

AUTOMATISMES LOGIQUES

Modélisation & Commande

HAMDY HOCINE



VOLUME 1

structure et principes de fonctionnement

nouvelle édition

- Les éditions de l'université Constantine 1 -

- Avant Propos -

Cet ouvrage est destiné aux élèves des grandes écoles et universités, ainsi qu'à ceux des centres de formation professionnelle et des lycées techniques. Il peut être également utilisé avec profit par les techniciens et ingénieurs non spécialistes, souhaitant entreprendre une action de formation continue.

Il a pour but de leur donner les bases des « automatismes logiques », leur permettant d'analyser de manière fonctionnelle n'importe quelle installation automatisée, en vue de la compréhension de son fonctionnement, puis de son exploitation et de sa maintenance éventuelle.

Il leur donne également les outils pour aborder l'étude et la définition du cahier de charges de l'automatisation d'un procédé, sa réalisation (dans le cas de procédés simples) ou le suivi de sa réalisation (dans le cas de procédés complexes).

Si les sciences associées aux technologies occupent une place importante dans « l'enseignement technique », il est par contre difficile de les aborder. En effet n'étant ni un enseignement général ni un enseignement professionnel, il faut d'une part faire appel à la théorie (uniquement quand il le faut) en l'illustrant avec des exemples, donner d'autre part des bases scientifiques et technologiques sans entrer dans les travers du « livre de recettes ».

Ce que nous proposons n'est pas un cours classique d'automatique. Il a été conçu totalement à l'opposé, sur la base de la décomposition fonctionnelle des systèmes, avec un recours minimum à la théorie. Il exploite surtout le volet pratique en décrivant les outils nécessaires et la manière de les utiliser. Certains paragraphes qui ont un aspect purement descriptif sont certes assez rébarbatifs. C'est pour cela qu'au lieu de détailler technologiquement la constitution des composants, nous avons préféré nous limiter à un aspect purement fonctionnel, abusant parfois du système des « boîtes noires », tellement plus attractif à cause des dessins. Nous espérons ainsi toucher un large public, et que ce document serve de référence à consulter en cas de besoin.

Un autre objectif de notre démarche est de rassembler dans un document unique, tous les éléments qui touchent à un système automatisé : de sa décomposition en partie opérative et partie commande, sa modélisation par l'outil grafcet, la description des capteurs et actionneurs utilisés, en passant par les différents modes de marche et les procédés de commande, pour finir par les différentes techniques de programmation des automates programmables industriels.

Si nous avons fait la part belle à la technologie électropneumatique, au détriment de la technologie électronique, ceci est un choix délibéré. Car si dans ce dernier cas la documentation existe à profusion, ce n'est malheureusement pas le cas de la technologie électropneumatique, qui a jusqu'ici été complètement **occultée** dans l'enseignement universitaire et pour laquelle les références sont rares, et ce en dépit de son importance dans le monde professionnel.

Cet ouvrage est-il la traduction fidèle de nos intentions pédagogiques ? Nous le souhaitons vivement, et remercions par avance toutes les personnes qui auront l'amabilité de nous faire part de leurs critiques ou de leurs conseils, afin d'en améliorer le contenu dans les prochaines éditions.

Hamdi Hocine

Chargé du cours d'automatismes logiques, à l'Institut d'Electronique de l'Université de Constantine.

- AUTOMATISMES LOGIQUES : modélisation et commande -

VOLUME 1 : STRUCTURE ET PRINCIPES DE FONCTIONNEMENT

- 2° édition revue et augmentée -

- TABLE DES MATIERES -

- PAGES-

Chapitre 1 : STRUCTURE D'UN AUTOMATISME LOGIQUE	1
ET NOTION DE CAHIER DE CHARGES	
-I- Notions sur l'automatisation et les automatismes	02
-II- Structure d'un automate logique	05
-III- Modes de marche d'un système automatisé	11
-IV- Approche du cahier de charges de la partie commande	14
-V- Exemple d'application : la machine à laver	16
Chapitre 2 : LE GRAFCET	21
-I- Introduction	22
-II- Eléments du grafcet	23
-III- Règles d'évolution	26
-IV- Représentation des séquences multiples	28
-V- Compléments introduits par la norme CEI-848	37
Chapitre 3 : ELEMENTS DE TECHNOLOGIE	41
-I- Rappel sur la structure d'un automate dans les technologies concurrentes	42
-II- Actionneur pneumatique : le vérin	43
-III- Préactionneur pour actionneur pneumatique: le distributeur	49
-IV- Les interfaces	59
-V- La fonction mémoire	69
-VI- Organe de commande : le programmeur cyclique à cames	71
Chapitre 4 : COMMANDE EN LOGIQUE CABLEE	75
-I- Synthèse de la partie commande	76
-II- Matérialisation de grafcet par des séquenceurs	82
-III- Le séquenceur électronique à bascules	87
-IV- Le séquenceur électrique	93
-V- Le séquenceur pneumatique	104
Chapitre 5 : COMMANDE EN LOGIQUE PROGRAMMEE : L'API	110
A.P.I. : AUTOMATE PROGRAMMABLE INDUSTRIEL	
-I- Introduction	111
-II- Architecture	112
-III- Programmation	118
-IV- Programmation de grafcets à séquences multiples	147
-V- Câblage des automates TSX 17	156
REFERENCES BIBLIOGRAPHIQUES	159
ANNEXES :	160
-I- Alimentation et commande des automatismes électriques	160
-II- Lois générales d'électrotechnique	162
-III- Formulaire (mécanique, électricité)	163

