'auto-maintenance $\overline{\text{arrêt}} \cdot \text{KM}$ (Figure 6.20c);

- dans les schémas et équations logiques, on convient soit de représenter la variable d'état en minuscule (la sortie étant en majuscule) pour insister sur le fait qu'elle est équivalente à une entrée, soit, de préférence, d'utiliser la lettre x (comme X_i dans le GRAFCET).

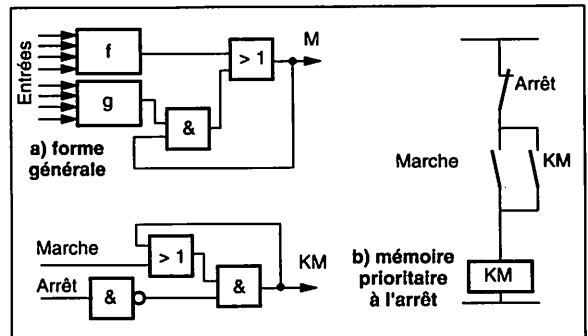


Figure 6.21. Principe de mémoire par auto-maintenance.

Ce type de mémoire auto-maintenue à arrêt prioritaire occupe une place particulière dans les systèmes automatisés : on la trouve systématiquement en tête de tout circuit d'alimentation en énergie des actionneurs (électrique, pneumatique,...) des équipements, imposée de fait par la normalisation européenne en matière de sécurité des machines (normes EN 292, EN 954-1, EN 60204-1). C'est la réalisation type du relais maître, ou relais autoalimenté de mise en énergie (Figure 6.21b).

EXEMPLE

Relais maître de la machine de conditionnement dans son ancienne version sans module « préventa » (Figure 6.22).

Son équation de commande est $KA1 = (SH3 + KA1) \cdot B1 \cdot SC1 \cdot SC2 \cdot AU \cdot DJ1$.

- SH3, associé à un bouton poussoir de réarmement, correspond à l'entrée marche (le choix du terme réarmement, plutôt que marche, indique bien que cette mémoire à un comportement orienté vers l'arrêt);

- toute information d'arrêt (de surcourse : SC1 ou SC2, d'arrêt urgence, ou de défaut moteur) interrompt l'alimentation en énergie;

- en cas de coupure d'énergie, le relais n'étant plus alimenté, le contact d'auto-maintenance KA1 retombe; il faut alors « réarmer » par appui sur SH3.

L'alimentation en énergie du palettiseur est également assurée par un relais autoalimenté : voir la fiche système du palettiseur.

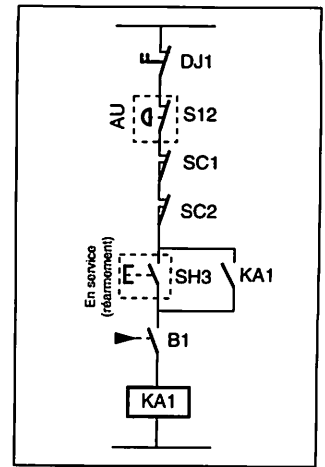


Figure 6.22. Relais maître de la machine de conditionnement (ancienne version sans module Préventa).

c) Mémoires bistables. Bascules r-s

Un second type de mémoire est la mémoire « deux états stables », ou « bascule », dont la caractéristique est de posséder deux états stables, d'où son nom de « bistable ». À chaque état est associée une sortie (Figure 6.23). Les deux entrées sont conventionnellement désignées par s (abréviation de l'anglais *set*) et r (abréviation de *reset*) :

- s est l'entrée d'activation (ou entrée de mise à un, équivalente à l'entrée *marche*);

- r est l'entrée de désactivation (ou entrée de mise à zéro, équivalente à l'entrée *arrêt*).

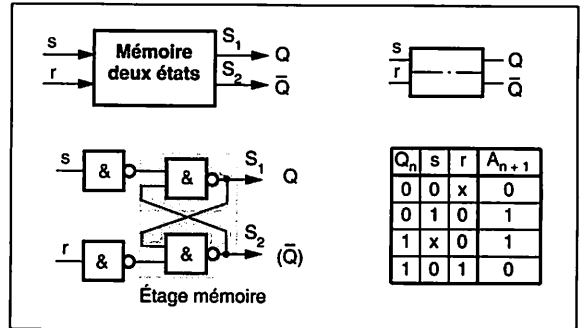


Figure 6.23. Mémoire à deux états stables (bascule r-s).

Les deux sorties, s_1 et s_2 , sont généralement notées Q et \bar{Q} (ou \bar{Q} , complément de Q).

Une réalisation simple d'une telle mémoire, la bascule r-s, à partir de quatre NAND (NON ET), est donnée par le schéma logique de la figure 6.20. Il fait apparaître deux retours d'état (avec deux variables d'états s_1 et s_2 , d'où le nom de mémoire deux états).

Le modèle logique de cette bascule s-r est défini par les deux équations :

$$S_1 = (s + \bar{s}_2) \text{ et } S_2 = (r + \bar{s}_1).$$

Ces équations montrent que, tant que $s = r = 1$, les deux sorties S1 et S2 prennent simultanément la valeur 1 : la notation (Q, \bar{Q}) des sorties n'a alors plus de sens et cette combinaison doit donc être interdite. La table de vérité de la figure 6.23 donne l'état futur Q_{n+1} de la bascule r-s à la date t_{n+1} (état futur aussi noté Q_+), en fonction de l'état actuel Q_n à la date t_n (ou Q_-), en se limitant aux seules combinaisons autorisées.

Il existe des variantes de cette bascule :

- avec priorité à l'activation ($S_1 = s + \bar{s}_2, S_2 = r \cdot \bar{s} + \bar{s}_1$);
- avec priorité à la désactivation ($S_1 = s \cdot \bar{r} + \bar{s}_2, S_2 = r + \bar{s}_1$);
- avec interverrouillage si $s = r = 1$ ($S_1 = s \cdot \bar{r} + \bar{s}_2, S_2 = r \cdot \bar{s} + \bar{s}_1$);
- à deux étages mémoire avec un fonctionnement en deux temps (bascules maître-esclave), dont l'étude sort du cadre de cet ouvrage;
- ou encore synchronisées à partir d'une « horloge ».

● Mémoires bistables « mécaniques » (pré-actionneurs, capteurs, effecteurs)

Si après avoir placé un élément mécanique dans une position donnée, on l'y maintient par un procédé quelconque (frottement, blocage, attraction magnétique, gravité, etc.), l'effet obtenu est un effet mémoire. En effet, la position reste maintenue après disparition de la cause motrice. Une mémoire bistable peut alors être obtenue à partir d'un même mouvement en créant deux maintiens sur deux positions extrêmes opposées. Cette technique est utilisée pour créer de nombreux constituants bistables : distributeurs 4/2 bistables (voir chapitre 10, paragraphe III), interrupteurs, capteurs de sens de déplacement, pinces.

d) Mémoires des constituants programmables

Une partie de la mémoire de données des constituants programmables est organisée pour le traitement logique. Chaque unité de mémoire, adressable séparément, peut être lue, écrite (mise à un) et effacée (mise à zéro). On y distingue deux types de « mémoires » :

- les mémoires « monostables », remises à zéro à chaque cycle de traitement, servent à mémoriser, pendant un cycle de traitement, le résultat d'équations combinatoires; elles sont associées au calcul des réceptivités et des actions;
- les mémoires « bistables », type s-r, qui conservent leur état d'un cycle au suivant tant que l'une ou l'autre des commandes d'activation ou de désactivation n'est pas vérifiée.

EXEMPLE

Commande de moteur (Figure 6.24).

La commande de désactivation de la mémoire bistable M associée à la sortie commandant le moteur s'effectue à partir de la variable arrêt = $a_1 + a_2 + a_3$ (ou $\text{arrêt} = \overline{a_2} \cdot \overline{a_2} \cdot \overline{a_3}$). La variable arrêt est associée à une mémoire monostable. Le programme est écrit en langage LD (Ladder Diagram de l'IEC61131-3).

REMARQUE

Il est donc important de retenir qu'une fonction mémoire peut être obtenue sur tous les constituants d'une chaîne fonctionnelle, et pas seulement sur la partie commande.

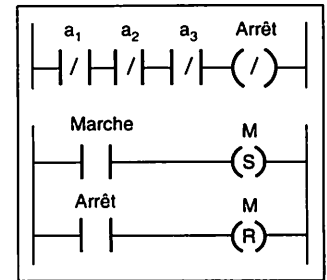


Figure 6.24. Commande de mémoire d'automate (langage LD).

2. Fronts, événements et durées

a) Changement de valeur d'une proposition logique. Front. Événement

Jusqu'à présent, seule a été considérée la valeur, vraie ou fausse, d'une proposition logique. Des propositions logiques comme *appui sur marche*, *le moteur tourne*, il est possible de dériver d'autres propositions logiques comme « l'appui sur marche commence », « le moteur se met à tourner », « le moteur s'arrête », qui présentent la propriété de n'être vraies qu'à une date donnée : ces propositions traduisent des événements, le début ou à la fin d'un état logique.

À toute variable d'état logique a peuvent être donc associées deux variables logiques : l'une traduisant son « front montant », notée $\uparrow a$, et l'autre son front descendant, $\downarrow a$.

Une variable logique d'état (ou variable « à niveau ») est vraie sur une durée, une variable type front n'est, elle, vraie qu'à certaines dates. Plus généralement, on appelle événement relatif à une variable logique tout changement de valeur de la variable.

EXEMPLE

La figure 6.25 présente les chronogrammes des variables fronts, $\uparrow a$ et $\downarrow a$, de la variable à niveau a. Quatre événements sont représentés : deux apparitions de $\uparrow a$, aux dates t_1 et t_3 , et deux apparitions de $\downarrow a$.

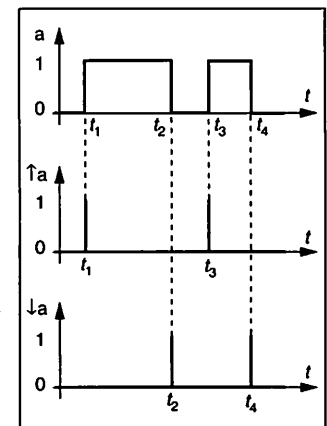


Figure 6.25. Fronts montant et descendant d'une variable logique a.

b) Mesure du temps dans un système logique

La mesure du temps dans un système logique, c'est-à-dire la mesure des durées entre événements successifs, peut aussi s'exprimer par une proposition logique de type « il s'est écoulé ΔT depuis l'événement ».

Pour représenter ces propositions sur le temps, il a été retenu une variable logique particulière, de notation $\Delta T/a$, représentation condensée de la proposition « il s'est écoulé ΔT depuis le front montant de la variable logique a ($\uparrow a$) » ($\overline{\Delta T/a}$ est la variable complémentaire « il ne s'est pas encore écoulé $\overline{\Delta T}$ depuis $\uparrow a$ »).

EXEMPLE

La figure 6.26 présente les chronogrammes des variables logiques de mesure de temps : $5s/a$, $9s/\bar{a}$ et $2s/\bar{a}$.

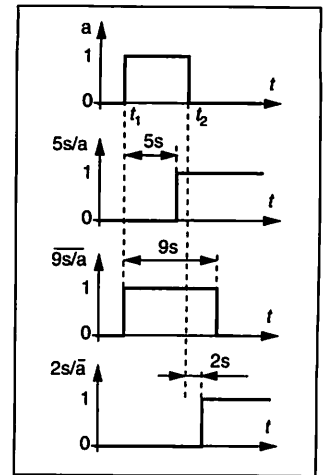


Figure 6.26. Variables logiques associées à la mesure du temps.

3. fonctions retard, temporisation, monostable

a) Fonction retard(s)

Retarder un signal logique d'entrée e , c'est décaler ses changements d'états dans le temps (Figure 6.27) :

- d'une durée T_1 à la montée (par rapport à $\uparrow a$);
- et/ou d'une durée T_2 à la descente (par rapport à $\downarrow a$).

Une fonction retard est symbolisée par un bloc-fonction, à l'intérieur duquel les deux retards sont marqués sur l'axe des temps (Figure 6.27). La fiche outil 5 présente la symbolisation des différentes variantes de retards, selon les combinaisons de T_1 et T_2 . Une telle fonction peut correspondre à un besoin fonctionnel, mais elle est aussi très utilisée pour représenter les temps de réponse des constituants (capteur, réactionneur). L'expression de la sortie est : $S = T_1/e/T_2$.

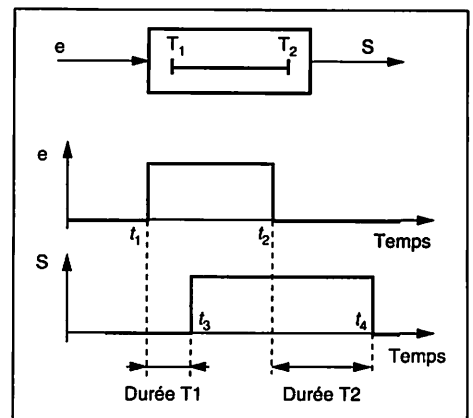


Figure 6.27. Fonction(s) retard(s).

EXEMPLE

Une variante possible du cahier des charges du fonctionnement de la trémie de la sole vibrante de la machine de conditionnement (voir FICHE SYSTÈME 2) peut être le suivant :

- la marche de la trémie dès que la sole n'est plus surchargée ($S_1 = 0$) et dès que la rampe n'est plus saturée ($S_2 = 1$);
- l'arrêt de la trémie si la sole est surchargée ($S_1 = 1$) depuis plus de trois secondes ou si la rampe est saturée ($S_2 = 0$) depuis plus de cinq secondes.

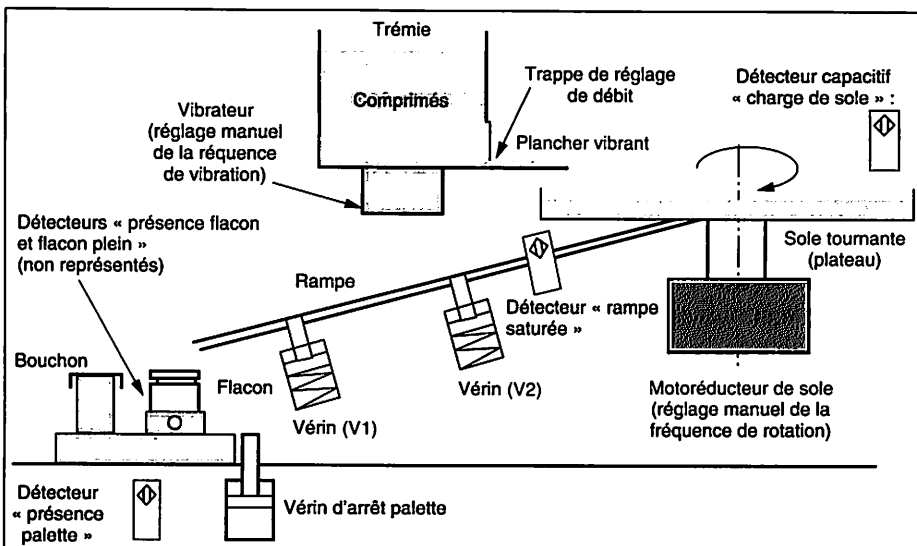


Figure 6.28. Poste de remplissage de la machine de conditionnement.

La figure 6.29a présente les chronogrammes des signaux S_1 et S_2 ainsi que la commande d'arrêt qui en résulte, se décomposant en deux signaux élémentaires y_1 (arrêt retardé de surcharge) et y_2 (arrêt retardé de saturation) :

- $y_1 = 1$ trois secondes après $\uparrow S_1$, et tant que $S_1 = 1$: il correspond à une fonction retard avec $t_1 = 3$ s et $t_2 = 0$ (Figure 6.29c) ;
- y_2 est élaboré à partir de $\overline{S_2}$ (Figure 6.29b) : il correspond à une fonction retard avec $t_1 = 5$ s et $t_2 = 0$ (Figure 6.29c) ;
- la commande marche trémie, complément de la commande d'arrêt (Figure 6.29b), correspond à une fonction NON-OU à partir des arrêts retardés y_1 et y_2 (Figure 6.29c).

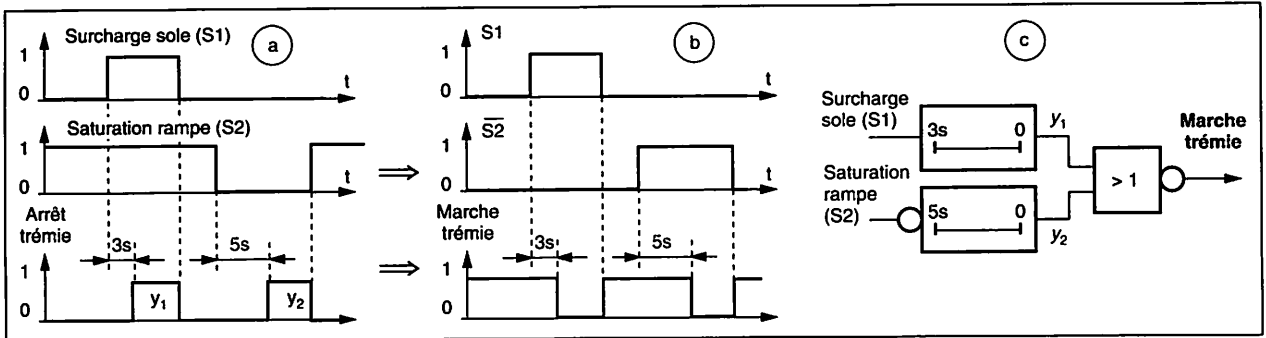


Figure 6.29. Élaboration de la consigne de marche trémie à partir de fonctions de retard.

● **Temporisation**

Cas particulier de fonction retard, une réalisation courante de temporisation est donnée figure 6.30 : elle correspond à une variable $\Delta T/a$ tant que $e = 1$, soit $s = (\Delta T/e) \cdot e$.

b) **Génération d'impulsions. Monostables**

Il est fréquemment nécessaire d'émettre un signal de sortie de durée calibrée ΔT , à l'apparition d'un événement ($\uparrow e$) pour, par exemple : piloter un distributeur, incrémenter un compteur. La fonction correspondante est appelée générateur d'impulsions ou monostable. La durée T (Figure 6.31) peut être supérieure ou non à la durée de maintien du signal d'entrée e .

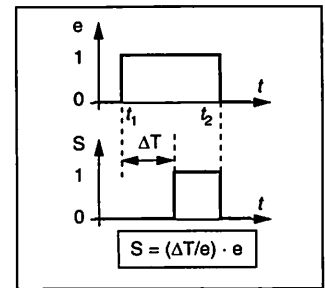


Figure 6.30. Temporisation d'une durée ΔT .

4. Fonction comptage (et/ou décomptage)

Le comptage/décomptage est l'une des fonctions les plus rencontrées dans les automatismes, pour le traitement ou pour l'acquisition des données. La fonction comptage permet de convertir une suite d'informations logiques en une information soit logique (par exemple, l'entrée e est apparue trois fois), soit numérique (le nombre d'apparitions de e).

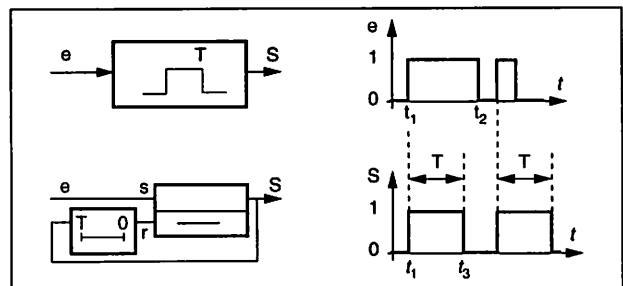


Figure 6.31. Générateur d'impulsions.

EXEMPLE

(voir FICHES SYSTÈMES 1 ET 2) :

- comptage des couches pour le palettiseur (information résultante logique) ;
- comptage du nombre de comprimés (information résultante logique) ;
- comptage du nombre d'incrément de position du capteur incrémental de l'axe z du palettiseur.

De multiples techniques de comptage existent, à partir de deux grands principes :

- le comptage « matériel », utilisant un composant appelé compteur ;
- le comptage « logiciel », par addition numérique ($n : = n + 1$).