CHAPITRE 1 : STRUCTURE D'UN AUTOMATISME ET NOTION DE CAHIER DE CHARGES

[Retour table des matieres](#)

-I- NOTIONS SUR L'AUTOMATISATION ET LES AUTOMATISMES	PAGE
-1-Objectifs de l'automatisation -----	2
-2-Fonctions des automatismes -----	3
-3-Technologies des automatismes -----	4
-II-STRUCTURE D'UN AUTOMATISME LOGIQUE	
-1-Parties opérative et commande d'un système automatisé -----	5
-2-Synthèse d'un automate séquentiel -----	6
-3-Fonctions de l'automate central -----	8
-4-Les périphériques -----	9
-5-Les interfaces -----	
-6-Structure d'un automate dans les technologies concurrentes -----	10
-III-MODES DE MARCHE D'UN SYSTEME AUTOMATISE	
-1-Rappels sur le fonctionnement séquentiel -----	11
-2-Marches normales ou automatiques -----	
-3-Marches d'essais et de maintenance -----	12
-4-Marches manuelles -----	
-IV-APPROCHE DU CAHIER DE CHARGES DE LA PARTIE COMMANDE	
-1-Spécifications fonctionnelles : niveau 1 -----	14
-2-Spécifications technologiques et opérationnelles : niveau 2 -----	
-3-Nécessité d'un outil de représentation -----	15
-V-EXEMPLE D'APPLICATION : LA MACHINE A LAVER	
-1-Description du fonctionnement de la partie opérative-----	16
-2-Spécifications fonctionnelles -----	
-3-Spécifications technologiques et opérationnelles -----	17
-4-Organe de commande -----	19
-5-Schéma de la machine à laver -----	20

REFERENCES BIBLIOGRAPHIQUES :

-1-Télémechanique, « Automatismes Industriels », méthodes-automatismes, notice technique NT-60 010-F-8N, édit La télémechanique, France, 1980

-2-G.Michel C.Laurgeau B.Espiau, « Les automates Programmables Industriels », chapitre 1, édit 1980, France

-I- NOTIONS SUR L'AUTOMATISATION ET LES AUTOMATISMES

L'automatisation d'un procédé (c'est à dire une machine, un ensemble de machines ou plus généralement un équipement industriel) consiste à en assurer la conduite par un dispositif technologique.

-1-Objectifs de l'automatisation

La compétition économique entraîne les industriels à vivre en permanence dans un esprit de concurrence, qui oblige à toujours améliorer les performances en termes de quantité et surtout de qualité (d'où le maître mot de l'économie : rapport qualité/prix). L'automatisation des processus industriels a pour finalité de réaliser ces vœux. Ses objectifs principaux au nombre de quatre sont complémentaires et liés. Ils peuvent s'énoncer ainsi :

-a- produire à qualité constante : contrairement à l'être humain, il est clair que la machine n'est pas sujette à la fatigue de fin de journée, par conséquent la qualité des produits sortant des chaînes est généralement la même ;

-b- fournir les quantités nécessaires : cette notion fait référence à l'adaptativité, c'est à dire pouvoir adapter l'offre à la demande. L'objectif est de produire juste les quantités nécessaires à un instant donné, de façon à tendre vers la notion de stock zéro. Pour pouvoir adapter l'offre à la demande, cela se fait rapidement et efficacement dans un environnement automatisé (arrêter par exemple une chaîne de fabrication en période de faible demande, ou au contraire en mettre en route d'autres pour répondre à la forte demande) ;

-c- augmenter la productivité : il s'agit donc d'augmenter le rendement. Pour ce faire l'automatisation a consisté à remplacer une grande partie des opérateurs humains par des machines, qui ont des cadences de travail élevées, ne connaissant ni les pauses café ni les congés payés ;

-d- améliorer les conditions de travail : il s'agit d'une part de remplacer l'homme par la machine pour les tâches pénibles ou qu'il ne peut pas faire (pour l'affecter ailleurs où il est censé faire un travail plus noble), d'autre part d'augmenter les possibilités de réaliser les objectifs "a", "b" et "c". En effet un employé qui mange bien, est bien soigné, et a de bonnes conditions de vie et de travail, n'est pas souvent malade, n'est pas fatigué, a peu d'absentéisme, et devient donc plus rentable économiquement.

-2-Fonctions des automatismes

Le degré d'automatisation d'un système varie selon la nature, la complexité, les objectifs assignés au projet. La surveillance d'une tour d'immeuble est différente de celle des ascenseurs qu'elle comporte ou de son dispositif de climatisation.

Il existe trois degrés d'automatisation ou modes de fonctionnement des automatismes :

-2-a-Mode surveillance

Dans ce mode l'automatisme a une fonction passive vis à vis du procédé qu'il pilote. L'organe de contrôle acquiert les informations et les analyse pour fournir journaux de bord et bilans. L'objectif est la connaissance technique et économique du procédé.

-2-b-Mode guide opérateur

Les traitements sont plus élaborés que dans le cas précédent, et l'automatisme propose des actions pour conduire le procédé selon un critère donné. L'automatisme ne réagit pas directement sur le procédé, il a donc un fonctionnement en boucle ouverte.

-2-c-Mode commande

L'automatisme a une structure en boucle fermée. On a une automatisation complète de certaines fonctions : acquisition des informations, leur traitement, et enfin l'action sur le procédé.

MODE FONCTIONNEMENT	ACQUI SITION	TRAITEMENT	ACTION	STRUCTURE
Surveillance	X			Boucle ouverte
Guide-Opérateur	X	X		Boucle ouverte
Commande	X	X	X	Boucle fermée

Figure 1.1 : différentes fonctions d'un automatisme

Les fonctions assurées dans chaque mode sont simples ou complexes selon le procédé ou la partie de procédé auxquelles elles sont assignées.

Prenons à titre d'exemple la surveillance d'une installation de chauffage central d'un édifice quelconque. Si le niveau d'eau diminue, la pression augmente. Au-delà d'une certaine pression la chaudière risque d'exploser. Pour faire baisser la pression il faut tout simplement rajouter de l'eau.

Dans le cas du mode surveillance, seul un indicateur visuel à aiguille nous permet de savoir que la pression a augmenté.

Dans le mode guide opérateur, on donne l'information sur l'action qu'il faut entreprendre, afin de baisser la pression. Un indicateur visuel ou sonore indique qu'il faut ouvrir la vanne d'eau.

Dans le mode commande, l'automatisme commande l'ouverture de la vanne, surveille le niveau d'eau, puis ferme la vanne quand le niveau désiré est atteint.

-3-Technologies des automatismes

L'automaticien dispose de nombreux outils technologiques pour réaliser l'organe de commande de son système, que l'on regroupe habituellement en deux catégories fondamentales : les solutions câblées et les solutions programmées (cf figure 1.2).

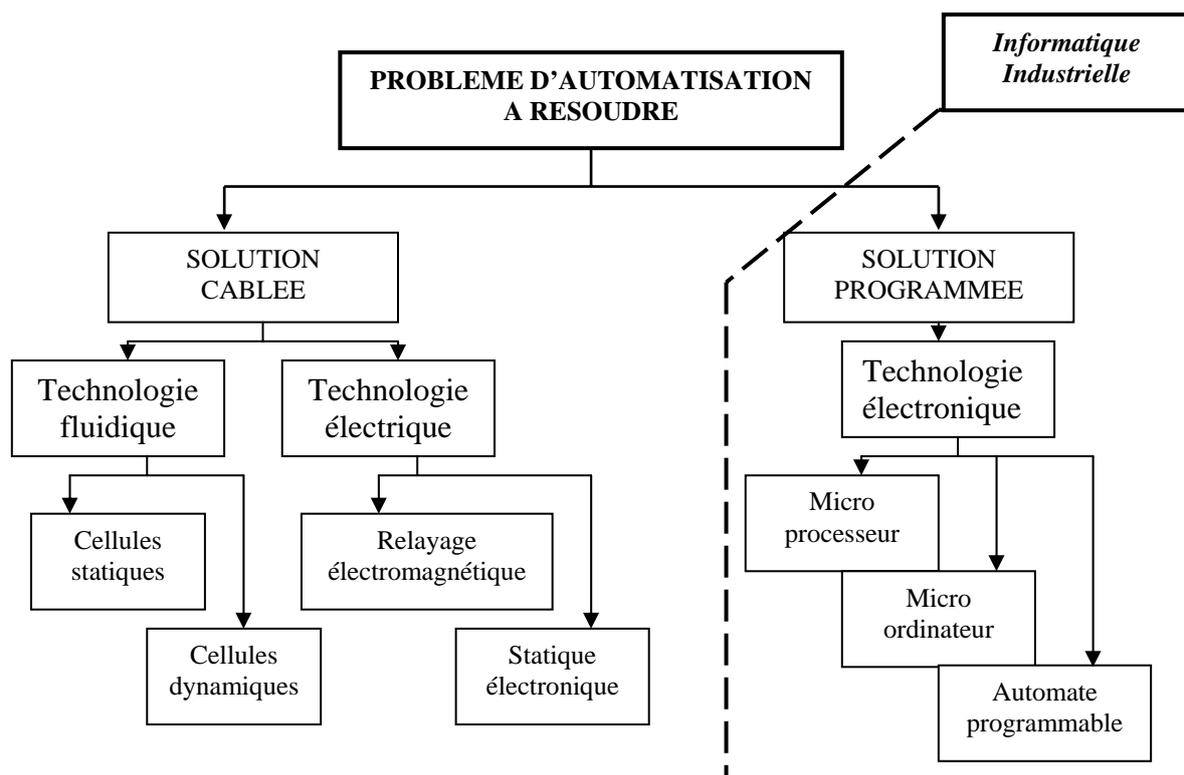


Figure 1.2 : principales solutions d'un problème d'automatisation

Remarques :