Résumé

- Un système est séquentiel si et seulement si :
 - toutes ses entrées et toutes ses sorties sont logiques ;
 - la valeur logique de chaque sortie, à une date donnée, dépend non seulement des valeurs logiques des entrées à cette même date, mais également des valeurs antérieures.
- Au caractère séquentiel d'un système logique ont été associés les aspects suivants :
 - un effet reste maintenu alors que la cause qui l'a produit a disparu. Un tel caractère est associé à la fonction mémoire : voir chronogramme de principe, Figure 6.32a ;
 - une sortie dépend non seulement de la valeur des entrées, mais aussi de la variable temps. Une illustration fréquente correspond à un décalage temporel d'une sortie par rapport à une entrée : c'est la fonction retard, ou temporisation, dont un principe est donné, Figure 6.32b.
- Deux autres fonctions séquentielles jouent un rôle particulier en automatisme :
 - la fonction monostable, ou générateur d'impulsions (Figure 6.32c) : le signal de sortie est émis pendant une durée ΔT mesurée à partir du front montant de l'entrée e ;
 - la fonction comptage : une sortie prend un état logique déterminé lorsque qu'une séquence de valeurs d'une entrée a été obtenue. Par exemple, Figure 6.32d, la sortie S devient vraie quand l'entrée e est devenue vraie trois fois de suite (comptage par 3).

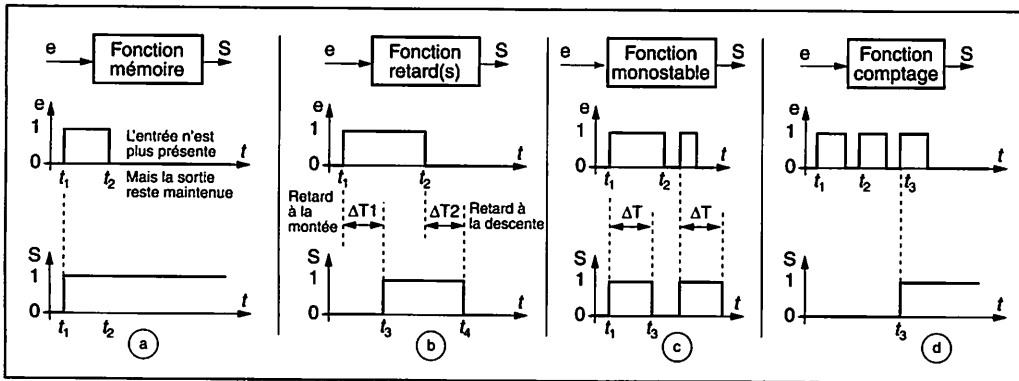


Figure 6.32. Principales fonctions séquentielles (et chronogrammes de principe).

Exercices

EXERCICE 1. Construire les chronogrammes de bpt, X1, X2 et L dans le cas de l'exemple du paragraphe I.1.

EXERCICE 2. Donner des représentations technologiques des fonctions suivantes :

$$KM1 = (M1 + KM1) \cdot \overline{KM2} \cdot \overline{au} \cdot \overline{s1}$$

$$KM2 = (M2 + KM2) \cdot \overline{KM1} \cdot \overline{au} \cdot \overline{s1}$$

EXERCICE 3. Construire une bascule r-s à partir de deux NOR (NON OU).

EXERCICE 4. Construire et comparer les chronogrammes des variables logiques : (5s/a) et (5s/a).a, issues de la variable logique a.

EXERCICE 5. Reprendre l'exemple de la sole des figures 6.28 et 6.29 avec les règles suivantes : marche de la trémie dès que la sole n'est plus surchargée ($S1 = 0$) et dès que la rampe n'est plus saturée ($S2 = 0$) ; arrêt de la trémie si la sole est surchargée ($S1 = 1$) depuis plus de trois secondes ou si la rampe est saturée ($S2 = 1$) depuis plus de cinq secondes.

Établir le chronogramme correspondant, puis le schéma logique.

7

Le GRAFCET

I. Bref historique

Le Grafcet est né en 1977 des travaux de l' AFCET (Association française pour la cybernétique économique et technique), en tant que synthèse théorique des différents outils existants à cette époque (organigramme, organiphase, diagramme de Girard, réseaux de Petri, etc.). Il fut mis sous sa forme graphique actuelle par l' ADEPA (Agence pour le développement de la productique appliqué à l' industrie) en 1979. Il s' agissait de disposer d' un langage graphique de spécification du comportement des systèmes logiques séquentiels indépendant de la technologie de commande, ergonomique, simple à utiliser, et pouvant servir de support de communication (concepteurs, réalisateurs, agents de maintenance, etc.).

Le GRAFCET a alors été normalisé au plan national, puis international.

La dernière modification de la norme CEI 60848 (février 2002) a permis d' enrichir l' outil GRAFCET afin de faciliter une description structurée et hiérarchisée du comportement des systèmes séquentiels. Certains aspects de la première édition nécessitaient également d' être précisés pour aboutir à la définition d' un véritable langage de spécification.

II. Définition et éléments de base du langage GRAFCET

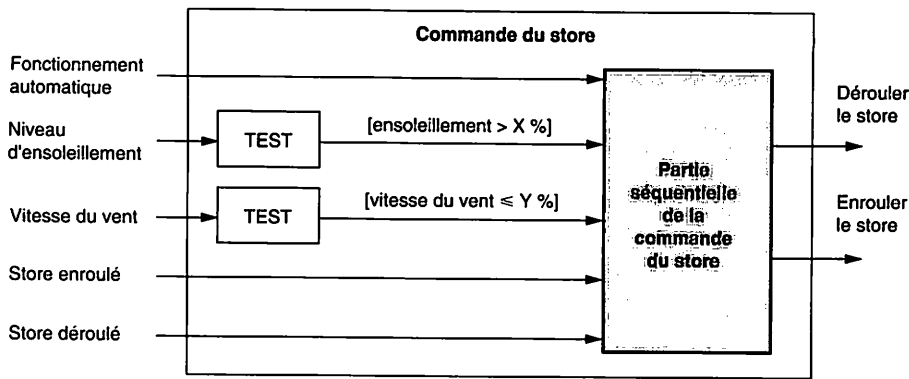
Le GRAFCET est un langage de spécification qui permet la description fonctionnelle du comportement de la partie séquentielle des commandes des systèmes automatiques.

La partie séquentielle de la commande d' un système automatique (chapitre 6) est caractérisée par ses entrées et ses sorties logiques (Figure 7.1) et son comportement. Le comportement indique la manière dont les variables de sortie dépendent des évolutions des variables d' entrée. Le GRAFCET a pour objet de spécifier ce comportement.

Dans le cas de la commande du store (Figure 7.1), toutes les entrées et les sorties ne sont pas logiques. En effet, les entrées « niveau d' ensoleillement » et « vitesse du vent » sont non booléennes. Un opérateur « test » permet de réaliser la comparaison de la valeur de chacune de ces variables d' entrées avec une valeur donnée (respectivement : $X \%$ et $Y \%$). Les résultats de chacune de ces comparaisons est une variable logique (comparaison vraie ou fausse) : $[\text{ensoleillement} > X \%]$, $[\text{vitesse du vent} \leq Y \%]$.

REMARQUE

bien que le langage GRAFCET ait servi de base au langage SFC de la norme CEI 61131-3 relative aux langages de programmation des automates programmables industriels (CHAPITRE 8), la syntaxe et la sémantique définies par chacune des deux normes sont néanmoins distinctes en raison de leurs différents domaines d' application (spécification fonctionnelle indépendante de la technologie d' une part, et spécification de réalisation programmée d' autre part).



REMARQUE

La partie séquentielle du système est caractérisée par ses variables d'entrée, ses variables de sortie et son comportement. Cette partie séquentielle ne comporte que des variables d'entrées et de sorties booléennes, toutefois le langage de spécification GRAFCET permet par extension (exemple : évaluation d'un prédicat ou affectation d'une valeur numérique à une variable) de décrire le comportement de variables non booléennes.

Figure 7.1. Représentation graphique de la partie séquentielle de la commande du store.

La norme CEI 60848 définit les symboles et les règles nécessaires à la représentation graphique de ce langage, ainsi que l'interprétation qui en est faite.

Le modèle GRAFCET est défini par un ensemble constitué :

- d'éléments graphiques de base : les étapes, les transitions et les liaisons orientées, formant l'ossature graphique du grafcet et sa structure;
- d'une interprétation, traduisant les comportements de la partie commande vis-à-vis de ses entrées/sorties et caractérisée par les actions associées aux étapes et les réceptivités associées aux transitions;
- de cinq règles d'évolution, définissant formellement le comportement dynamique.

Le langage de spécification GRAFCET permet d'établir une représentation (appelée grafcet) exprimant le comportement attendu de la partie séquentielle d'un système déterminé. Ce langage se caractérise principalement par ses éléments graphiques qui offre une représentation synthétique du comportement reposant sur une description indirecte de la situation du système (grafcet de commande du store, Figure 7.2).

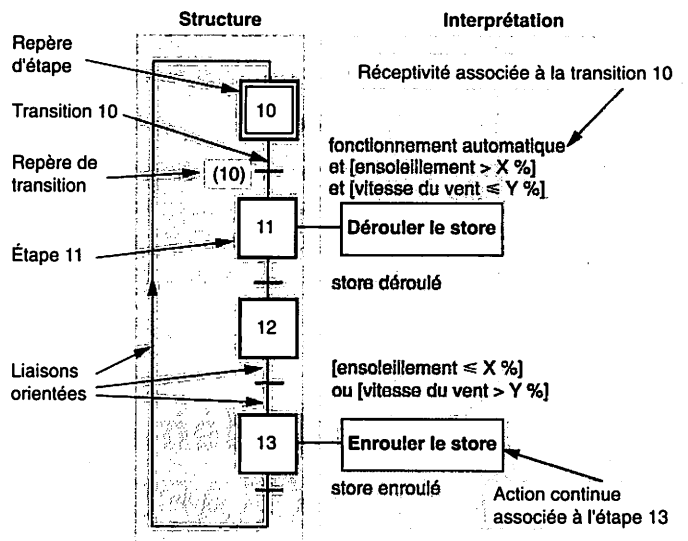


Figure 7.2. Éléments de structure et d'interprétation utilisés dans le grafcet de commande du store pour décrire le comportement de la partie séquentielle définie par ses variables d'entrée et de sortie.

a) La structure d'un grafcet est constituée des éléments de base suivants :

● **Étape**

Une étape représente un comportement qui correspond :

- soit à des opérations, ou des actions logiques, à effectuer sur le système. Par exemple : enrouler le store, dérouler le store, compter le nombre de flacons fabriqués, etc. ;
- soit à l'attente d'une autorisation d'évolution (étapes 12 et 10, Figure 7.2).

L'étape est symbolisée par un carré et repérée par un identificateur (en général une étiquette alphanumérique). L'étape initiale (voir règles d'évolution) est symbolisée par un double carré (voir étape 10 du grafcet, Figure 7.2).

Une étape est soit active, soit inactive, l'ensemble des étapes actives d'un grafcet à un instant donné représente la situation de ce grafcet à l'instant considéré.

● **Transition**

Une transition indique la possibilité d'évolution d'activité entre deux ou plusieurs étapes. Cette évolution s'accomplit par le franchissement de la transition. Une transition représente une possibilité de changement de comportement (la transition repérée (10) représente une possibilité de passage de l'étape 10 vers l'étape 11). Le passage d'un comportement au suivant correspond au franchissement de la transition. La proposition logique qui conditionne la transition est appelée la réceptivité. Une transition possible entre étapes est indiquée par un ensemble de traits comportant :

- un trait fort sur la liaison : la réceptivité est écrite à droite de ce trait ;
- dans les cas où la transition s'effectue entre plusieurs étapes, deux traits parallèles auxquels sont reliés les liaisons entre les étapes (paragraphe IV.4).

Chaque transition peut être repérée par un identificateur. Par exemple (10) est la transition entre l'étape 10 et l'étape 11 (Figure 7.2). Une transition représente une, et une seule, possibilité d'évolution; à deux possibilités d'évolution distinctes correspondent deux transitions distinctes, même si la réceptivité est identique.

● **Liaison orientée**

Une liaison orientée relie soit une ou plusieurs étapes à une transition, soit une transition à une ou plusieurs étapes.

b) L'interprétation, qui donne du sens vis-à-vis de l'application, se fait grâce aux éléments suivants :

● **Réceptivité**

Associée à chaque transition, la réceptivité exprime le résultat d'une expression booléenne. Une réceptivité qui est toujours vraie se note : 1.

La réceptivité associée à la transition (10) est la variable logique résultat de l'expression booléenne : Fonctionnement automatique et [ensoleillement > X %] et [vitesse du vent ≤ Y %].

● **Action**

L'action indique, dans un rectangle, comment agir sur la variable de sortie, soit par assignation (action continue telle que : Enrouler le store), soit par affectation (action mémorisée telle que : Incrémenter le nombre de flacons fabriqués). Plusieurs actions peuvent être associées à une même étape.

REMARQUE

Une séquence est un bloc d'étapes successives où chaque étape est suivie d'une seule transition et chaque transition n'est validée que par une seule étape (FICHE MÉTHODE 3).

III. Règles d'évolution

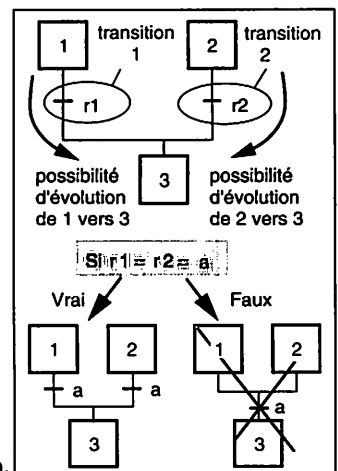
1. Règles de syntaxe

L'alternance étape-transition et transition-étape doit toujours être respectée quelle que soit la séquence parcourue.

En conséquences :

- deux étapes ou deux transitions ne doivent jamais être reliées par une liaison orientée;
- la liaison orientée relie obligatoirement une étape à une transition ou une transition à une étape.

Figure 7.3. Une transition représente une seule possibilité d'évolution.



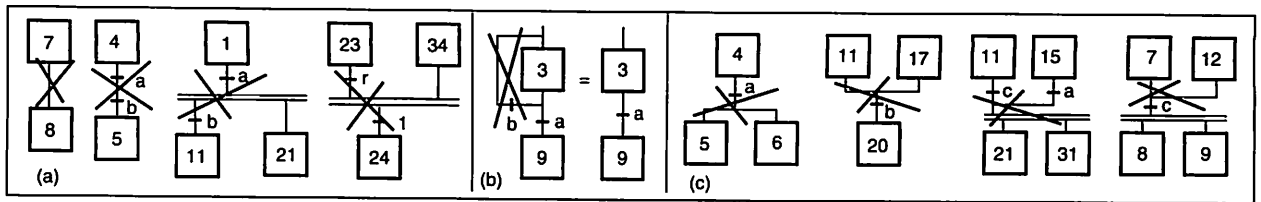


Figure 7.4. Quelques erreurs de syntaxe à éviter.

2. Règles d'évolution

Dans un grafcet, plusieurs étapes peuvent être actives simultanément, la situation étant alors caractérisée par l'ensemble des étapes actives à l'instant considéré. Les conditions d'évolution d'un ensemble d'étapes vers un autre sont alors portées par une ou plusieurs transitions, caractérisées chacune par :

- ses étapes amont;
- ses étapes aval;
- sa réceptivité associée.

Sachant que toute situation est caractérisée par l'ensemble des étapes actives à l'instant considéré, les règles d'évolution du GRAFCET ne sont que l'application, sur les étapes, du principe d'évolution entre les situations de la partie séquentielle du système.

a) Situation initiale

La situation initiale est la situation à l'instant initial, elle est donc décrite par l'ensemble des étapes actives à cet instant. Le choix de la situation à l'instant initial repose sur des considérations méthodologiques et relatives à la nature de la partie séquentielle du système visé.

RÈGLE 1

La situation initiale, choisie par le concepteur, est la situation à l'instant initial.

La situation initiale de la commande du store est identifiée dans le grafcet grâce à l'étape initiale 10 (double carré, Figure 7.5). Dans ce cas, l'ensemble des étapes initiales constituant la situation initiale est : {10}.

b) Franchissement d'une transition

RÈGLE 2

Une transition est dite validée lorsque toutes les étapes immédiatement précédentes reliées à cette transition sont actives. Le franchissement d'une transition se produit :

- lorsque la transition est validée,
- et que la réceptivité associée à cette transition est vraie.

L'étape 11 n'est pas active (Figure 7.6 à gauche), la transition (11) n'est pas validée quelque soit la valeur (vraie ou fausse) de la réceptivité « store déroulé ».

L'étape 11 est active (Figure 7.6 à droite), alors la transition (11) est validée.

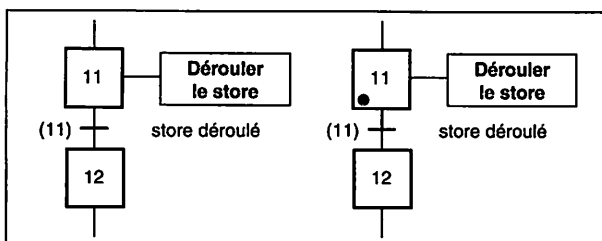


Figure 7.6. Illustration de la règle 2.

REMARQUE

L'expression de la règle 1 a été modifiée par rapport à la première édition de la norme. En effet, cette première édition était exprimée pour des systèmes automatisés industriels de production où il était aisé de distinguer partie commande et partie opérative. La norme élargit maintenant le champ d'application à tout type de système à la seule condition d'être un système logique séquentiel.

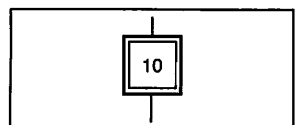


Figure 7.5. Étape initiale du grafcet de commande du store.

REMARQUE

Pour identifier une étape active à un instant donné on la « marque » par une puce (ou jeton).

c) Évolution des étapes actives

RÈGLE 3

Le franchissement d'une transition entraîne simultanément l'activation de toutes les étapes immédiatement suivantes et la désactivation de toutes les étapes immédiatement précédentes.

L'étape 11 est active et la réceptivité «store déroulé» est fausse, alors la transition (11) est validée (Figure 7.7 à gauche).

L'étape 11 est active, alors la transition (11) est franchissable et automatiquement franchie (Figure 7.7 au centre).

Le franchissement de la transition (11) provoque simultanément l'activation de l'étape 12 et la désactivation de l'étape 11 (Figure 7.7 à droite).

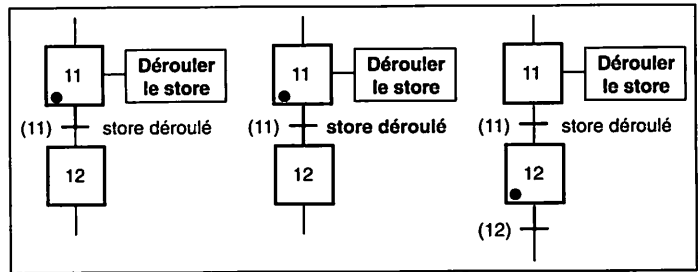


Figure 7.7. Illustration de la règle 3.

d) Évolutions simultanées

L'évolution entre deux situations actives implique qu'aucune situation intermédiaire ne soit possible, on passe donc instantanément d'une représentation de la situation par un ensemble d'étapes à une autre représentation.

RÈGLE 4

Plusieurs transitions simultanément franchissables sont simultanément franchies.

Dans l'exemple de la figure 7.8, les étapes 41 et 55 sont actives, les transitions (55) et (41) sont donc validées. Lorsque la réceptivité «a» devient vraie, les deux transitions (41) et (55) sont franchies simultanément. Simultanément ces franchissements provoquent l'activation des étapes 42 et 56 et la désactivation des étapes 41 et 55.

REMARQUE

Une autre règle (règle 5) est définie dans la norme, elle permet de préciser ce qui se passe dans le cas où une même étape doit être activée et désactivée simultanément. L'utilisation de cette règle sort du cadre de cet ouvrage.

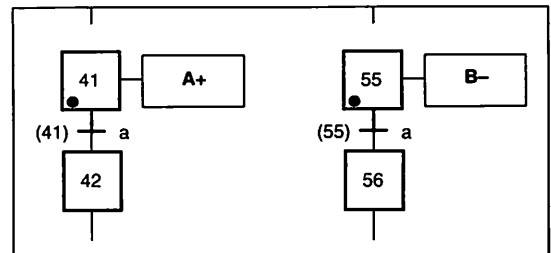


Figure 7.8. Illustration de la règle 4.

3. Logique associée aux éléments de base du GRAFCET

Des propriétés logiques sont associées à l'état d'un grafcet à chaque instant.