-1-L'informatique industrielle est une discipline conjuguant les théories de l'automatique et les moyens de l'informatique pour résoudre des problèmes de nature industrielle.

-2-Un automate programmable industriel ou API (cf chapitre 4) est une machine électronique, programmable par un personnel non informaticien, destinée à piloter en ambiance industrielle et en temps réel, des procédés logiques séquentiels.

-II-STRUCTURE D'UN AUTOMATISME LOGIQUE

-1-Parties opérative et commande d'un système automatisé

D'une façon tout à fait générale, un système automatisé peut se décomposer en deux parties qui coopèrent : une *partie opérative* ou puissance, et une *partie commande* ou automate ou automatisme (cf figure 1.3).

La partie opérative effectue des opérations (transformation de pièces brutes en pièces usinées dans le cas d'une machine outil à commande numérique, translation de la cabine d'un ascenseur de l'étage de départ à l'étage d'arrivée), lorsque l'ordre lui en est donné par la partie commande. En revanche elle fournit à la partie commande des informations sur les opérations effectuées.

La partie commande reçoit des consignes de l'extérieur (paramètres des opérations à effectuer) et fournit des comptes-rendus visuels ou sonores.

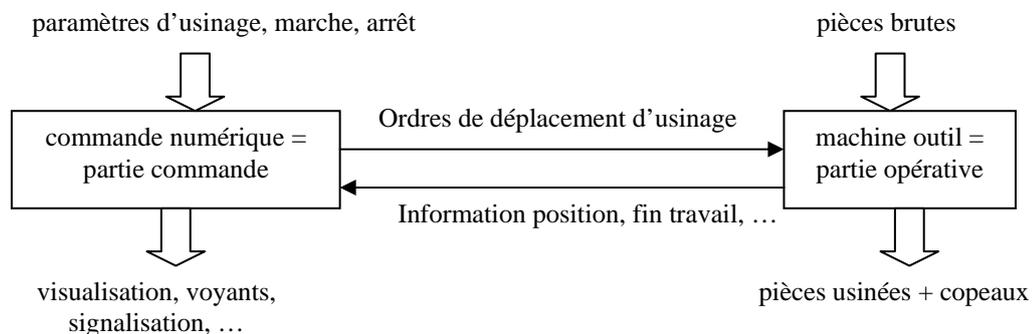


Figure 1.3-a : machine outil à commande numérique

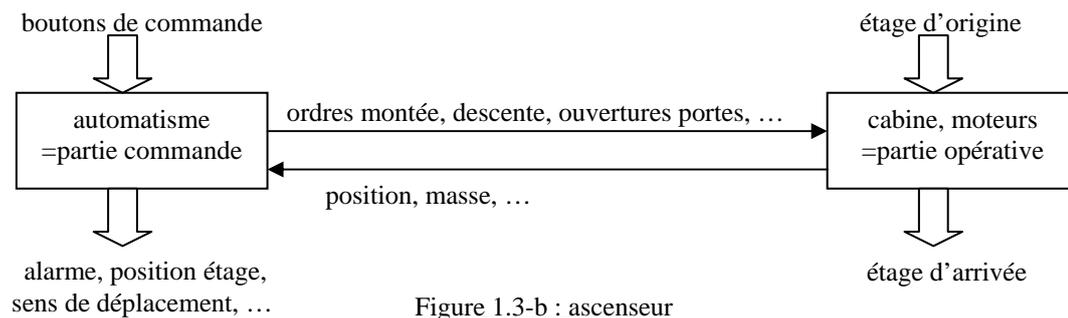


Figure 1.3-b : ascenseur

Figure 1.3 : exemples de systèmes automatisés

En conclusion la partie opérative est le processus physique à automatiser. La partie commande est un automatisme qui élabore en sortie des ordres destinés au processus, et des signaux de visualisation en fonction des comptes-rendus venant du processus et des consignes qu'il reçoit en entrée.