● Une étape est soit active, soit inactive (à un instant donné)

Par l'ensemble de ses étapes, le grafcet représente l'ensemble des comportements possibles du système. Cependant, à un instant donné, seuls les comportements correspondant aux étapes actives sont obtenus. Pour visualiser les seules étapes actives, on convient de les marquer par un point, par de la couleur, par une inversion vidéo, etc.

On associe à chaque étape repérée *i* une variable logique notée X_i , avec $X_i = 1$ lorsque l'étape est active, ou $X_i = 0$ lorsque l'étape est inactive.

Figure 7.8 : les étapes 41 et 55 sont actives, $X_{41} = 1$ et $X_{55} = 1$; les autres étapes sont inactives, $X_{42} = X_{56} = 0$.

On appelle **situation** d'un grafcet à une date donnée, l'ensemble des étapes actives à cette date. Une situation s'indique par la liste, entre parenthèses (ou accolades), des étapes actives. Par exemple : Figure 7.8, la situation est notée (41, 55) ou {41, 55}.

Une situation donnée est établie ou non établie à une date donnée. Elle est établie à une date donnée si toutes les étapes à partir desquelles la transition peut s'effectuer (les étapes immédiatement précédentes) sont actives.

● Une transition est soit validée, soit non validée

Une transition est validée lorsque toutes les étapes à partir desquelles la transition peut s'effectuer (les étapes immédiatement précédentes) sont actives.

● Une réceptivité est soit vraie, soit fausse

Une réceptivité associée à une transition est une expression logique. Le résultat de cette expression logique est logique.

● Représentation temporelle d'une évolution de situation

À la date initiale d_0 , l'étape 11 est active, la réceptivité «Store déroulé» fausse (Figure 7.9). Dès que «Store déroulé» devient vraie (date d_1), la règle 3 s'applique et simultanément, l'étape 11 est désactivée ($X_{11} = 0$), et l'étape 12 est activée ($X_{12} = 1$).

À la date d_1 , le grafcet a évolué de la situation {11} vers la situation {12}.

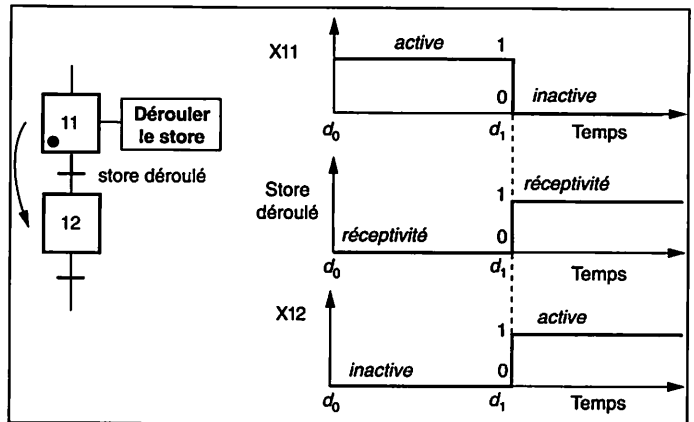


Figure 7.9. Exemple 1 de représentation d'évolution de situation.

À la date initiale d_0 , les étapes 41 et 55 sont considérées actives, la réceptivité «a» fausse (Figure 7.10).

Dès que «a» devient vraie (date d_1), les règles 3 et 4 s'appliquent et simultanément, les étapes 41 et 55 sont désactivées ($X_{41} = X_{55} = 0$), et les étapes 42 et 56 sont activées ($X_{42} = X_{56} = 1$).

À la date d_1 , le grafcet a évolué de la situation {41, 55} vers la situation {42, 56}.

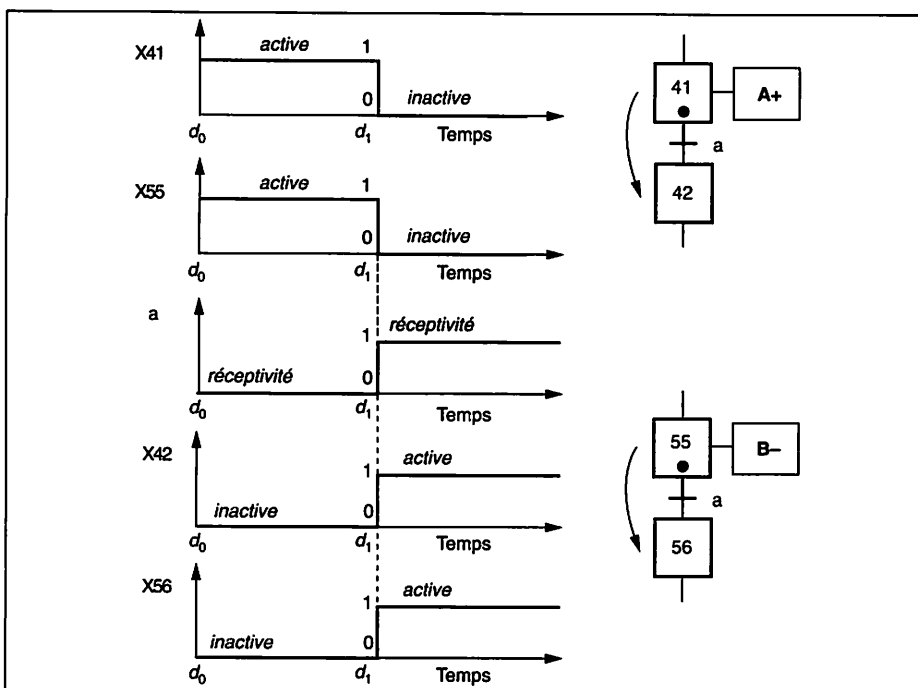


Figure 7.10. Exemple 2 de représentation d'évolution de situation.

IV. Les structures de base des grafjets

Les structures de base sont des configurations de grafjet associées aux concepts de base des systèmes logiques. Elles permettent d'exprimer des successions d'états (paragraphe I), des choix entre plusieurs séquences et des parallélismes de séquences.

REMARQUE

La séquence est dite « active » si au moins une de ses étapes est active, elle est dite « inactive » lorsque aucune de ses étapes n'est active. Le nombre d'étapes formant une séquence est aussi grand que l'on veut.

1. Séquence

On appelle séquence une succession d'étapes telles que :

- chaque étape, exceptée la dernière, ne possède qu'une seule transition aval ;
- chaque étape, exceptée la première, ne possède qu'une seule transition amont validée par une seule étape de la séquence.

Dans l'exemple, Figure 7.11, dans la séquence de quatre étapes, chaque étape est suivie d'une seule transition et chaque transition n'est précédée que par une seule étape. Cette séquence est rebouclée sur elle-même, c'est un cycle d'une seule séquence.

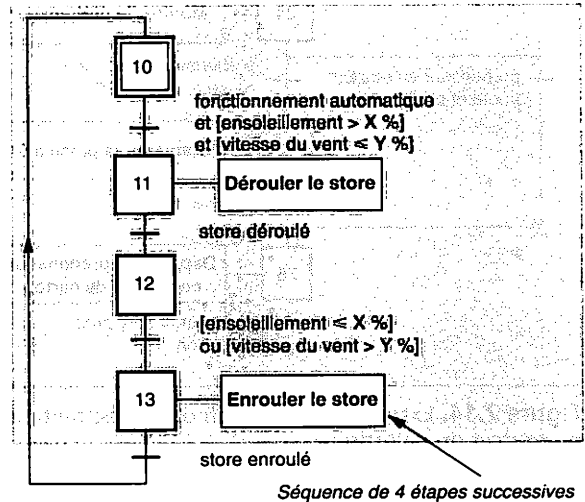


Figure 7.11. Exemple de séquence.

2. Sélection de séquence

La sélection de séquences (Figure 7.12) exprime un choix d'évolution entre plusieurs séquences à partir d'une ou de plusieurs étapes. Cette structure se représente par autant de transitions validées simultanément qu'il peut y avoir d'évolutions possibles.

L'exclusion entre les séquences n'est pas structurelle. Pour l'obtenir, le spécificateur doit s'assurer soit de l'incompatibilité mécanique (sélecteur à deux positions par exemple) ou temporelle des réceptivités (plus rarement utilisé), soit de leur exclusion logique. Il est possible également de donner une priorité à une séquence particulière.

Dans l'exemple, Figure 7.13, si les deux réceptivités a et b sont vraies à l'activation de l'étape 1 (deux boutons poussoirs actionnés avant l'activation de l'étape 1, panne d'un capteur, etc.), les deux transitions (2) et (3) validées par l'étape 1 sont franchissables et donc franchies simultanément (règle 4). Les deux séquences débutant par les étapes 2 et 3 sont alors activées simultanément, ce qui peut engendrer des problèmes dans le fonctionnement du système. Il est possible alors de compléter les réceptivités associées aux transitions (2) et (3) pour rendre une séquence prioritaire ou pour bloquer l'évolution du grafjet.

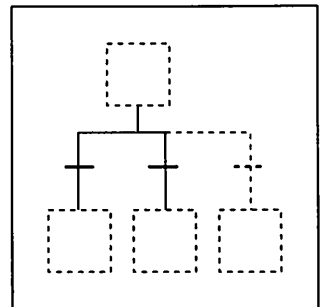


Figure 7.12. Sélection de séquences.

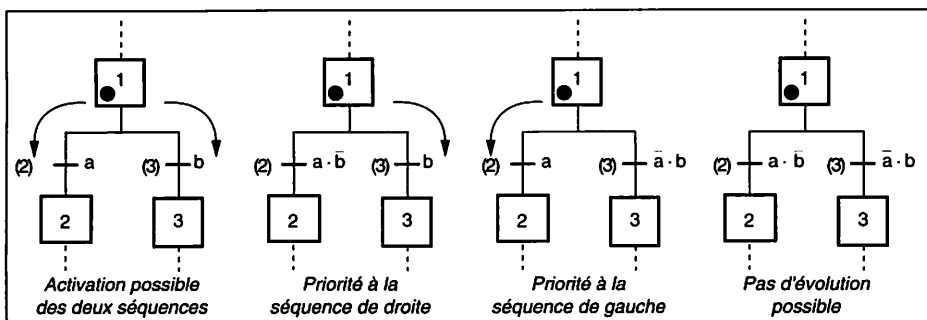


Figure 7.13. Choix entre deux séquences. Choix exclusif ou non, ou avec une priorité.

3. Reprise de séquence et saut d'étape(s)

La reprise de séquence est un cas particulier de sélection de séquences, qui permet de recommencer la même séquence jusqu'à ce que, par exemple, une condition fixée soit obtenue. La figure 7.15 illustre une reprise de la séquence comprenant les étapes 154 et 155 du grafcet GPRE du palettiseur.

REMARQUE

Il est possible pour des raisons de représentation graphique (si l'on ne peut pas l'éviter), de placer des transitions sur des segments de liaison horizontaux.

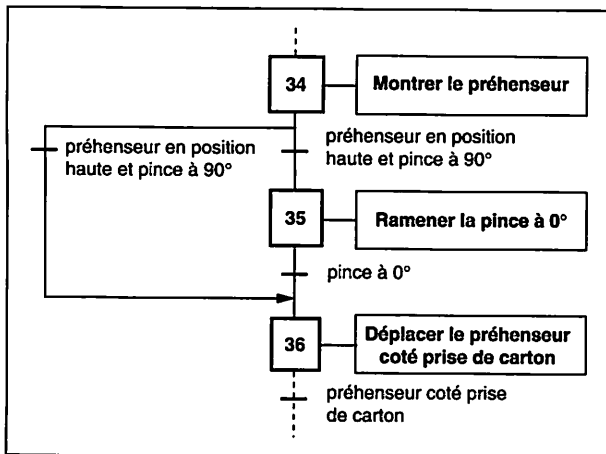


Figure 7.14. Extrait de l'expansion de la macro-étape M3 du grafcet du palettiseur.

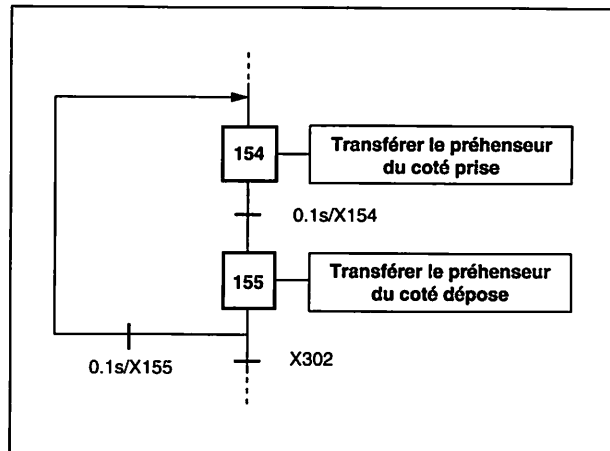


Figure 7.15. Extrait du grafcet de remise en énergie GPRE du palettiseur.

Le saut d'étape(s) est un cas particulier de sélection de séquences, qui permet soit de parcourir la séquence complète soit de sauter une ou plusieurs étapes de la séquence lorsque, par exemple, les actions associées à ces étapes deviennent inutiles. La figure 7.14 illustre un saut d'étape dans le cas de l'expansion de la macro-étape M3. En effet, si la pince est déjà en position 0° il est inutile d'activer l'étape 35 pour ramener la pince à 0°.

4. Parallélisme de séquences

a) Activation de séquences parallèles

Le symbole de synchronisation est utilisé dans cette structure pour indiquer l'activation simultanée de plusieurs séquences à partir d'une ou plusieurs étapes. Après leur activation simultanée, l'évolution des étapes actives dans chacune des séquences parallèles devient alors indépendante.

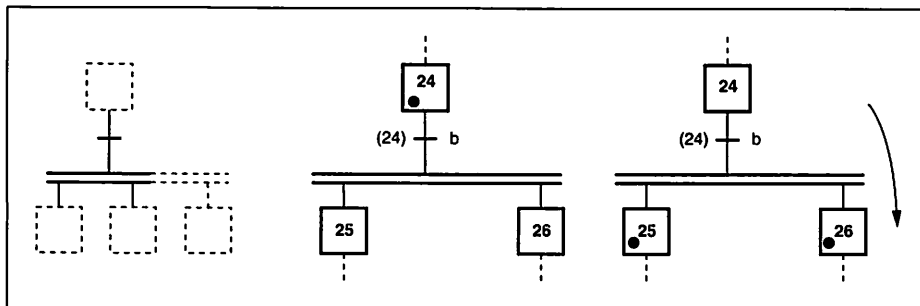


Figure 7.16. Symbole du parallélisme de séquences et illustration.

Dans l'exemple, Figure 7.16, l'étape 24, lorsqu'elle est active, valide la transition (24). Alors si la réceptivité b est vraie, la transition (24) est franchie et simultanément, les étapes 25 et 26 sont activées, et l'étape 24 est désactivée.

b) Synchronisation de séquences

Le symbole de synchronisation (Figure 7.17 à gauche) est utilisé dans cette structure pour indiquer l'attente de la fin des séquences amont avant d'activer la séquence aval. La transition n'est validée que lorsque toutes les étapes amont sont actives.

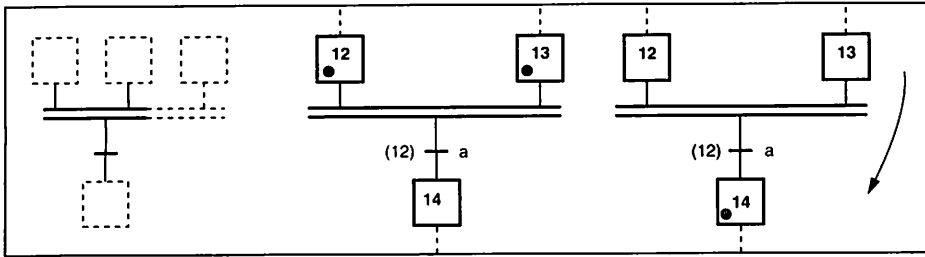


Figure 7.17. Symbole de la synchronisation de séquences et illustration.

Dans l'exemple, Figure 7.17, les étapes 12 et 13, lorsqu'elles sont actives, valident la transition (12). Alors si la réceptivité *a* est vraie, la transition (12) est franchie, et simultanément, l'étape 14 est activée, et les étapes 12 et 13 sont désactivées.

c) Synchronisation et activation de séquences parallèles

Le symbole de synchronisation est utilisé deux fois dans cette structure (Figure 7.18 à gauche) pour indiquer l'attente de la fin des séquences amont avant l'activation simultanée des séquences aval.

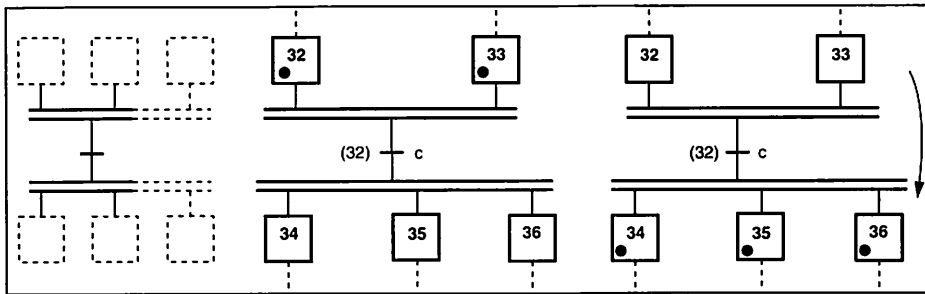


Figure 7.18. Illustration de la synchronisation et de l'activation de séquences parallèles.

Dans l'exemple, Figure 7.18, les étapes 32 et 33, lorsqu'elles sont actives, valident la transition (32). Alors si la réceptivité *c* est vraie, la transition (32) est franchie, et simultanément, les étapes 34, 35 et 36 sont activées, et les étapes 32 et 33 sont désactivées.

REMARQUE

Dans le domaine de l'automatique, on associe en général le mot « information » aux données d'entrée de la commande, c'est-à-dire au récepteur. Lorsque la commande devient l'émetteur et qu'elle adresse une information vers l'extérieur (préactionneurs, pupitre), on utilise plutôt les mots « ordre » ou « compte rendu ».

5. Macro-étape

Une macro-étape est une représentation unique d'une partie détaillée de grafcet, appelé expansion de la macro-étape. Elle est utilisée pour simplifier la représentation, pour la rendre plus lisible, ou pour insister sur certaines structures sans se perdre dans les détails. La macro-étape, identifiable par son symbole et la lettre M utilisée dans son repère (Figure 7.19), ne possède pas toutes les propriétés des autres étapes, car seule l'étape de sortie de son expansion valide ses transitions aval.

L'expansion d'une macro-étape (par exemple M2, Figure 7.19) est une partie de grafcet munie d'une étape d'entrée E2 et d'une étape de sortie S2. L'étape d'entrée devient active lorsque l'une des transitions amont de la macro-étape est franchie. La ou les transitions aval de la macro-étape ne sont validées que lorsque l'étape de sortie est active. L'expansion d'une macro-étape peut comporter elle-même une ou plusieurs macro-étapes.

Figure 7.19, le franchissement de la transition (2) provoque l'activation de l'étape d'entrée E2, la transition (3) ne sera validée que lorsque l'étape de sortie S2 sera

active. La réceptivité associée à la transition (3) étant toujours vraie (1), si elle est validée elle sera franchie, l'étape S2 sera alors désactivée. La macro-étape M3 reste active tant que l'une des étapes de son expansion est active.