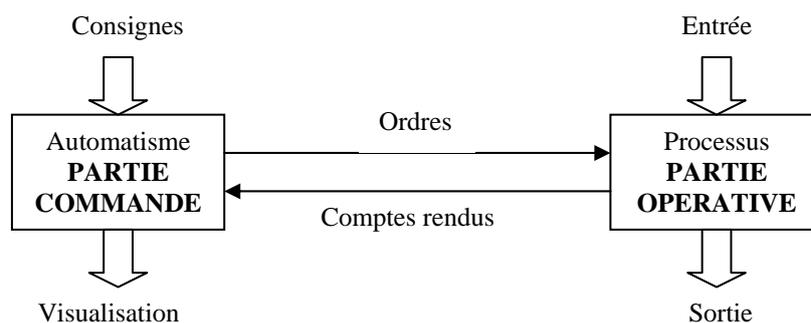


Figure 1.4 : schéma de principe d'un système automatisé

-2-Synthèse d'un automatisme séquentiel

-2-a-Rappel sur les logiques combinatoire et séquentielle

En logique combinatoire la sortie d'un système ne dépend que de la combinaison des informations d'entrée, et ceci quel que soit l'ordre d'arrivée de ces informations.

En logique séquentielle le système évolue en fonction des entrées et des états antérieurs du système.

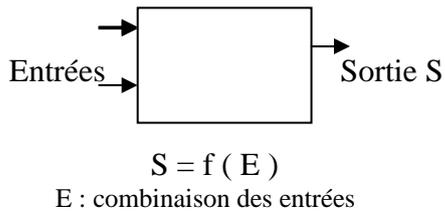


Figure 1.5-a : système combinatoire

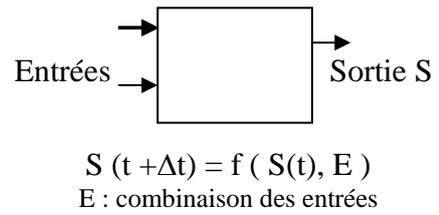


Figure 1.5-b : système séquentiel

Figure 1.5 : systèmes combinatoire et séquentiel purs

-2-b-Synthèse

La synthèse d'un automate séquentiel consiste à établir une relation entre les informations délivrées par les capteurs (entrées) d'une part, et les ordres donnés aux actionneurs (sorties) d'autre part, à travers un automate séquentiel autour duquel gravitent des fonctions spécifiques. La décomposition et le principe de fonctionnement sont généralement ceux donnés par la figure 1.6 .

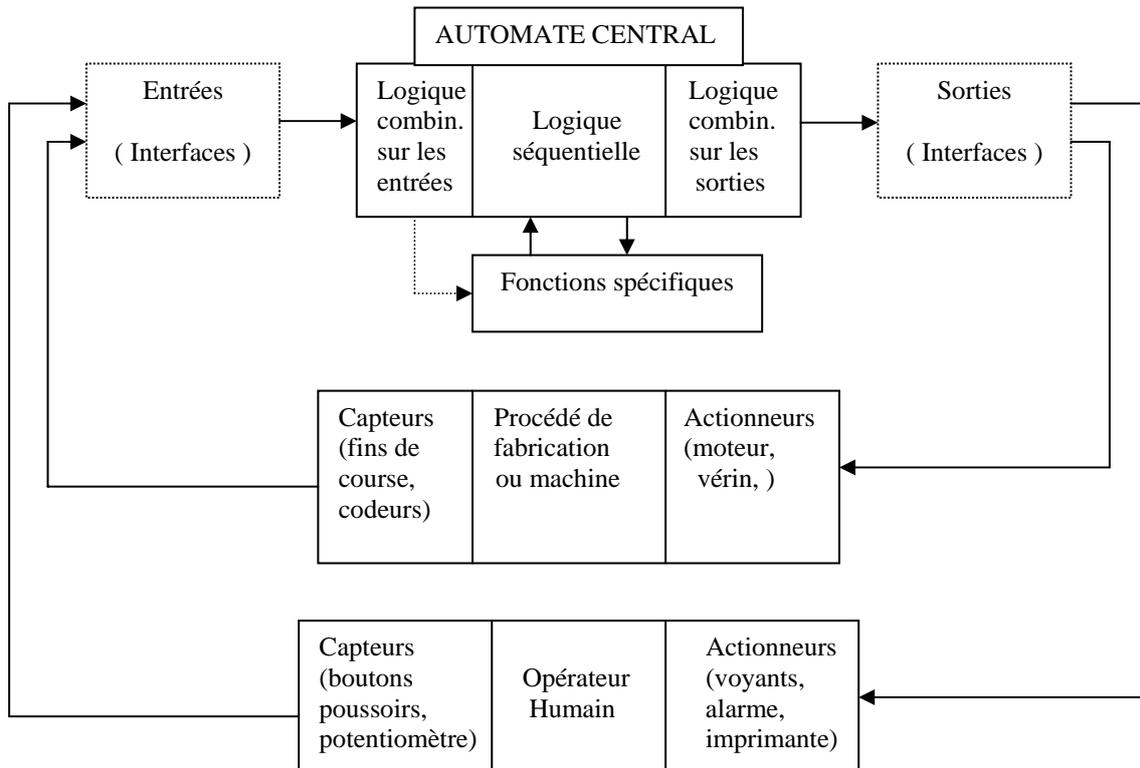


Figure 1.6 : structure d'un automate séquentiel

-3-L'automate central

-3-a-Structure

On désigne par automate ou automate central la partie centrale ou cerveau de l'automatisme, dont il constitue la structure séquentielle.