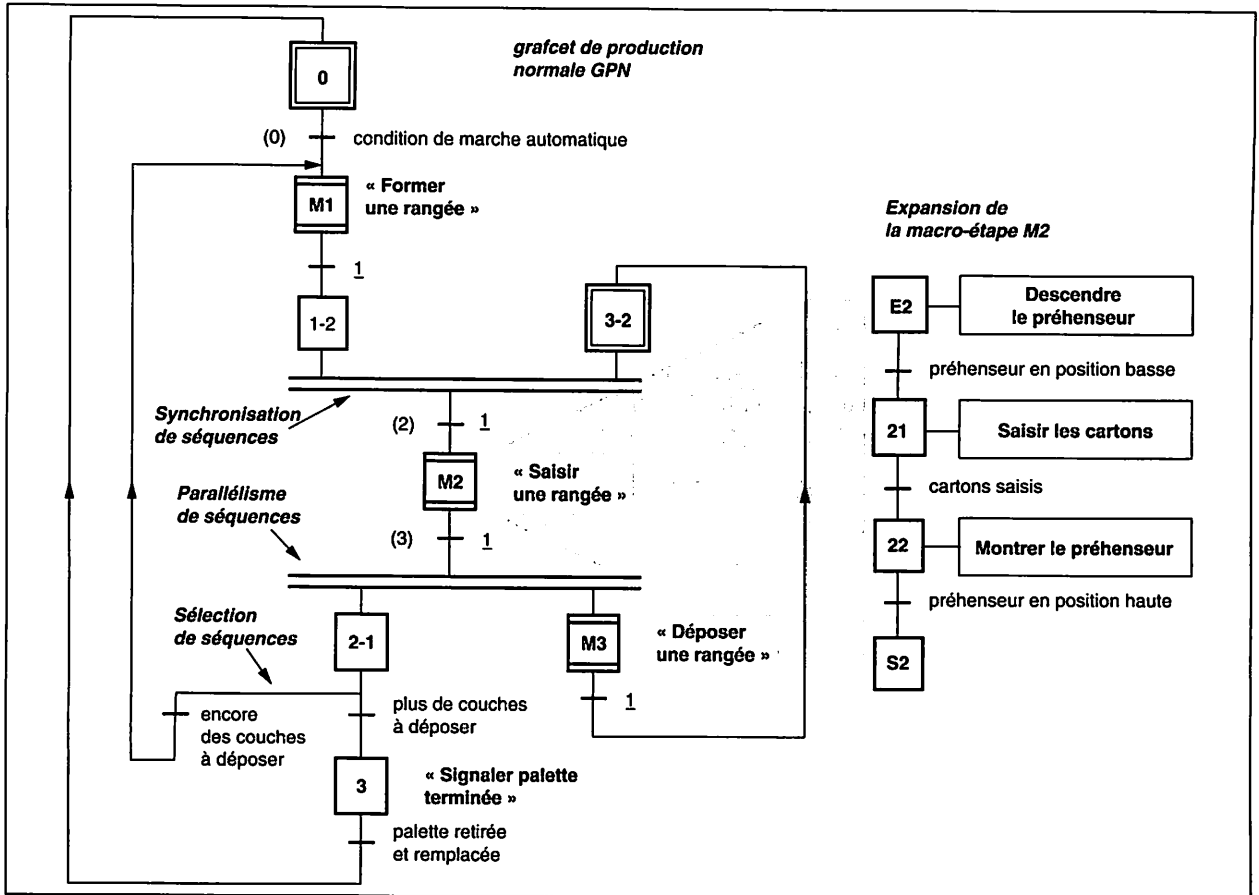


Figure 7.19. Grafcet de production normale du palettiseur et expansion de la macro-étape M2.

V. L'interprétation. Réceptivités, événements et actions

1. Réceptivité

Une réceptivité est un élément du langage GRAFCET. Associée à une transition, la réceptivité exprime le résultat d'une expression booléenne. Une réceptivité est soit vraie soit fausse.

Dans l'extrait de l'expansion de la macro-étape M3 du grafcet du palettiseur (Figure 7.20), la réceptivité associée à la transition (34) est une expression booléenne. Dans cette expression un ET logique est réalisé entre deux variables logiques :

- Préhenseur en position haute;
- Pince à 90°.

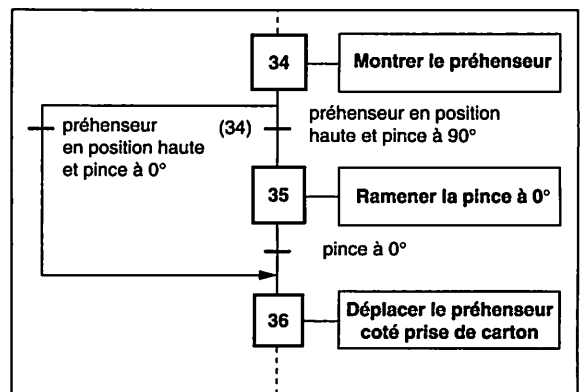


Figure 7.20. Extrait de l'expansion de la macro-étape M3 du grafcet du palettiseur.

Il faut donc que ces deux variables soient vraies pour que la réceptivité associée à la transition (34) soit vraie.

a) Réceptivité dépendante du temps

L'utilisation la plus courante est la temporisation de variable d'étape, la réceptivité devenant fautive dès la désactivation de l'étape temporisée. L'étape temporisée doit rester active pendant un temps supérieur ou égal au temps indiqué pour que la réceptivité puisse être vraie. Il est possible d'utiliser cette notation lorsque l'étape temporisée n'est pas l'étape amont de la transition.

Dans l'exemple, Figure 7.21, la transition (154) n'est franchie qu'après une durée d'activation de l'étape 154 de 0,1 s.

b) Utilisation de prédicats dans la réceptivité

Un prédicat est une expression contenant une ou plusieurs variables et qui est susceptible de devenir une proposition vraie ou fautive selon les valeurs attribuées à ces variables. Les prédicats sont souvent utilisés dans les réceptivités pour réaliser des comparaisons (niveau d'une température atteint ou non, pression inférieure à une valeur donnée,...).

La notation « [...] » signifie que la valeur booléenne du prédicat constitue la variable de réceptivité. Ainsi lorsque l'expression (assertion) contenue entre les deux crochets est vérifiée le prédicat vaut 1, dans le cas contraire il vaut 0.

La réceptivité associée à la transition (12) (Figure 7.22 à gauche) est un OU logique entre deux prédicats. La réceptivité est vraie si le niveau d'ensoleillement est inférieur ou égal à X % ou si la vitesse du vent est supérieure à Y %.

La variable booléenne de prédicat peut être associée à d'autres variables logiques pour constituer une proposition logique de réceptivité (réceptivité associée à la transition (10) (Figure 7.22 à droite).

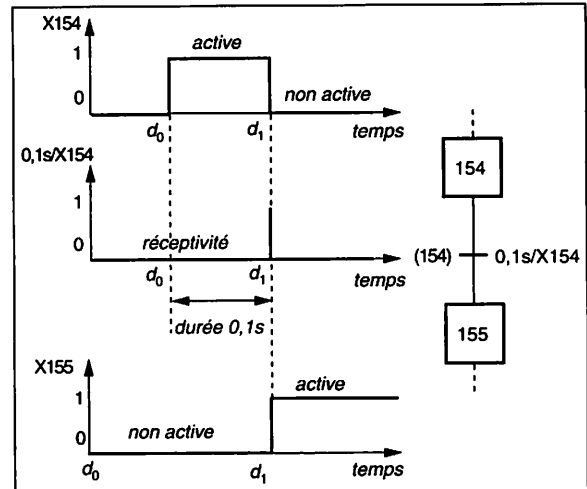


Figure 7.21. Extrait du grafcet de remise en énergie GPRE du palettiseur et chronogramme d'évolution de situation.

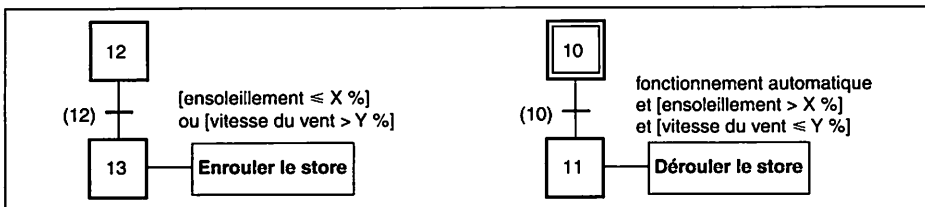


Figure 7.22. Extraits du grafcet de commande du store, exemple de réceptivités utilisant des prédicats.

2. Événements d'entrée

Un événement d'entrée est caractérisé par le changement de valeur d'une ou plusieurs variables d'entrée de la partie séquentielle du système

Les règles d'évolution montrent que seul un changement des valeurs des variables d'entrée est susceptible de provoquer l'évolution d'un grafcet. Dans la pratique, il est possible de spécifier un événement d'entrée caractérisé par le changement d'état d'une ou plusieurs variables booléennes d'entrée (front montant noté ↑, ou front descendant noté ↓).

Le front montant d'une variable logique n'est vrai qu'au passage de la valeur 0 à la valeur 1 de la variable concernée (↑a n'est vrai qu'au passage de 0 à 1 de la variable logique a). De même, ↓a n'est vrai qu'au passage de 1 à 0 de la variable a.

L'événement se produit à la date du changement d'état des variables d'entrée qui le caractérisent.

Dans le grafcet de la commande du store, lorsque l'étape 11 est active, c'est le passage de l'état 0 à l'état 1 de la réceptivité «Store enroulé» qui provoque l'évolution du grafcet vers la situation {12}.

Dans l'exemple de la figure 7.23, c'est le passage de l'état 0 à l'état 1 de l'entrée : Demande de POM, lorsque l'entrée : conditions de départ POM est vraie, qui rend la réceptivité associée à la transition (90) vraie.

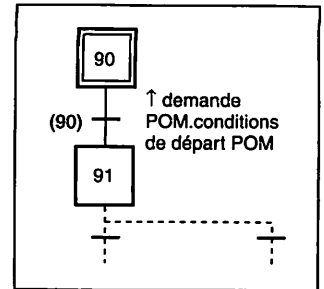


Figure 7.23. Événement d'entrée, illustration avec un extrait du grafcet de prise d'origine du palettiseur.

3. Modes de sortie

Les actions permettent d'établir le lien entre l'évolution du grafcet et les sorties. La norme propose deux modes de sortie : le mode continu et le mode mémorisé (nouvellement introduit).

a) Action continue

En *mode continu* c'est l'association d'une action à une étape qui permet d'indiquer qu'une variable de sortie a la valeur vraie si l'étape est active. On appelle **assignation le fait d'imposer la valeur (vraie ou fausse) des variables de sortie.**

Dans le grafcet de commande du store, Figure 7.24, l'action continue «Dérouler le store» est associée uniquement à l'étape 11. La variable de sortie associée, soit : «Dérouler le store», est donc assignée à la valeur vraie lorsque l'étape 11 est active. Elle est assignée à la valeur fausse lorsque l'étape 11 est inactive (donc lorsque l'une des étapes 10, 12 ou 13 est active). L'expression logique de la variable de sortie est donc : Dérouler le store = X11.

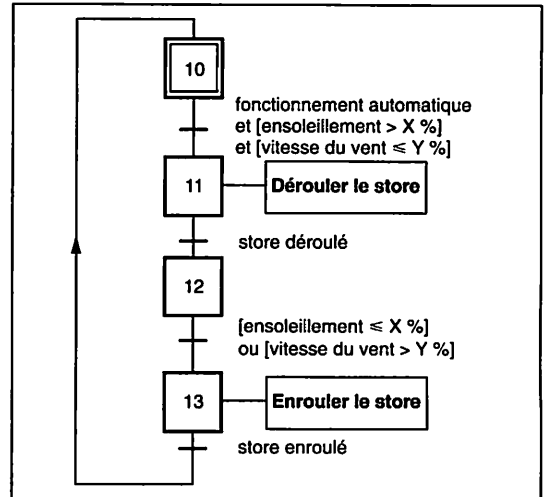


Figure 7.24. Utilisation d'actions continues dans le grafcet de commande du store.

Le même raisonnement peut être tenu pour l'action continue «Enrouler le store». Enrouler le store = X13.

Dans l'expansion de la macro-étape M1, Figure 7.25, l'association de l'action continue «Pousser le carton» est associée à deux étapes (11 et 13). Dans ce cas la valeur de la variable logique de sortie associée, soit Pousser le carton, est assignée à la valeur vraie lorsque l'étape 11 est active OU lorsque l'étape 13 est active. Cette variable est assignée à la valeur fausse dans le cas où l'étape E1, 12 ou S1 est active.

L'équation de la variable de sortie est donc :

$$\text{Pousser le carton} = X11 + X13.$$

De même, l'équation de la variable de sortie, Faire avancer le carton, est donc :

$$\text{Faire avancer le carton} = XE1 + X12.$$

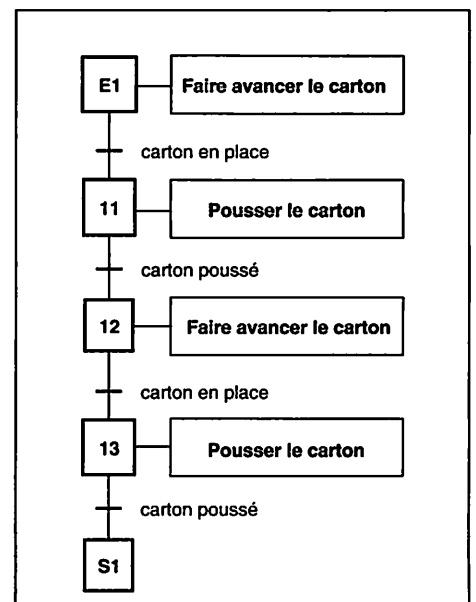


Figure 7.25. Utilisation d'une même action continue associée à plusieurs étapes.

● Action continue conditionnée

Le concepteur a la possibilité de conditionner une **action continue** (Figure 7.26), pour répondre par exemple à des contraintes de sécurité. Une proposition logique, appelée condition d'assignation, qui peut être vraie ou fausse, est alors utilisée pour conditionner toute action continue. Figure 7.26, l'action continue est « Percer », la condition d'assignation est « Foret tourne ». La variable de sortie : Percer, est assignée à la valeur vraie si l'étape 61 est active et si foret tourne. Percer = X61. (foret tourne).

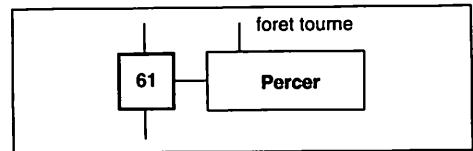


Figure 7.26. Action continue conditionnée.

REMARQUE

L'utilisation d'un opérateur retard dans l'expression de la condition d'assignation, permet d'obtenir une action continue retardée ou limitée dans le temps. La condition d'assignation ne doit jamais comporter de front de variable (front montant ou descendant), car l'action continue n'est pas mémorisée. L'assignation sur événement n'a aucun sens.

Pour une situation donnée, les valeurs des sorties relatives aux actions continues sont assignées :

- à la valeur vraie, pour chacune des sorties relatives aux actions associées aux étapes actives et pour lesquelles les conditions d'assignation sont vérifiées ;
- à la valeur fausse, pour les autres sorties (qui ne sont pas assignées à la valeur vraie).

● Action limitée dans le temps

L'utilisation de l'opérateur retard dans la réceptivité associée à une transition qui suit immédiatement une étape, permet de limiter dans le temps la durée de l'action continue associée à cette même étape. Dans la figure 7.27, l'étape 154 ne restera active que 0,1s. L'assignation de la variable de sortie associée à l'étape 154 ne sera donc vraie que pendant 0,1 s.

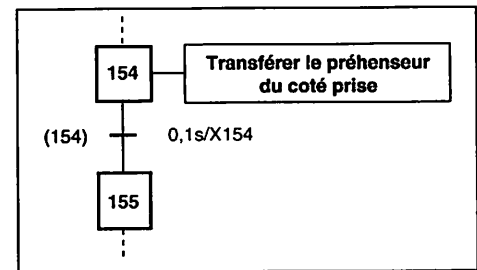


Figure 7.27. Extrait du grafcet de remise en énergie GPRE du palettiseur.

b) Action mémorisée

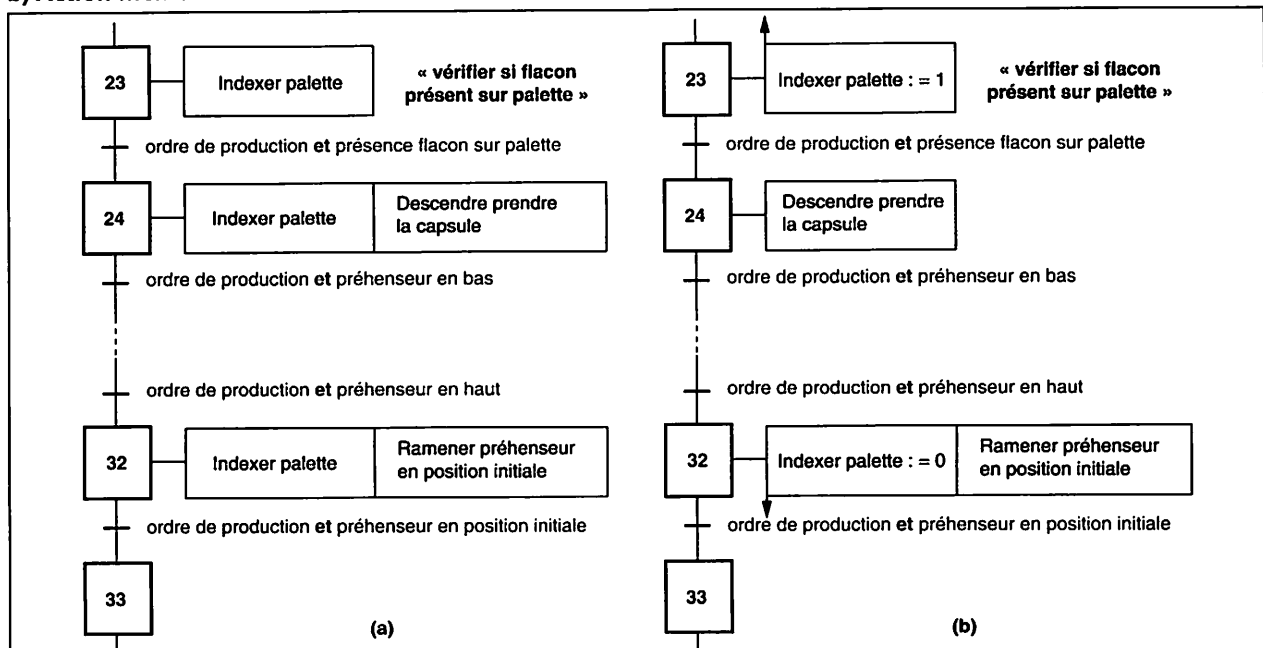


Figure 7.28. Utilisation des ordres mémorisés sur activation et désactivation d'étapes (poste de bouchage : machine de conditionnement de comprimés).

En mode *mémorisé*, c'est l'association d'une action à des événements internes (activation d'une étape, désactivation d'une étape, ..) qui permet d'indiquer qu'une variable de sortie prend **et garde** la valeur imposée si l'un de ces événements se produit. La valeur d'une variable de sortie relative à une action mémorisée reste inchangée tant qu'un nouvel événement spécifié ne la modifie pas.

On appelle *affectation* le fait de mémoriser, à un instant donné, la mise à une valeur déterminée d'une variable de sortie.

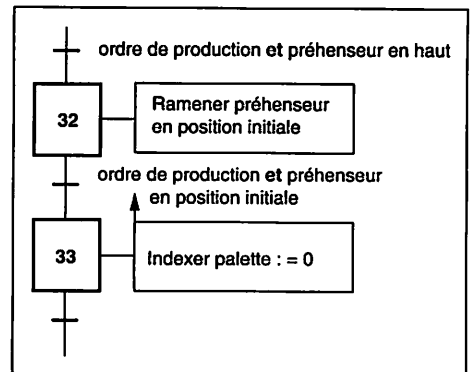
● **Affectation d'une variable booléenne**

La figure 7.28 (a), montre un extrait du grafcet du poste de bouchage (machine de conditionnement de comprimés) utilisant l'action continue pour spécifier le fonctionnement de « Indexer plateau », la figure 7.28 (b) utilise l'écriture d'une action mémorisée associée à deux événements internes : front montant de l'étape 23 ou action à l'activation pour affecter la valeur 1 (valeur vraie de la variable booléenne) et front descendant de l'étape 32 ou action à la désactivation pour affecter la valeur 0 (valeur fausse de la variable booléenne), rendant le grafcet plus lisible (deux actions au lieu de 9 pour décrire le comportement de « Indexer palette »).

Noter toutefois que ces deux représentations ne sont pas complètement équivalentes. En effet, en cas de désactivation du grafcet (par forçage notamment), « Indexer palette » s'il était activé est désactivé dans le premier cas, mais reste activé dans le second cas (sauf en cas d'ordre contraire ou ordre d'initialisation).

Une autre possibilité aurait été de mettre la variable « Indexer palette » à 0 lors de l'activation de l'étape 33 (Figure 7.29)

Figure 7.29. Autre possibilité : affectation de la variable à 0 sur activation d'une étape.



Règle d'affectation : La valeur d'une sortie, relative à une action mémorisée et associée à un événement, est affectée à la valeur indiquée si l'événement interne spécifié se produit ; à l'initialisation la valeur de cette sortie est nulle.