Il comprend une logique combinatoire, une logique séquentielle, et un certain nombre de périphériques qui assurent des fonctions spécifiques (comptage, temporisation). Ces périphériques dits « internes » peuvent être intégrés ou non à l'automate.

La logique combinatoire sur les entrées concerne les conditions de transition du système d'un état vers un autre. Elle se présente sous forme d'équations booléennes qui permettent de faire évoluer l'automatisme de manière séquentielle.

La logique combinatoire sur les sorties permet de faire la relation entre l'état de l'automate central et des sorties. Là aussi elle se présente sous forme d'équations booléennes en logique combinatoire.

Les périphériques internes sont des fonctions auxquelles l'automate central a recours pour élaborer des informations supplémentaires nécessaires à l'enchaînement des étapes (temporisations, comparateurs, calcul, etc...).

-3-b-Fonctions

Son fonctionnement est cyclique. Pour une phase élémentaire de fonctionnement séquentiel, ses fonctions principales peuvent se résumer de la manière suivante :

- *dialogue* avec l'opérateur dont il reçoit les ordres (début de cycle, arrêt, changement du mode de marche,...) et à qui il envoie des informations (alarme, visualisation d'état, bilan des entrées sorties, etc...),
- *acquisition* des informations fournies par les capteurs, et leur mise en forme par les interfaces d'entrée si nécessaire,
- *traitement* de ces informations,
- *élaboration des ordres puis commande* des actionneurs par l'intermédiaire des interfaces de sortie.

-4-Les périphériques

Ce sont des sous-ensembles de l'automatisme distincts de l'automate central, qui transmettent les informations et les ordres.

-4-a-Capteurs

-*Liés à la machine* : ils transmettent les informations à l'automate. Parmi les exemples simples on peut citer les fins de course, les codeurs, les capteurs à seuil de grandeurs analogiques.

-*Liés à l'opérateur* : ils transmettent les ordres de l'opérateur à l'automate. On peut citer les interrupteurs, les boutons poussoirs, les potentiomètres.

-4-b-Actionneurs

-*Liés à la machine* : ils transmettent des ordres à la partie mécanique de la machine. On peut citer les plus courants qui sont le moteur et le vérin.

-*Liés à l'opérateur* : ils transmettent des informations à l'opérateur. On peut citer les voyants, les alarmes sonores ou visuelles, l'écran, l'imprimante.

-4-c-Périphériques internes

Ils reçoivent des informations des capteurs et des ordres de la logique séquentielle et combinatoire. Ils transmettent des informations à cette même logique. Les exemples les plus courants sont le temporisateur, le compteur et le comparateur.

-5-Les interfaces

-5-a-Interfaces d'entrée

En plus de la protection de l'automate (rôle secondaire), ils assurent surtout un rôle d'adaptation (niveau de tension et courant) et de mise en forme (conversion analogique numérique) de l'information d'entrée.

-5-b-Interfaces de sortie

Comme les interfaces d'entrée ils assurent également une fonction de protection et d'isolation de la partie commande par rapport à la partie opérative. Mais ils servent principalement d'interface d'amplification.

Il est à remarquer que lorsque l'actionneur nécessite une interface de puissance, cette dernière est distincte de l'interface de sortie de l'automate. On l'appelle préactionneur (contacteur par exemple pour la commande de moteurs).

-6-Structure d'un automate dans les technologies concurrentes

Le tableau ci-dessous résume de manière succincte, en fonction des technologies employées, les différentes combinaisons des éléments intervenant dans la structure d'un système automatisé. Les détails sur le fonctionnement et les aspects technologiques seront traités au chapitre 3.

TYPE D'AUTO-MATISME	ELECTRIQUE	ELECTROPNEUMATIQUE OU ELECTROHYDRAULIQUE	TOUT PNEUMATIQUE
LOGIQUE DE COMMANDE	-Concevoir soi même le boîtier électronique de commande -Relais ou séquenceur électrique -API ou micro ordinateur -Programmateur à cames à E/S électriques	Idem à électrique	-Cellules, -Relais, -Séquenceurs (pneumatiques) -Programmateur à cames à E/S pneumatiques
CAPTEURS	Electriques	Electriques	Pneumatiques
PREACTION-NEURS	-Contacteur -Electroaimant, tyristor, transistor de puissance	Electrodistributeurs	Distributeurs à commande pneumatique
ACTION-NEURS	Moteur électrique	-Moteur hydraulique -Vérin pneumatique ou hydraulique	-Vérin pneumatique

-III-MODES DE MARCHE D'UN SYSTEME AUTOMATISE

-1-Rappels sur le fonctionnement séquentiel

Dans ce qui suit on s'intéresse aux automatismes logiques, pour lesquels les informations traitées ont un caractère « tout ou rien ».

De plus bien qu'on puisse réaliser des automatismes combinatoires purs, ils sont très souvent séquentiels et combinatoires à la fois, la partie séquentielle étant la partie la plus importante. En effet dans un processus de fabrication, les machines comportent au minimum un cycle, et presque toujours plusieurs cycles se déroulant simultanément ou successivement .

La décomposition du fonctionnement d'un système automatisé en étapes ou en phases repose sur la notion de chronologie, elle intègre donc de fait la notion de temps (aspect séquentiel).

-Etape : elle correspond à un **état stable** de la machine ou de tout le système. Dans une étape active, un ensemble d'**actions** élémentaires s'exécutent (montée chariot, rotation de broche etc...).

La transition entre étapes (c'est à dire le passage d'un état stable vers un autre état stable) ne s'effectue que si les conditions de transition liées à ce passage sont réalisées.

-Séquence : c'est un ensemble ordonné d'étapes. Par exemple on parlera de séquence de perçage d'une pièce, d'impression d'un ticket.

Dans une séquence seule une étape est active à la fois.

-Cycle : c'est un ensemble de séquences ordonnées.

On parlera par exemple du cycle de fonctionnement d'une machine à laver, qui comprend une séquence de préparation, une séquence de prélavage, une séquence de lavage, une séquence de rinçage, une séquence d'essorage et de vidange, et enfin une séquence de séchage éventuelle. Chaque séquence est constituée d'un ensemble d'étapes ordonnées. Par exemple la séquence de préparation comprend les étapes successives suivantes qui doivent se dérouler dans l'ordre : remplissage des bacs de détergent et assouplissant, remplissage de la cuve d'eau, chauffage de l'eau.

-2-Marches normales ou automatiques

-2-a-Cycle bouclé ou répété

L'opérateur donne uniquement l'ordre de départ cycle et l'ordre d'arrêt (l'arrêt s'effectue en fin de cycle).

-2-b-Cycle unique ou non bouclé

Le cycle s'exécute une seule fois (sans répétition) avec arrêt en fin de cycle. Il ne peut se répéter sans intervention extérieure.

-3-c-Cycle semi-automatique ou séquence par séquence

Une fois le départ cycle donné par l'opérateur, le passage d'une séquence à l'autre nécessite une nouvelle intervention de l'opérateur.

-3-Marches d'essai et de maintenance

-3-a-Cycle étape par étape avec exécution des actions

Le déroulement du cycle s'arrête à chaque étape. Au niveau d'une étape les actions prévues s'exécutent. Le passage d'une étape à l'autre nécessite en plus des conditions d'évolution, la pression sur le bouton « étape suivante ».

-3-b-Cycle étape par étape sans exécution des actions

Le fonctionnement est identique au précédent mais les actionneurs ne sont pas commandés. Le verrouillage s'effectue au niveau de la logique combinatoire sur les sorties.

-3-c-Cycle de réglage

Ce mode de fonctionnement cumule les modes cycle unique, étape par étape et marche manuelle asservie. L'opérateur présélectionne un arrêt sur une étape déterminée. Dans cette étape les actionneurs peuvent être commandés manuellement par l'opérateur au niveau de la logique combinatoire sur les sorties.

Sur action de l'opérateur et si les conditions d'évolution sont vérifiées, le cycle reprend en mode automatique jusqu'à la fin du cycle ou s'arrête à la prochaine étape présélectionnée pour le réglage.

-4-Marches manuelles

-4-a-Asservie

Tous les actionneurs peuvent être commandés par l'opérateur dans un ordre quelconque, « manuellement » au niveau de la logique combinatoire sur les sorties. Tous les asservissements subsistent comme an automatique.

-4-b-Non asservie

Mis à part les asservissements indispensables à la sécurité tous les autres sont déconnectés.

Le fonctionnement est identique au précédent mais en dehors de la logique combinatoire sur les sorties.

-4-c-En automatique

Ce mode permet en même temps les deux modes précédents avec un principe de fonctionnement automatique étape par étape. Au niveau de chaque étape les actionneurs peuvent être commandés soit directement (manuellement) soit par le biais de l'automatisme (logique combinatoire sur les sorties).

-4-d-Intervention

Ce cas se présente lorsque le cycle reste bloqué sur l'étape en cours, sans possibilité d'évolution. On peut utiliser le mode asservi ou non asservi pour commander les actionneurs et débloquent le système. Un fois la cause de blocage du système identifiée et éliminée, on repasse en marche d'essai pour s'assurer que le système fonctionne de manière fiable.

-IV-APPROCHE DU CAHIER DE CHARGES DE LA PARTIE COMMANDE

Le cahier de charges est une description claire du rôle et des performances de l'équipement à réaliser. Cette description se fait à deux niveaux successifs et complémentaires : la description fonctionnelle ou niveau 1 qui permet au concepteur de comprendre ce que l'automatisme doit faire ; la description technologique et opérative ou niveau 2 donne des précisions sur les conditions de fonctionnement des matériels.