● **Affectation d'une variable numérique**

Cet ordre permet d'affecter une valeur à une variable (Figure 7.30).

Le libellé « $C := C + 1$ » permet de décrire l'opération d'incrémenta-tion de la variable C (affectation de C + 1 à la valeur numérique C). Cette opération est très utilisée dans la description d'opération de comptage.

Le libellé « $C := C - 1$ » permet de décrire l'opération de décrémenta-tion de la variable C (affectation de C - 1 à la valeur numérique C). Cette opération est très utilisée dans la description d'opération de décomptage.

Dans l'exemple de la figure 7.31, une action mémorisée associée à deux événements internes : front montant de l'étape 11 ou front montant de l'étape 13, est utilisée pour compter le nombre de carton C amenés dans le palettiseur. La valeur de la variable C est incrémentée d'une unité à chaque fois que l'étape 11 est activée ou à chaque fois que l'étape 13 est activée, donc à chaque fois qu'un carton a été mis en place en entrée de palettiseur. La valeur de C obtenue après incrémenta-tion est mémorisée, même après désactivation des étapes 11 ou 13. La remise à 0 de la variable C (initialisation du compteur ($C := 0$)) peut apparaître explicitement dans le grafcet du palettiseur. Si elle n'a pas été explicitée par le spécificateur, il convient d'appliquer la règle suivante qui précise que la valeur de C sera remise à 0 lors de l'initialisation.

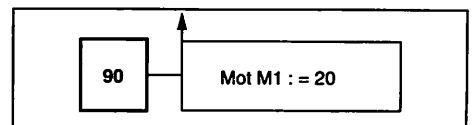


Figure 7.30. Affectation de la valeur 200 dans le mot M1, à l'activation de l'étape 190.

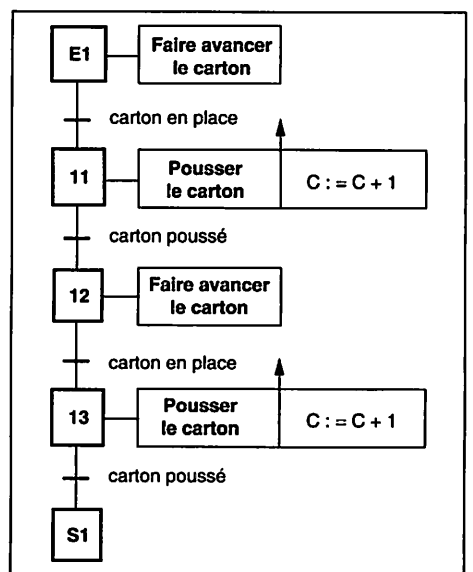


Figure 7.31. Expansion de la macro-étape M1 avec compteur de cartons.

● Action sur événement

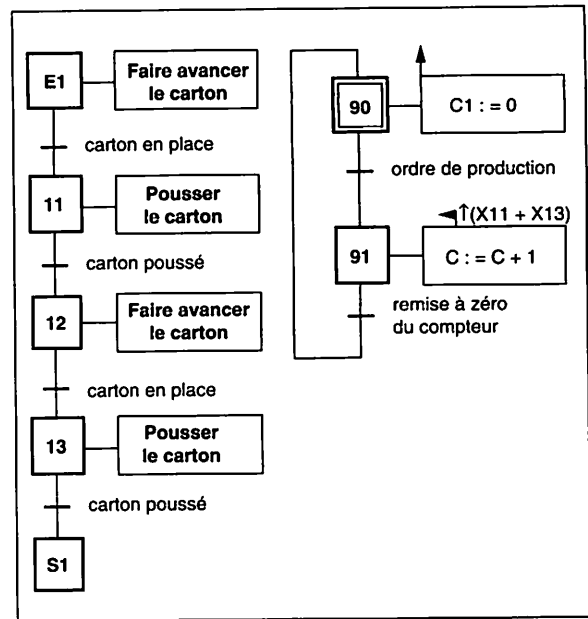
Une action sur événement est une action mémorisée conditionnée par des événements internes décrits par une expression logique, à condition que l'étape, à laquelle l'action est reliée, soit active. Il est impératif que cette expression logique comporte un ou plusieurs fronts de variables d'entrée.

Dans l'exemple, Figure 7.32, le comptage est réalisé, par un traitement séparé, lorsque la macro-étape M1 est active (donc si l'une des étapes de son expansion est active).

Il est alors inutile d'associer une action mémorisée à une ou plusieurs étapes de l'expansion de M1.

Cette action est repérée par le symbole associé au rectangle dans lequel est écrite celle-ci (étape 91 du grafcet).

Figure 7.32. Expansion de la macro-étape M1 avec comptage de cartons traité séparément.



REMARQUE

La norme propose également une action mémorisée associée au franchissement de transition, mais elle n'est pas traitée dans cet ouvrage.

VI. Structuration des descriptions réalisées à l'aide du GRAFCET

La complexité des systèmes automatisés rend nécessaire de disposer de moyens pour structurer la spécification. Cette structuration, assistée ou non par des méthodologies adaptées, peut se limiter à un simple « découpage » de la spécification, utiliser des macro-étapes, intégrer des notions de hiérarchie par forçage, ou utiliser l'encapsulation (non traitée dans l'ouvrage).

1. Structuration d'un grafcet à l'aide de macro-étapes

Pour améliorer leur compréhension, les descriptions du comportement élaborées sous forme de grafquets, peuvent être représentées à plusieurs niveaux de description (chapitre 4) par « macro-représentation » exprimant la fonction à remplir sans se soucier de tous les détails superflus à ce niveau de description.

L'utilisation de macro-étapes dans le grafcet de production normale GPN du palettiseur, permet une description de la coordination des sous-ensembles fonctionnels (formation d'une rangée, prise d'une rangée, dépose d'une rangée) sans encombrer le grafcet obtenu par le détail de la commande des actions élémentaires (descendre le préhenseur, saisir les cartons, ...). Les expansions des macro-étapes permettent une description détaillée du fonctionnement (voir paragraphe IV.5.).

2. Partition d'un grafcet

La description du comportement peut être réalisée aussi par différents grafquets communiquant entre eux à l'aide de variables d'étapes.

Il est possible par exemple de remplacer les expansions des macro-étapes M1, M2 et M3 du grafcet GPN du palettiseur par des grafquets de tâches exécutées par la commande (Formation d'une rangée, Prise d'une rangée et Dépose d'une rangée).

La partition en plusieurs grafquets amène à créer une situation initiale dans chaque grafquet de tâche (étape 20 dans le grafquet, Figure 7.33). Pour assurer le lancement de la tâche de commande à l'activation de l'étape 2 du grafquet GPN (Figure 7.33), la réceptivité associée à la transition (20) est X2 (variable d'étape associée à l'étape 2). La création d'une étape 24FIN permet au grafquet GPN d'évoluer vers la situation suivante [2-1, 3]. L'étape 24FIN active, le grafquet de la tâche « Former une rangée » revient en situation initiale après désactivation de l'étape 2 (réceptivité X2).

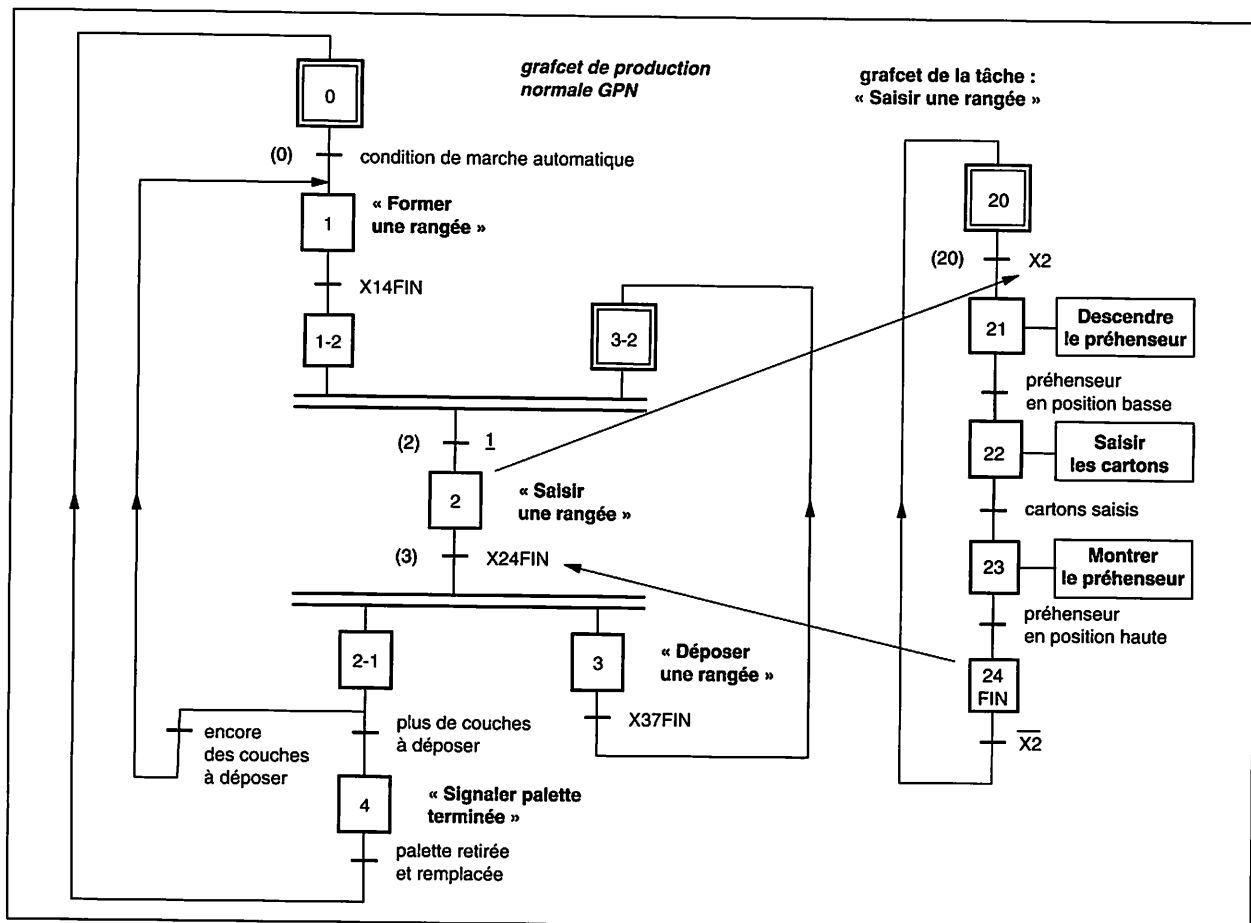


Figure 7.33. Grafcet GPN de coordination de trois tâches. Seule la tâche Saisir une rangée est représentée.

Cette technique de communication entre grafquets par variables d'étapes peut être utilisée également pour l'échange de toutes les variables de synchronisation entre les grafquets des autres niveaux de décision : surveillance et conduite notamment (chapitre 4).

3. Structuration par forçage de situation d'un grafcet partiel

Dans le cas d'une structuration de la spécification de la partie séquentielle d'un système par les modes de marche et d'arrêt, on utilise souvent les ordres de forçage. Ces ordres permettent d'imposer une situation spécifique à un grafcet donné, à partir de la situation d'un autre grafcet. Associé à l'activité d'une étape d'un grafcet hiérarchiquement supérieur (par exemple : un grafcet de surveillance qui, dans une situation donnée, impose une situation à un grafcet de conduite), l'ordre de forçage est un ordre interne qui permet d'imposer une situation à un grafcet hiérarchiquement inférieur.

L'ordre de forçage est représenté dans un double rectangle associé à l'étape pour le différencier d'une action (Figure 7.34).

Forçage d'un grafcet à une situation déterminée	Forçage d'un grafcet à la situation courante	Forçage d'un grafcet à la situation initiale	Forçage d'un grafcet à la situation vide
<p>Lorsque l'étape 1 est active, le grafcet G10 est forcé dans la situation caractérisée par l'activité des étapes 12, 14. Les autres étapes du grafcet G10 sont désactivées.</p>	<p>Lorsque l'étape 20 est active, le grafcet G30 est forcé dans la situation où il se trouve à l'instant du forçage. On appelle également cet ordre « figeage ».</p>	<p>Lorsque l'étape 33 est active, le grafcet G40 est forcé dans la situation dans laquelle seules ses étapes initiales sont actives.</p>	<p>Lorsque l'étape 41 est active, le grafcet G5 est forcé dans la situation vide. Dans ce cas aucune des étapes de G5 n'est active</p>

Figure 7.34. Les différents ordres de forçage.

Dans le grafcet GDEF, Figure 7.35, l'activation de l'étape 201 provoque le figeage des grafquets de conduite GCOND et de production normale GPN. Ces grafquets hiérarchiquement inférieurs dans la hiérarchie de décisions (chapitre 4), sont donc figés dans la situation en cours. Seule la désactivation de l'étape 201 est susceptible d'autoriser à nouveau une évolution de la situation dans ces grafquets.

L'étape 202 permet de forcer le grafcet dans la situation {104}, ce qui positionne la commande en mode manuel tout en garantissant la cohérence entre la situation de GCOND et le mode manuel.

L'utilisation des ordres de forçage dans une spécification implique une structuration hiérarchique en différents grafquets telle que, tout grafcet forçant soit de niveau hiérarchique supérieur à celui de tous les grafquets forcés (Figure 7.36).

Le forçage est un ordre interne dont l'exécution est prioritaire sur l'application des règles d'évolution. Le grafcet forcé ne peut pas évoluer tant que dure l'ordre de forçage.

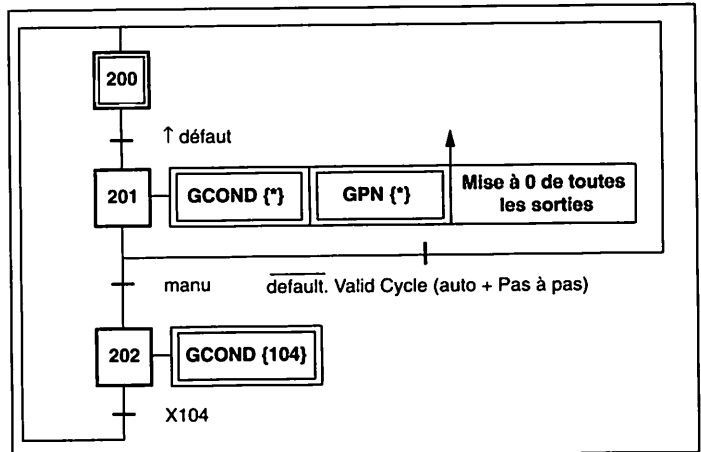


Figure 7.35. Grafcet de surveillance des défauts du palettiseur GDEF.

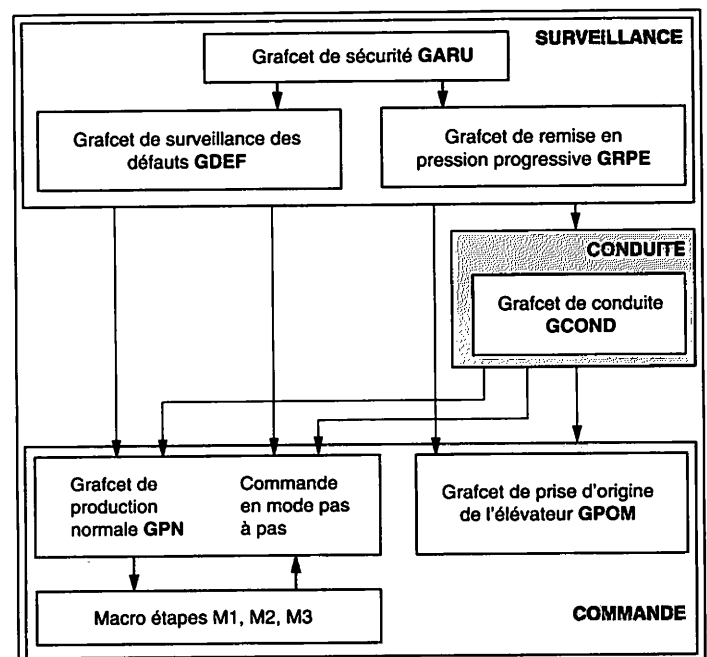


Figure 7.36. Hiérarchie de décision retenue dans le cas du palettiseur des défauts du palettiseur GDEF.

VII. Exemple d'application

Au cours des différentes étapes du cycle de vie, de la conception générale à la réalisation, l'approche de conception de la commande est progressive (chapitre 4 paragraphe III).

Une première illustration est fournie par l'exemple du palettiseur dont les éléments caractéristiques sont donnés dans la fiche système 1. À partir de l'analyse fonctionnelle, il est possible de dégager les principales fonctions à assurer et de proposer un enchaînement qui servira de base aux affinements successifs. L'application ne concerne pas les niveaux de décision : surveillance et conduite.

1. Commande du procédé

a) Description globale du procédé

Une première analyse permet de retenir les tâches de commandes suivantes :

- « Acquérir les paramètres de palettisation » (hauteur des cartons, nombre de rangées) ;
- « Charger et décharger manuellement la palette » ;
- « Amener les cartons » (qui proviennent d'une installation en amont) ;
- « Former une rangée de deux cartons » ;
- « Transférer la rangée formée ».

Une description est proposée sur la figure 7.37, qui établit un parallèle entre une description de type algorithmique et une description de type GRAFCET. Cette première description introduit deux boucles d'itération, la première est relative au nombre de rangées de deux cartons à palettiser (paramètres de palettisation), la seconde est relative au nombre de palettes d'un type donné à réaliser (décision de l'opérateur). Dans le grafcet, les boucles sont représentées par des reprises de séquences.

La description intègre une tâche de commande (« Acquisition des paramètres ») à côté des tâches opératives dont une est manuelle : « Charger et décharger la palette ».

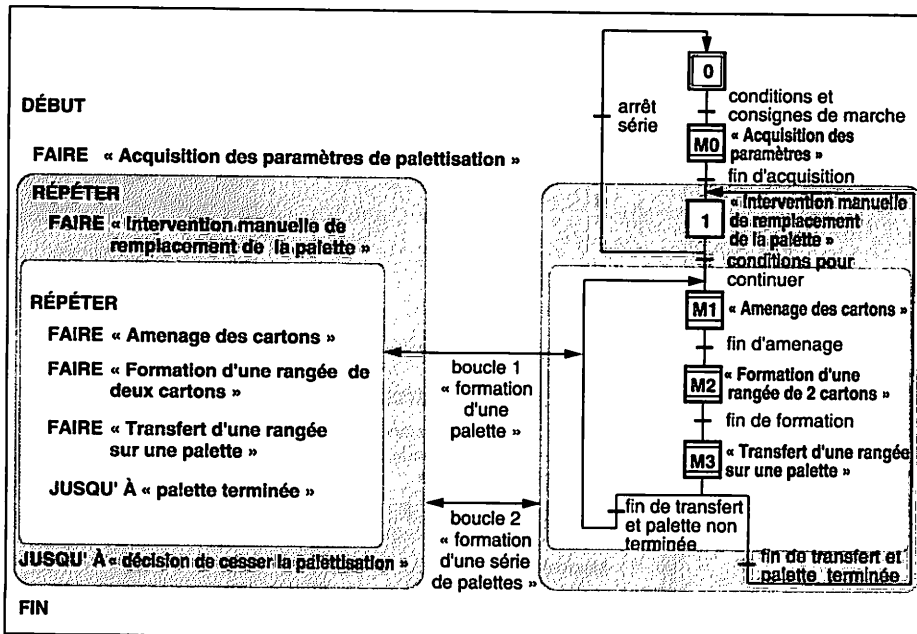


Figure 7.37. Description globale du procédé.

b) Coordination des tâches

La succession des tâches dans le temps peut être décrite par le chronogramme, Figure 7.38, où toutes les opérations sont montrées en même temps. Du fait du caractère linéaire de la description proposée, le temps de cycle est la somme des durées de chaque activité. Durant l'exécution du transfert sur la palette des deux cartons, l'aménagement est suspendu, de même que la formation d'une rangée.

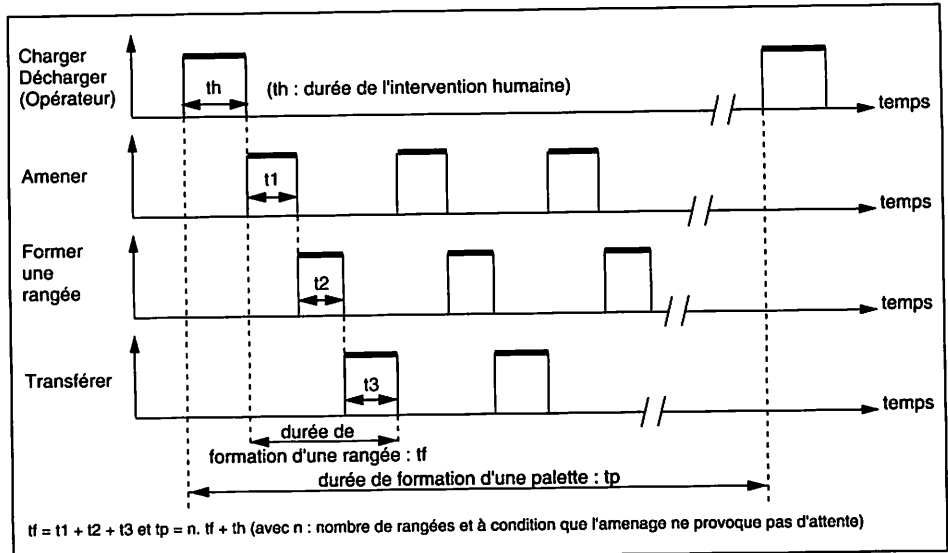


Figure 7.38. Équipement de palettisation, description par chronogrammes.

Cela est acceptable si le temps de cycle obtenu est compatible avec la cadence imposée par le client. Dans cette application, le temps de cycle spécifié impose l'exécution d'opérations en parallèle. Cette exigence implique de revoir le découpage des tâches. La description proposée se limite aux seules tâches opératives pour lesquelles on retient le découpage suivant :

- tâche 1 : « Formation d'une rangée de deux cartons » (fonctions : « Amener les cartons introduits » et « Former une rangée de deux cartons »);
- tâche 2 : « Saisie de la rangée »;
- tâche 3 : « Dépose des deux cartons ».

Un graphe de coordination des tâches appelé grafcet de production normale (GPN) peut alors être proposé après avoir fait un bilan des entrées et des sorties (Figure 7.39).

Les expansions des macro-étapes sont définies dans un deuxième temps. Elles n'apparaissent pas dans GPN, Figure 7.40, ce qui explique que les réceptivités qui suivent les macro-étapes ne soient pas toujours vraies comme c'est le cas dans le paragraphe IV.5, Figure 7.19.

Dans le grafcet de la figure 7.40, les tâches M1 et M3 peuvent s'exécuter en parallèle (séquences parallèles) ce qui permet de réduire le temps de cycle.

La situation initiale {0, 3-2} a été définies à partir des précisions apportées par le rectangle état A1 dans le GEMMA du palettiseur FICHE SYSTÈME I (palette vide, POM effectuées, ...).

Une autre représentation, Figure 7.41, permet de mieux le faire apparaître.

Le nouveau chronogramme montre le parallélisme des tâches et la réduction du temps de cycle qui en résulte (tM3 dépend de l'endroit de dépose : cas a et b).

Figure 7.40. Grafcet de production normale GPN du palettiseur.

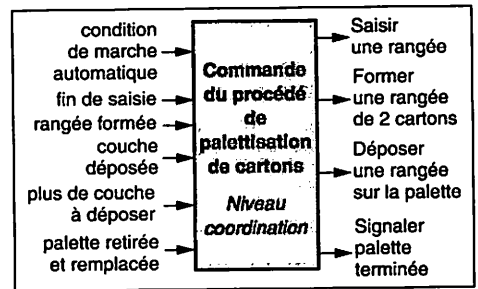
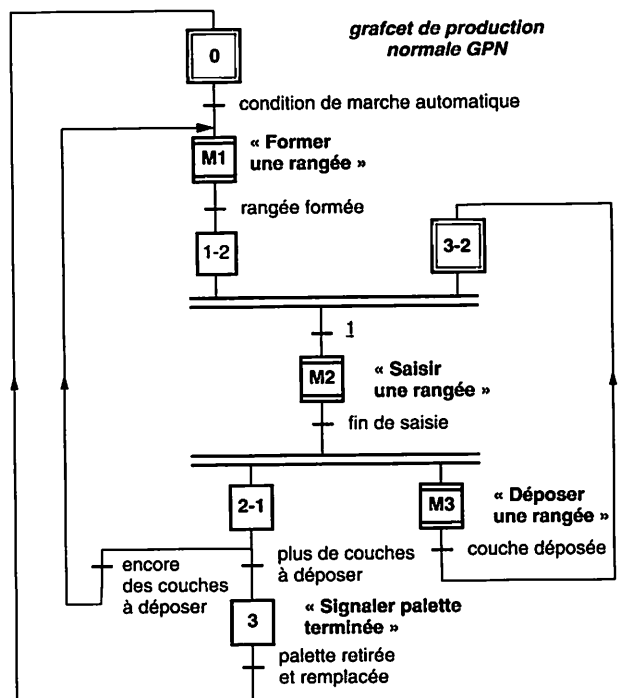


Figure 7.39. Entrées et sorties utilisées dans le grafcet GPN



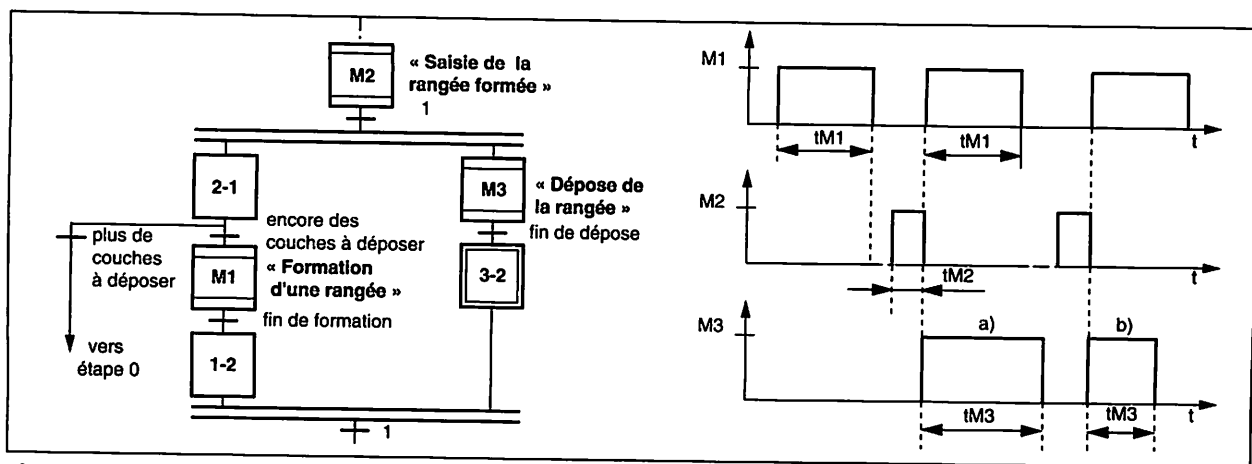


Figure 7.41. Description sous forme de séquences parallèles.

c) Description de tâches indépendantes

Un exemple emprunté à la machine de conditionnement automatique illustre une coordination des tâches réalisées par des informations associées à l'état des produits. «Conditionner des comprimés en flacons» constitue la fonction globale de l'équipement. Les flacons et bouchons, placés sur des palettes, circulent librement sur un convoyeur et peuvent être traités sur la ligne par des postes de travail. Cette structure de transfert permet des modifications ou évolutions aisées du processus. Ainsi, le procédé étant choisi, il est possible au premier abord de distinguer les trois tâches de transfert, remplissage, bouchage et d'élaborer une description du fonctionnement qui se présente sous la forme de graphes séparés.

Les liens de synchronisation étant réalisés par la présence de produits adéquats aux deux postes de remplissage et bouchage, ils sont repérés sur le grafcet de la figure 7.42 par des conditions écrites en italiques.

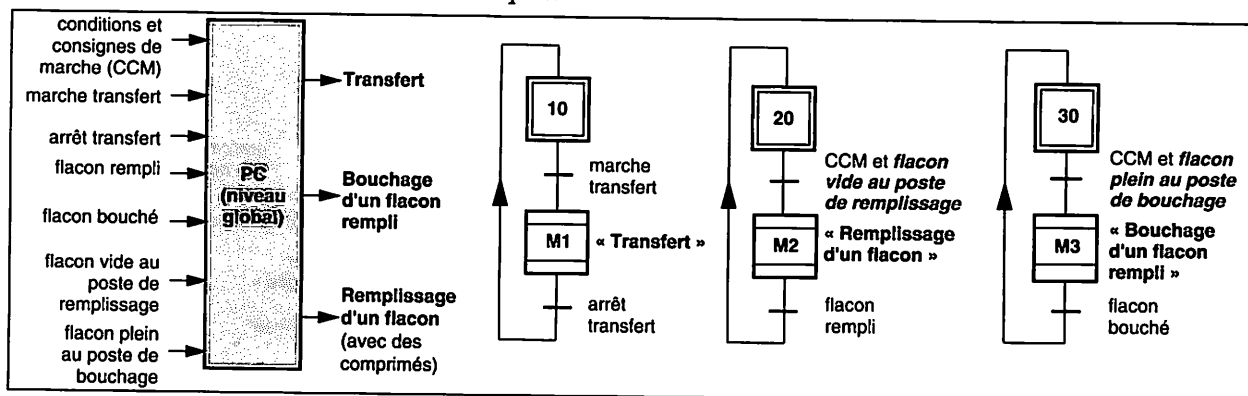


Figure 7.42. Graphes indépendant de coordination de tâches.

Le chronogramme montre que plusieurs tâches peuvent être réalisées en même temps (Figure 7.43) :

- à la date t1, seule la tâche « Transfert » (M1) est en cours. Tous les flacons avancent sur le convoyeur;
- à t2, un flacon vide est présent au poste de remplissage et lance la tâche M2 (les flacons continuent d'avancer);
- à t3, le remplissage est terminé, les flacons avancent sous l'action de la tâche « Transfert »;
- à t4, remplissage d'un nouveau flacon;
- à t5, mise en position d'un flacon plein et lancement de la tâche M3 (« Bouchage »);
- à t6, les trois tâches s'exécutent parallèlement.

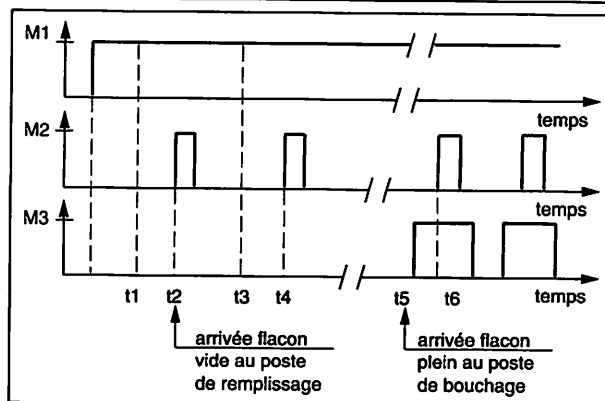


Figure 7.43. Repères temporels.

On observera, cependant, que les opérations ne sont pas synchrones et que le début d'exécution des tâches est lié à l'arrivée du produit adéquat sur le poste (Figure 7.44).

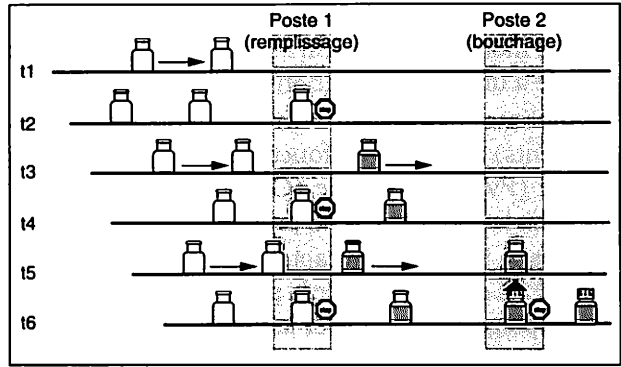


Figure 7.44. États des postes et des flacons sur le convoyeur aux dates t1 à t6.

2. Commande des organes permettant d'agir sur la partie opérative et d'informer l'opérateur

a) Des effecteurs aux actionneurs

La figure 7.45 illustre les choix techniques de l'équipement de palettisation. C'est un équipement qui associe les technologies pneumatique et électrique.

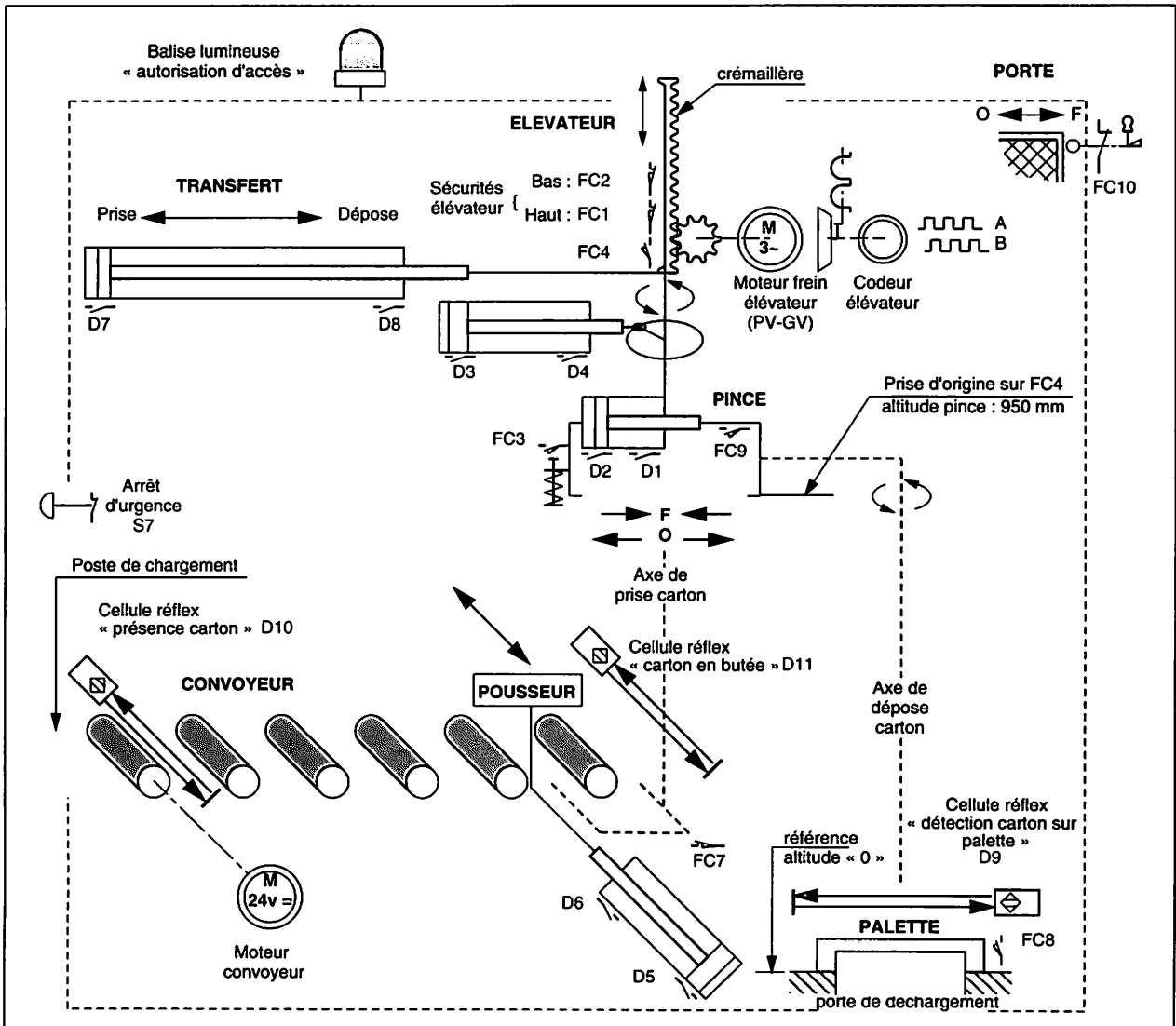


Figure 7.45. Éléments techniques du palettiseur.

À côté des actionneurs conventionnels du type vérins et moteurs, on retiendra pour le mouvement vertical du manipulateur le choix d'un moteur à deux vitesses, associé à une transformation de mouvement de type « pignon-crémaillère ».

Ce moteur possède deux enroulements statoriques distincts (nombre de pôles différents) alimentés en fonction de la vitesse souhaitée. Le rapport entre la petite vitesse (PV) et la grande (GV) ne dépasse pas 6 pour des moteurs standard. Au-delà, le coût élevé rend prohibitif leur utilisation industrielle.

On notera le choix d'un capteur incrémental sur ce même mouvement pour rendre compte du déplacement de la pince (et du carton). Un traitement spécifique permet le comptage des signaux et la mesure des déplacements. On notera également le choix de nombreux détecteurs de proximité (magnétiques, cellules photo-électriques).

Une première approche selon les objectifs de représentation : commande des mouvements des « effecteurs » puis commande des mouvements des « actionneurs », est limitée à la tâche « Formation d'une rangée de deux cartons ».

La figure 7.46 montre les entrées sorties relatives à cette tâche.

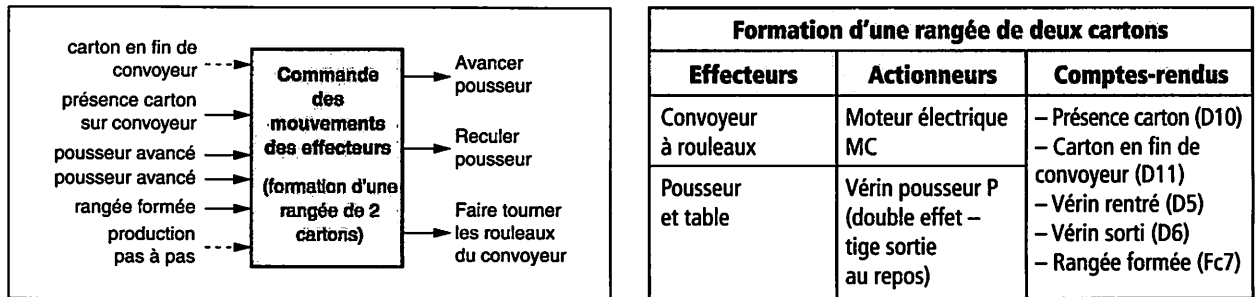


Figure 7.46. Liste des entrées et des sorties de la tâche « formation d'une rangée de 2 cartons ».

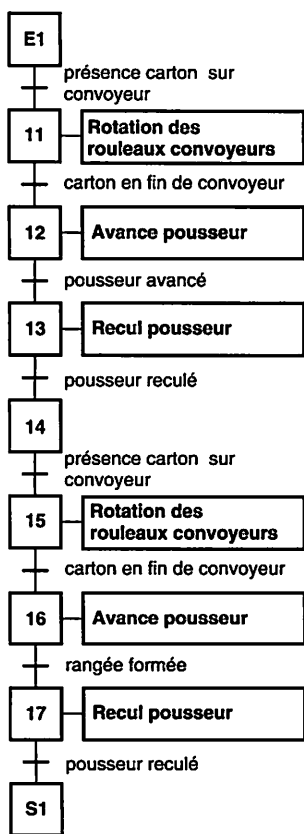


Figure 7.47. Expansion de la macro-étape M1, commande des effecteurs.

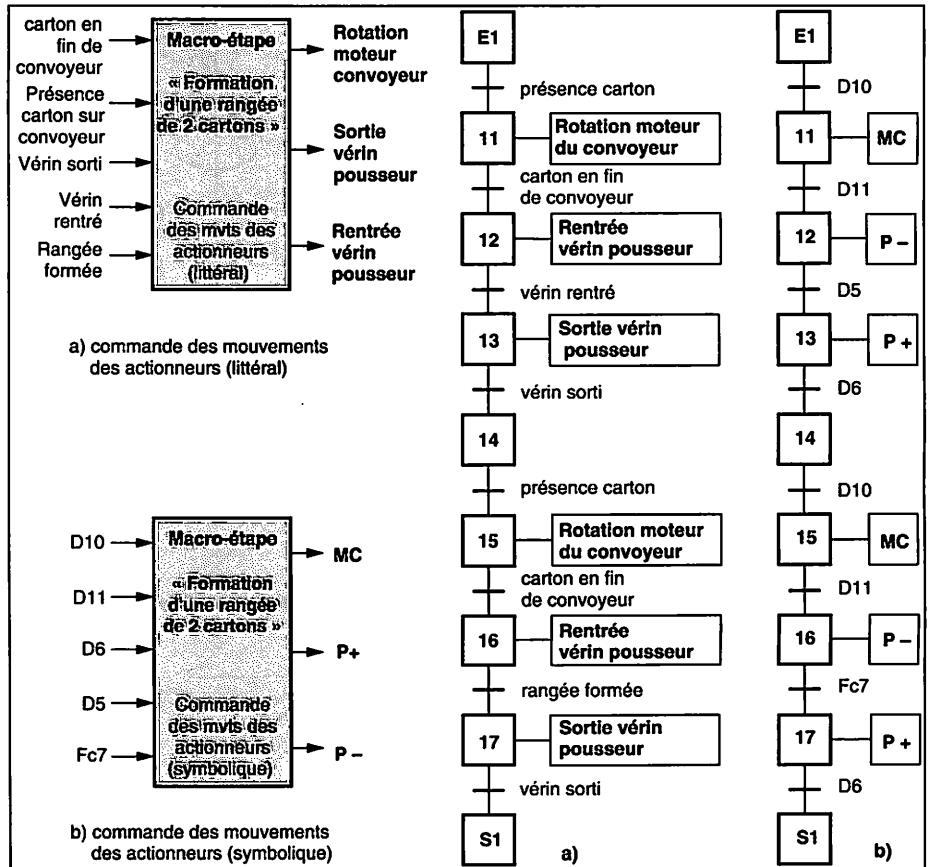


Figure 7.48. Expansion de la macro-étape M1, commande des mouvements des actionneurs.

La figure 7.47 (ci-contre) présente le grafcet de l'expansion de la macro-étape résultant des seuls choix « effecteurs ». La figure 7.48 donne une description du comportement de la commande des mouvements des actionneurs.

Trois points sont à noter :

- la disposition technique du vérin pousseur (P) inverse la logique des déplacements entre le dispositif pousseur et le vérin : le carton avance quand la tige du vérin rentre, et réciproquement ;
- la position de référence du vérin est la position tige sortie ;
- les descriptions a) et b) diffèrent simplement par les désignations utilisées (en clair pour la première, symbolique pour la seconde). La symbolisation simplifie l'écriture des graphes, elle ne change absolument pas la signification de la description.

On retiendra principalement que les actionneurs sont repérés par des identificateurs ayant une signification (A, PI, V1, V2, MC, caractérisant la fonction assurée), que les actions de sortie de tige sont affectées du signe + et celles de rentrée de tige - (ces signes pourront être omis quand une seule action est commandée). Pour les moteurs, on pourra utiliser la même convention de représentation : + (ou AV) pour une rotation dans le sens des aiguilles d'une horloge, - (ou AR) dans l'autre sens.

b) Des actionneurs aux pré-actionneurs

L'association actionneur/pré-actionneur est imposée par la technologie. Pour obtenir des comportements de type logique, le moteur électrique est alimenté par un contacteur, le vérin par un distributeur. L'un des choix essentiels réside dans le type de commande du pré-actionneur selon que celui-ci est monostable ou bistable.

● Commande des pré-actionneurs TOR

Un pré-actionneur est dit monostable (un seul état stable) s'il a besoin, pour passer de sa position de repos à sa position de travail, et s'y maintenir, d'un ordre permanent. Si cet ordre disparaît, le pré-actionneur revient dans sa position repos (fréquemment sous l'effet d'un ressort).

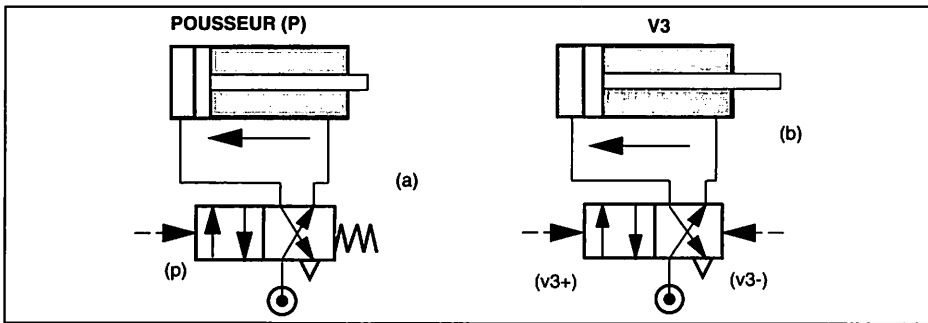


Figure 7.49. Distributeurs simple et double pilotage.

Une seule commande (ou pilotage) est suffisante (Figure 7.49 cas a).

Un préactionneur est dit bistable (deux états stables), s'il a besoin pour passer de sa position repos à sa position travail et s'y maintenir d'un ordre à caractère impulsionnel (qui ne dure pas). Le changement d'état nécessite l'envoi d'un second ordre antagoniste.

Le préactionneur possédant un comportement de type « mémoire » deux commandes sont donc nécessaires : commandes « travail » et « repos » (Figure 7.49 cas b).

Les commandes sont souvent repérées par des minuscules reprenant le nom de l'actionneur. Pour la commande de pré-actionneur bistable, on affectera le signe + ou - pour signifier l'ordre « sortie de tige » ou « rentrée de tige »; pour la commande de pré-actionneur monostable, ces signes pourront être omis, permettant un repérage plus aisé.

REMARQUE

Il existe des normes de représentation des circuits électriques et pneumatiques. Cependant le maniement de ces symbolisations normalisées n'est pas toujours pratique en phase de conception et les automaticiens préfèrent souvent utiliser une symbolisation plus signifiante.

REMARQUE

Les contacteurs sont essentiellement monostables. Un vérin double effet pourra être alimenté à partir d'un distributeur monostable (Figure 7.49, cas a), alors qu'un vérin simple effet pourra être alimenté à partir d'un distributeur bistable (Figure 7.50, cas c).

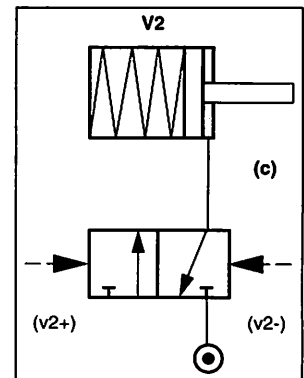
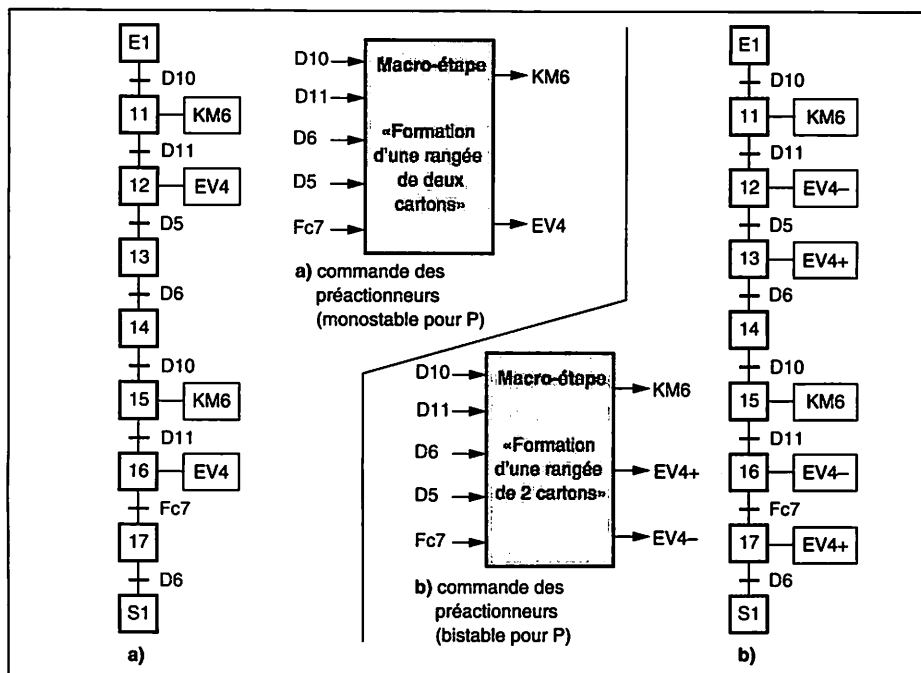


Figure 7.50. Vérin simple effet et son distributeur double pilotage.

Le choix du type de pilotage des distributeurs répond à des exigences de sûreté de fonctionnement en cas de coupure des énergies de commande : retour en position repos pour la commande monostable, maintien dans la position pour la commande bistable.

Pour le palettiseur, la figure 7.51 illustre, à partir de l'expansion de la macro-étape « Formation d'une rangée de deux cartons » (commande des mouvements des actionneurs), une description obtenue à partir des choix concernant le pilotage (EV4) du distributeur « P » du vérin pousseur (monostable et bistable). On notera que le constructeur a retenu la première possibilité (a). Cette description qui traduit des préoccupations de comportement en cas de défaillance des pilotages est du domaine des spécifications opérationnelles.



REMARQUE

Dans le cas de la figure 7.51, les repères utilisés pour les entrées et les sorties désignent les échanges informationnels entre la commande et les composants (capteurs, éléments du pupitre, pré-actionneurs) en utilisant les mnémoniques, plus faciles à employer que les repères technologiques. Ainsi le pilote du distributeur est repéré EV4, alors que le repérage technologique (sur schéma notamment) est 4YV1. Cette pratique, même si elle est commode, nécessite que la codification soit univoque.

Figure 7.51. Expansion de M1, commande des pré-actionneurs.

c. Commande programmée

L'objectif est de décrire les ordres codés, élaborés par la commande programmée.

- Il s'agit de décrire comment sont élaborés les sorties de la commande programmable (%Q4,10; %Q2,3) à partir des entrées (%I3.3; %I3.4; %I1.14; %I1.13; %I3.5). C'est l'ensemble du traitement qui est alors pris en compte, y compris le traitement logique intégré dans les schémas câblés (mise en marche, sécurités,...).
- La partie opérative n'est vue ici qu'à travers les entrées/sorties logiques de la partie commande, donc uniquement sous forme informationnelle.

L'élaboration des modèles de comportement (SFC : langage inspiré du Grafcet, voir chapitre 8) se fait directement dans des ateliers logiciels (FICHE TECHNIQUE 4) capables de générer automatiquement le code pour la commande programmée.

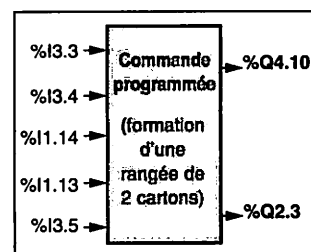


Figure 7.52. Ordres codés, élaborés par la commande programmée.

- Le GRAFCET est un langage de spécification qui permet la description fonctionnelle du comportement de la partie séquentielle des commandes des systèmes automatiques.
- Le bilan des entrées et des sorties de la commande dont on s'attache à décrire le comportement est un préalable à toute élaboration d'un grafcet.
- L'étape est l'élément du langage GRAFCET utilisé pour définir la situation de la partie séquentielle du système. Une étape est soit active, soit inactive. La situation désigne l'état du système spécifié par un grafcet et caractérisé par les étapes actives à un instant considéré.
- Un événement d'entrée est caractérisé par le changement de valeur d'une ou plusieurs variables d'entrées de la partie séquentielle du système. Un événement interne est caractérisé par un événement d'entrée associé à la situation de la partie séquentielle du système.
- La dernière révision de la norme GRAFCET propose de manière explicite un ensemble de règles (syntaxe et évolution) et des commentaires associés qui permettent une interprétation fine du comportement indiqué. Les actions permettent d'établir le lien entre l'évolution d'un grafcet et les sorties. Deux modes de sortie, mode continu (assignation sur état) ou mode mémorisé (affectation sur événement), décrivent comment les sorties dépendent de l'évolution des entrées du système. L'utilisation des macro-étapes, des ordres de forçage ou de la partition en différents grafquets permet de structurer les descriptions comportementales.

Exercices

EXERCICE 1. Identifier les différentes séquences et les structures de base des grafquets, FICHE SYSTÈME I pour les grafquets GARU, GDEF et GPRE.

EXERCICE 2. Indiquer, dans les cas de choix de séquences des grafquets GDEF et GPRE de la FICHE SYSTÈME I, les réceptivités associées exclusives.

EXERCICE 3. On considère le grafcet de prise d'origine du palettiseur. La situation actuelle est {91}. Analyser comment le grafcet va évoluer vers les situations {92} ou {93} et représenter les chronogrammes des activités de ces mêmes étapes ainsi que des réceptivités concernées (Élévateur en position haute, 3s/X92).

EXERCICE 4. Identifier, dans les FICHES SYSTÈMES I ET 2, actions mémorisées et les actions continues conditionnées ou non.

EXERCICE 5. Proposer pour le poste de bouchage, compte tenu du procédé choisi, une décomposition en tâches similaire à celle du remplissage. Donner une description par GRAFCET.

EXERCICE 6. Est-il possible de réduire la durée du cycle de bouchage ? À quelle(s) condition(s) ?

EXERCICE 7. Élaborer le grafcet décrivant le comportement de l'effecteur « préhenseur à ventouse » de la machine de conditionnement après avoir établi la liste des entrées/sorties.

EXERCICE 8. Donner le grafcet du comportement souhaité des mouvements des actionneurs du manipulateur de la machine de conditionnement après avoir établi la liste des entrées/sorties.

8

Réalisation d'une commande

Réaliser une partie commande, c'est créer un ensemble de circuits capables de contrôler les énergies, de traiter l'information et de communiquer avec son environnement, dans le but de commander et de piloter des parties opératives d'un système automatique.

I. Mixité câblé – programmé

Le développement de la microélectronique et de ses capacités d'intégration considérables a permis l'essor de la logique programmée. La réalisation matérielle des chaînes fonctionnelles d'un équipement automatisé et les liaisons avec l'environnement (interfaces homme/machine, communication entre diverses commandes, ...) nécessitent toutefois l'interconnexion des divers composants ou constituants par câblage. Le câblage consiste à relier les composants ou constituants de manière à véhiculer les énergies et les signaux selon une logique permettant le fonctionnement attendu.

Dans cet ouvrage, les réalisations câblées électroniques ne sont pas abordées.

II. Réalisation par câblage

Le lecteur pourra se reporter également à la **FICHE MÉTHODE 5** pour l'élaboration des schémas relatifs aux réalisations câblées. De plus en plus, les équipements automatiques sont réalisés autour de constituants standard couvrant des cas les plus élémentaires (commandes logiques) aux plus complexes (fonctions d'asservissement, de communication, etc.). D'autres technologies de commande (cartes électroniques standard ou spécifiques) sont employées quand le nombre d'équipements identiques est important et que les coûts de développement d'un ensemble singulier sont compatibles avec l'économie du projet.

● Entrées fonctionnelles et de sécurité

Les entrées fonctionnelles sont celles qui permettent de contrôler le processus, dans le cadre de son évolution normale.

Les entrées de sécurité sont relatives au contrôle du processus en cas d'anomalies et de dysfonctionnements. Ces entrées doivent permettre d'interrompre le plus rapidement possible le processus, ou le conduire vers un comportement orienté afin d'éviter ou de limiter les dégâts aux personnes et aux biens.

L'INRS (Institut National de Recherche sur la Sécurité), en 1984, recommandait de ne pas faire confiance au seul automate programmable pour assurer la gestion des fonctions de sécurité et préconisait le recours aux solutions câblées extérieures. En effet, les modes de défaillances des constituants électroniques, peuvent induire des comportements cataclysmiques (toutes les sorties actives par exemple en cas de court-circuit), à l'inverse des composants électromécaniques, plus robustes et dont la défaillance est généralement circonscrite au proche environnement.

En tout état de cause, il est nécessaire de s'assurer :

- que la gestion des fonctions de sécurité soit séparée de la gestion de la partie fonctionnelle;
- les fonctions de sécurité soient figées et non modifiables par l'utilisateur.

La coexistence des logiques câblées et programmées est la règle dans une majorité d'applications pour la réalisation des circuits de sécurité et de contrôle de l'énergie (Figure 8.1).

Toutefois, certains constructeurs s'appuyant sur une meilleure connaissance des modes de défaillance, et des constituants plus sûrs proposent, pour les équipements de production, des solutions à partir de composants électroniques et d'automates dédiés sécurité (APIdS).

● **Commande avec entrées de sécurité câblées**

En logique électrique, l'élément technologique de base est le contact « à établissement de circuit » (la variable complémentée est matérialisée par le contact « à interruption de circuit »). Ce contact peut être actionné mécaniquement (bouton-poussoir, fin de course), ou électro-mécaniquement (relais ou contacteur).

La figure 8.2 empruntée à la notice du palettiseur illustre cette technique. Par exemple, la bobine du contacteur KM3 (Descente de l'élévateur) sera alimentée si la sortie Q2.5 est activée et si les fins de course S12 (Fc2) et S13 (Fc3), ne sont pas actionnés, de même que le contact d'inter verrouillage KM2 (KM2 non alimenté). Ceci peut se traduire, sous la forme d'une expression, par :

$$KM3 = Q2.5 \cdot \overline{KM2} \cdot \overline{Fc3} \cdot \overline{Fc2}$$

entrées de sécurité

La mise en série des contacts à ouverture réalise une fonction de sécurité (cas de la figure 8.2). Quelques règles générales sont à retenir :

- les ordres d'arrêts (sécurité d'arrêt) sont associés à des contacts à ouverture;
- les contacts de sécurité F10 (RT1) (Figure 8.2) agissent directement sur l'alimentation du préactionneur. Cela permet des réactions plus sûres et plus courtes (liaison directe avec le préactionneur);
- le traitement est doublé (redondance : le contact F10 (RT1) est également une entrée de l'automate).

● **Principe d'une commande séquentielle câblée**

La réalisation d'une commande séquentielle câblée (fonction mémoire) est illustrée par le schéma de la figure 8.3 (voir également le chapitre 6). L'appui sur le bouton-poussoir S₂ permet l'alimentation de la bobine du relais K₁, qui entraîne la fermeture des contacts associés. Un double circuit d'alimentation est donc réalisé. L'action sur S₂ peut alors être relâchée, la bobine K₁ continue à être alimentée par un de ses propres contacts, c'est l'auto alimentation ou « auto maintien ». Cette activation durera tant que la bobine sera alimentée. Cette alimentation cessera par appui sur S₁.

La fonction mémoire ainsi réalisée est une mémoire à arrêt prioritaire : le maintien de l'appui simultané sur S₁ et S₂ fera que la bobine ne sera pas alimentée. C'est la forme classique utilisée dans la commande des machines notamment pour la mise en énergie d'un système automatique. Il existe également des circuits électroniques (câblés ou programmés) remplissant cette fonction.

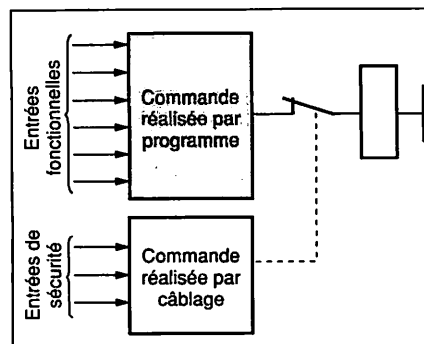


Figure 8.1. Commande mixte d'une chaîne fonctionnelle.

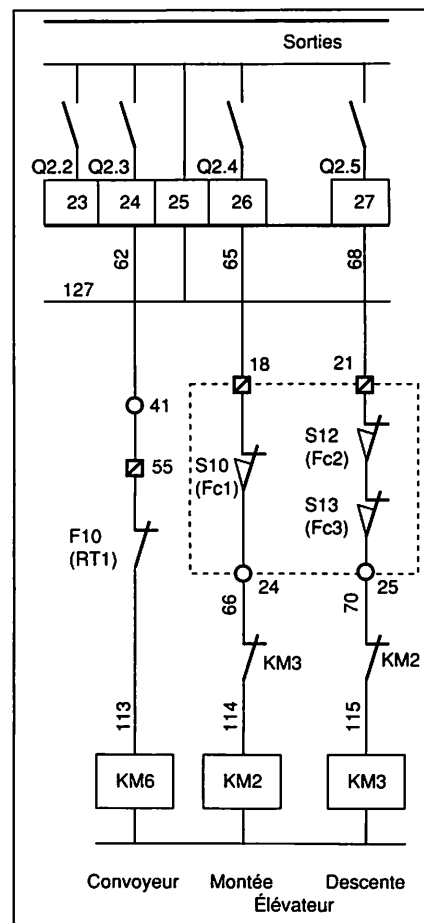


Figure 8.2. Schéma de câblage des pré-actionneurs (extrait).

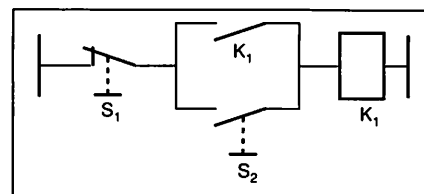


Figure 8.3. Schéma de réalisation câblée d'une mémoire.

III. Introduction aux réalisations programmées

● La programmation des automates programmables industriels

La volonté des utilisateurs d'automates programmables de disposer de configurations d'automates (automates et périphériques) aux caractéristiques et aux langages de programmation normalisés a permis à partir de 1992 la parution d'une norme repérée CEI 61131.

Cette norme est composée de cinq parties : informations générales, spécifications et essais des équipements (partie destinée plutôt aux constructeurs), langages de programmation, guide pour l'utilisateur et communication (réseaux).

Elle s'applique :

- aux automates programmables ;
- et aux périphériques associés tels que les outils de programmation et de mise au point, les équipements de test, les interfaces homme - machine.

La normalisation des langages de programmation des automates programmables (CEI61131-3), a permis de standardiser :

- la terminologie (reprise à chaud, à froid, instance, délimiteur, double mot, ...);
- les types de données, les symboles utilisés, l'identification des variables (entrées, sorties, mémoire interne);
- les langages (liste d'instructions, langages graphiques et langage littéral).

Langages littéraux
- IL liste d'instructions - ST langage littéral structuré
Langages graphiques
- LD langage à contacts - FBD langage à blocs fonctionnels
Langage graphique inspiré du GRAFCET
Langage SFC (Sequential Function Chart)

1. Les langages de programmation

La norme propose trois classes de langages ci-contre.

a) Éléments communs aux différents langages

Les éléments communs à tous les langages de programmation sont les suivants :

Les identificateurs	ARRET_TECHN, RETOUR_OK	Un identificateur désigne de manière univoque une variable. Il a du sens pour l'exploitant ou l'automaticien.
Les mots clés	FUNCTION, END_FUNCTION_BLOCK	Ce sont des mots réservés pour exécuter des opérations ou déclarer des modules logiciels.
Les commentaires	(*production normale*)	
Les libellés	+234, 16#E0, 'ARRET', TIME#2.7s	Numériques (dans une base donnée), de chaînes de caractères, de datation et de temps.

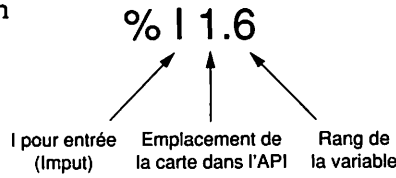
Pour pouvoir traiter des données, il est nécessaire de connaître leur emplacement (entrée API, sortie API, mémoire interne à l'API), leur format et leur type (bit, mot de 8 bits mot de 16 bits, ...).

Type de variable			Variables	
Mnémonique	Signification	Format	Préfixe	Signification
BOOL	Booléen	1 bit	I	Emplacement d'entrée
BYTE	Chaîne de bits	8 bits	Q	Emplacement de sortie
WORD	Mot	16 bits	M	Emplacement de mémoire
DWORD	Mot double	32 bits	X	Taille d'un seul bit (peut être omis)
LWORD	Mot long	64 bits	aucun	"
INT	Entier	16 bits	B	Taille d'un octet (8 bits)
UINT	Entier non signé	16 bits	W	Taille d'un mot (16 bits)
UDINT	Entier double non signé	32 bits	D	Taille d'un double mot (32 bits)
			L	Taille d'un mot long (64 bits)

b) Représentation des variables à un seul élément

La représentation directe d'une variable à un seul élément est assurée par l'enchaînement :

- du signe «%»;
- d'un préfixe d'emplacement;
- d'un préfixe de taille;
- et d'un ou plusieurs entiers non signés;
- séparés par le symbole «. ».



EXEMPLES

- %I1.3 (pour une variable logique d'entrée)
- %Q2.6 (pour une variable logique de sortie)
- %M31 (bit interne)
- %MW32 (mot de 16 bits)

c) Langage graphique LD (Ladder ou langage à contacts, Figure 8.4)

S'inspirant des pratiques issues des réalisations par câblage de composants électriques (opération ET réalisée par la mise en série des éléments, opération OU réalisée par la mise en parallèle, ...), l'écriture sous forme Ladder permet une transcription aisée et très intuitive d'expressions logiques, sous forme de réseau de contacts. Enfin, elle permet en phase de mise au point et de maintenance une visualisation commode de l'état des variables par un passage en surbrillance ou une modification de couleur des symboles des variables.

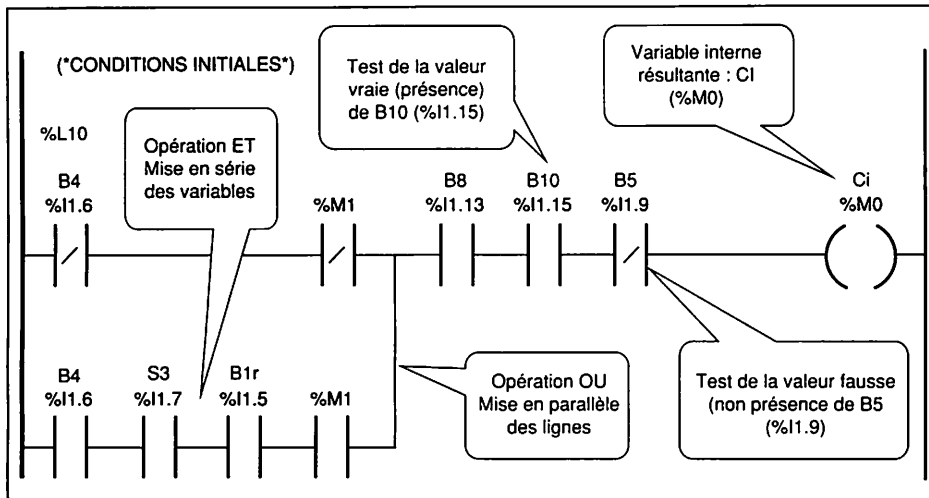


Figure 8.4. Langage LD.

d) Langage graphique FDB (langage à blocs fonctionnels)

Des blocs fonctionnels standard (préprogrammés et paramétrables) sont proposés par les constructeurs (temporisateurs, compteurs, registres, mémoire, ...).

La figure 8.5 illustre l'utilisation d'un temporisateur, bloc fonctionnel standard (%T3) permettant l'obtention d'une variable interne (%M33) retardée par rapport à la valeur vraie de la proposition : %X51.%I1.4.

Il est également possible, pour l'utilisateur de développer des blocs spécifiques (appelés blocs fonctionnels utilisateurs), caractéristiques d'un type d'application (voir page 124).

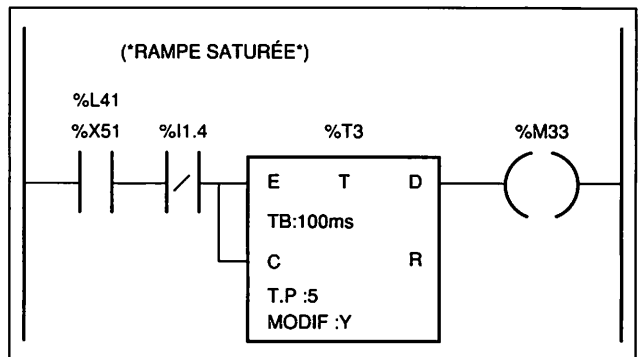


Figure 8.5. Exemple de bloc fonctionnel.

e) Langage graphique SFC inspiré du GRAFCET (Sequentiel Function Chart)

Il permet de structurer l'organisation de tout ou partie d'un programme. C'est un langage qui s'inspire du GRAFCET. Toutefois, le GRAFCET désigne un langage de spécification du comportement indépendant de la technologie employée et est défini par la norme CEI 60848, présenté chapitre 7. Le SFC est un des langages de programmation, définis dans la norme, mais il n'y a pas d'identité entre les deux représentations même si elles sont graphiquement semblables.

Il n'existe pas à ce jour de règles permettant l'interprétation et la conversion automatique d'un grafcet en programme. Il est donc nécessaire d'effectuer l'interprétation du cahier des charges (utilisant la norme CEI 60 848) avant son écriture et implémentation en programme utilisant la norme CEI 61131-3.

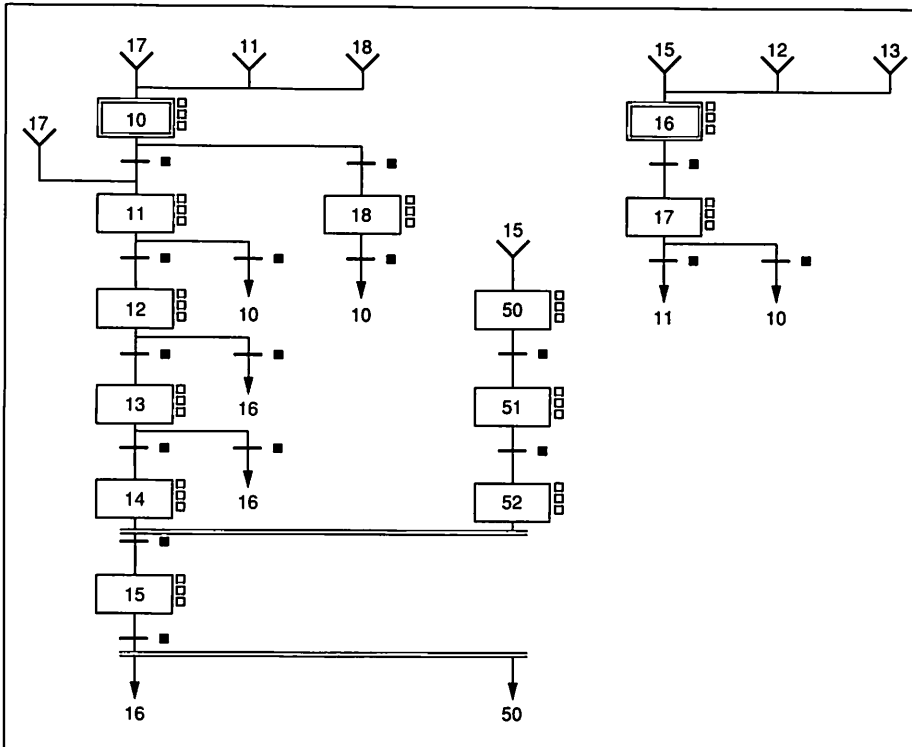


Figure 8.6. Exemple de SFC.

f) Langages littéraux

● IL : liste d'instructions

Ils s'apparentent à la technique d'un langage de bas niveau de type assembleur. Une section écrite en langage liste d'instructions se compose d'une suite d'instructions exécutées séquentiellement par l'automate.

Chaque instruction est formée d'un opérateur (ou «code instruction») et d'un opérande (variable). L'étiquette permet de repérer un début de traitement, elle est optionnelle, de même que le commentaire qui est destiné à faciliter la compréhension à la lecture.

Étiquette	Opérateur	Opérande	Commentaires
%L10	LD	%I1.3	
	AND	%M1	
	ST	%Q2.0	Activation sortie bol vibrant

REMARQUE

Dans l'exemple, Figure 8.6, le SFC a été élaboré grâce à un atelier logiciel. Les réceptivités n'apparaissent pas directement pour éviter de trop charger le graphe. Elles sont définies en cliquant sur la transition concernée (Figure 8.7). Les sorties sont souvent définies dans une zone mémoire différente (Figure 8.8). Essentiellement parce qu'une même sortie peut être affectée à plusieurs étapes pouvant aussi appartenir à des SFC différents. Il est parfois nécessaire d'ajouter un combinatoire lié aux modes de marche et d'arrêt.

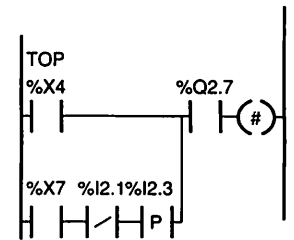


Figure 8.7. Exemple concernant la réceptivité associée à la transition entre les étapes 10 vers 11.

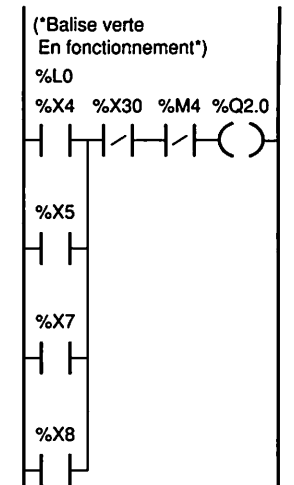


Figure 8.8. Équation de la sortie correspondant à la balise verte.

● **ST langage littéral structuré**

Le langage littéral structuré est un langage évolué de type algorithmique particulièrement adapté à la programmation de fonctions arithmétiques complexes, manipulations de tableaux et gestions de messages.

La figure 8.9 donne un exemple d'une écriture d'un module de comptage en langage littéral structuré (ST).

```
(* Compteur de litres de produit *)
IF RE Raz THEN
  Val_cou:=0;
END_IF;

IF RE Count THEN
  Val_cou:=Val_cou+1;
END_IF;

IF Val_cou>=Presel THEN
  SET Done;
ELSE RESET Done;
END_IF;
```

Figure 8.9. Module de comptage.

2. La structuration des programmes

L'ampleur des programmes développés dans les API et la variété des tâches logicielles à assurer (gestion du cycle machine, sécurités, surveillance, supervision, commande d'axe, régulation, ...) nécessitent l'utilisation d'une approche structurée. Elle facilite l'écriture du programme de l'application, ses modifications et sa réutilisation par une organisation modulaire.

La norme met en œuvre ces principes par l'intermédiaire de la notion de **module logiciel**.

Il existe trois types de modules logiciels écrits dans les langages normalisés :

- la **fonction**;
- le **bloc fonctionnel**;
- le **programme**.

a) **La fonction**

La fonction est un module logiciel ayant :

- plusieurs variables d'entrée possibles;
- une seule variable de sortie;
- pas de mémoire interne;
- parfois une entrée EN (validation) et une sortie ENO (pas d'erreur).

Les modules logiciels permettant de réaliser une conversion de type de variable, des opérations arithmétiques, des sélections, des comparaisons, sont des fonctions.

EXEMPLE

Soit l'opération arithmétique suivante : $D = (A+B).C$. La spécification de la fonction nécessite d'abord de préciser son nom (F1 dans l'exemple) et le type de données en entrée et en sortie (Figure 8.10, BOOL pour booléenne et INT pour entier signé). Puis des variables cohérentes avec cette typologie sont affectées en entrées et en sortie (F1_EN, A, B et C, en entrée; F1_OK et D en sortie) et le traitement permettant d'obtenir la sortie à partir des entrées est défini (Figure 8.11).

Le langage de programmation graphique utilisé dans ce cas est le FBD.

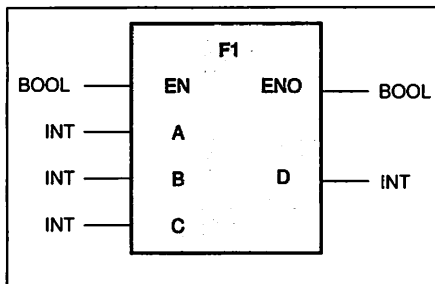


Figure 8.10. Spécification externe de la fonction F1.

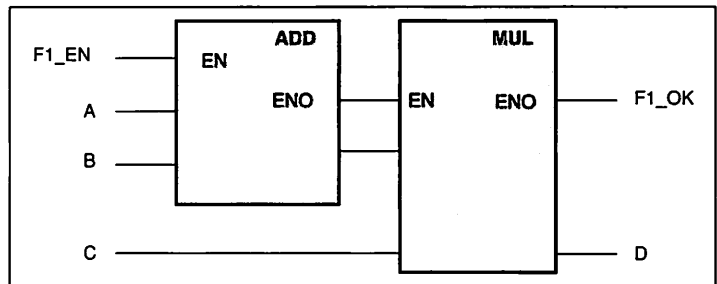


Figure 8.11. Spécification interne du corps de la fonction F1.

Les entrées A, B et C et la sortie D sont des entiers signés. La fonction ADD permet de réaliser une addition entre A et B, le résultat de cette opération arithmétique est multiplié à C grâce à la fonction MUL. L'entrée logique F1_EN valide l'exécution de l'addition, l'addition entre A et B effectuée la sortie ENO passe à l'état 1. La multiplication entre ce résultat et C peut alors s'exécuter (entrée EN vraie). La sortie F1_OK passe à l'état 1 et le résultat D est obtenu simultanément.

b) Le bloc fonctionnel

Le bloc fonctionnel est un module logiciel :
 - ayant plusieurs variables de sortie possibles ;
 - une mémoire interne.

Les mémoires, les détecteurs de front montant et descendant, les compteurs et les temporisateurs sont des blocs fonctionnels.

EXEMPLE

Mémoire à mise à 1 prioritaire

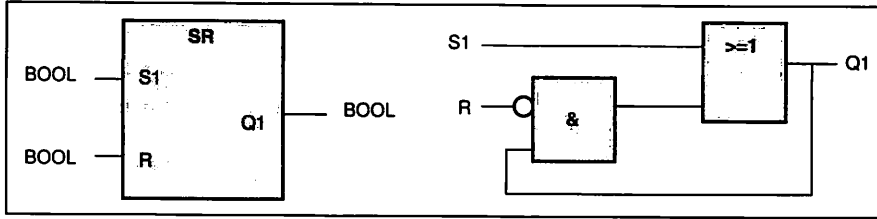


Figure 8.12. Déclaration d'un bloc fonctionnel mémoire en langage FBD.

La mémorisation est obtenue par auto-maintien de la variable Q1 (chapitre 6).

Ce type de bloc fonctionnel est un bloc fonctionnel standard (tout comme les opérateurs retard, les compteurs, ...) fourni par les constructeurs avec les logiciels de programmation des API.

● **Blocs fonctionnels utilisateurs**

Il est possible également de développer ses propres blocs fonctionnels appelés blocs fonctionnels utilisateurs et réutilisables dans des applications clients similaires.

Dans l'exemple figure 8.13 et figure 8.14, le bloc fonctionnel utilisateur « BF commande et alarme » est construit à l'aide de fonctions (ET, OU, NON) et de blocs fonctionnels (TON temporisation avec retard à l'enclenchement et SR opérateur mémoire). Dans cette application, il est utilisé sous le nom « Consigne de commande CMD et alarme », ce qui permet de le distinguer d'une autre utilisation éventuelle.

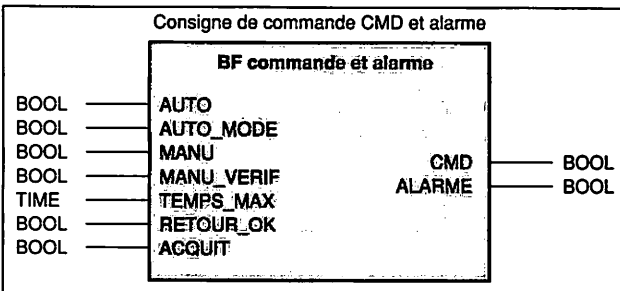


Figure 8.13. Description externe du bloc fonctionnel utilisateur.

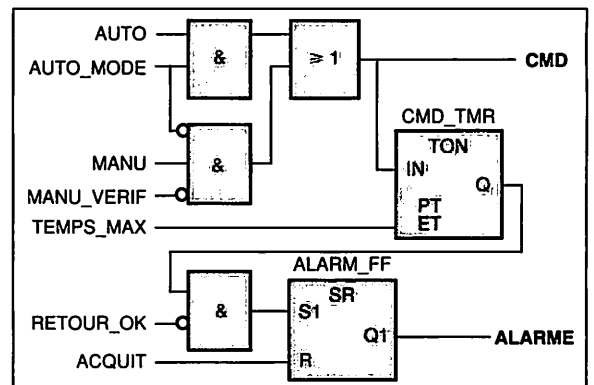


Figure 8.14. Description interne du bloc fonctionnel utilisateur.

c) Le programme

Le programme est un module logiciel construit à l'aide :
 - de fonctions ;
 - et de blocs fonctionnels.

« Programme » peut désigner le programme complet de l'application dans les cas les plus simples, ou une partie du programme spécifique à un traitement particulier (programmes pour cartes métiers, tâches rapides, communication, ...).