-1-Spécifications fonctionnelles : niveau 1

Elles donnent le *rôle de la partie commande* à construire, donc définissent de façon claire et précise les différentes fonctions, informations et commandes impliquées dans l'automatisation de la partie opérative, *sans préjuger* en aucune façon *des technologies* qui seront *utilisées*.

Ni la nature, ni les caractéristiques techniques des différents capteurs ou actionneurs utilisés ne sont données dans ces spécifications. Par contre les sécurités de fonctionnement sont prévues dans les spécifications fonctionnelles, dans la mesure où elles ne dépendent pas de la technologie de ces capteurs ou actionneurs.

-2-Spécifications technologiques et opérationnelles

-2-a-Spécifications technologiques

Elles donnent la manière dont l'automatisme devra physiquement s'insérer dans l'ensemble que constitue le système automatisé et son environnement.

On donne ici les renseignements sur la nature exacte des capteurs et actionneurs employés, leurs caractéristiques et les contraintes qui peuvent en découler.

A ces spécifications d'interface s'ajoutent éventuellement des spécifications d'environnement de l'automatisme : température, humidité, poussière, alimentations etc...

-2-b-Spécifications opérationnelles

Elles ont trait au suivi de fonctionnement de l'automatisme : fiabilité (MTBF : temps moyen entre deux pannes), disponibilité (périodicité de la maintenance obligatoire), absence de pannes dangereuses, facilité de maintenance, dialogue homme machine, possibilités de modification de l'équipement en fonction des transformations de la partie opérative, etc...

Ces considérations, souvent négligées dans le cahier des charges, sont pourtant primordiales pour l'exploitant du processus à automatiser, parce qu'elles ont des répercussions économiques indéniables, tant sur le coût de l'équipement à réaliser que lors de l'exploitation.

Même si leurs répercussions économiques sont parfois difficiles à quantifier, les spécifications opérationnelles ont quand même une incidence certaine sur la manière de réaliser l'équipement

-3-Nécessité d'un outil de représentation

Pour décrire le fonctionnement d'un système en langage courant, les mots sont peu précis ou possèdent plusieurs sens. Cela est particulièrement vrai pour les mots du jargon technique : ils ont un sens dans un contexte précis, sont soit hermétiques au non initié, soit peuvent conduire à des contresens.

Le langage courant est peu adapté à la description précise des systèmes séquentiels, en particulier lorsqu'ils comportent un choix entre diverses évolutions possibles, et/ou des séquences à déroulement simultané. D'où l'intérêt de disposer d'un outil de représentation graphique d'un cahier de charges, qui soit normalisé, dépourvu d'ambiguïté et facile à comprendre et à utiliser. Deux outils principaux de description sont utilisés: les réseaux de Pétri et le grafcet (cf chapitre 2).

Le premier outil fut développé au début des années soixante par Pétri, et a donné lieu à beaucoup d'améliorations et d'extensions. Ses derniers développements sont surtout utilisés comme langage de spécification au niveau algorithmique et dans les protocoles de communication. Son principal atout réside dans sa formulation mathématique qui ouvre des horizons à des traitements informatiques.

Le grafcet est un outil de conception plus récente (créé par l'AFCEP et devenu norme française en 1982) qui a été normalisé au niveau international en 1988. Il est basé sur les principes mêmes du fonctionnement séquentiel (notions d'étape, de séquence et de cycle), et possède un nombre réduit de règles ce qui facilite son apprentissage. Il s'adapte très bien à la description et la modélisation du fonctionnement des automatismes, tant au niveau 1 qu'au niveau 2.

-V-EXEMPLE D'APPLICATION : LA MACHINE A LAVER

Dans cet exemple nous allons partir du travail normal d'une blanchisseuse, et automatiser certaines de ses fonctions pour arriver à la machine à laver telle que nous la connaissons aujourd'hui. Pour la simplicité et la clarté de l'exposé, nous ne nous intéressons ici qu'au lavage proprement dit et sa préparation. Il ne sera question ni de prélavage, ni de rinçage, ni d'essorage et encore moins de séchage.

-1-Description du fonctionnement de la partie opérative

Le fonctionnement s'effectue selon les phases ou étapes successives suivantes :

- 0°Phase de préparation : dépôt du linge dans la cuve, versement du détergent
- 1°Ouverture du robinet d'eau déversant l'eau directement dans la cuve
- 2°Fermeture du robinet d'eau dès le trop plein visuel
- 3°Chauffage de l'eau dans la cuve au bois ou au gaz
- 4°Arrêt de chauffage dès que la température désirée est estimée atteinte
- 5°Agitation manuelle dans les deux sens du linge et de l'eau (à l'aide d'un bâton)
- 6°Durée de l'agitation (du lavage) atteinte estimée par la blanchisseuse
- 7°Vidange manuelle de la cuve

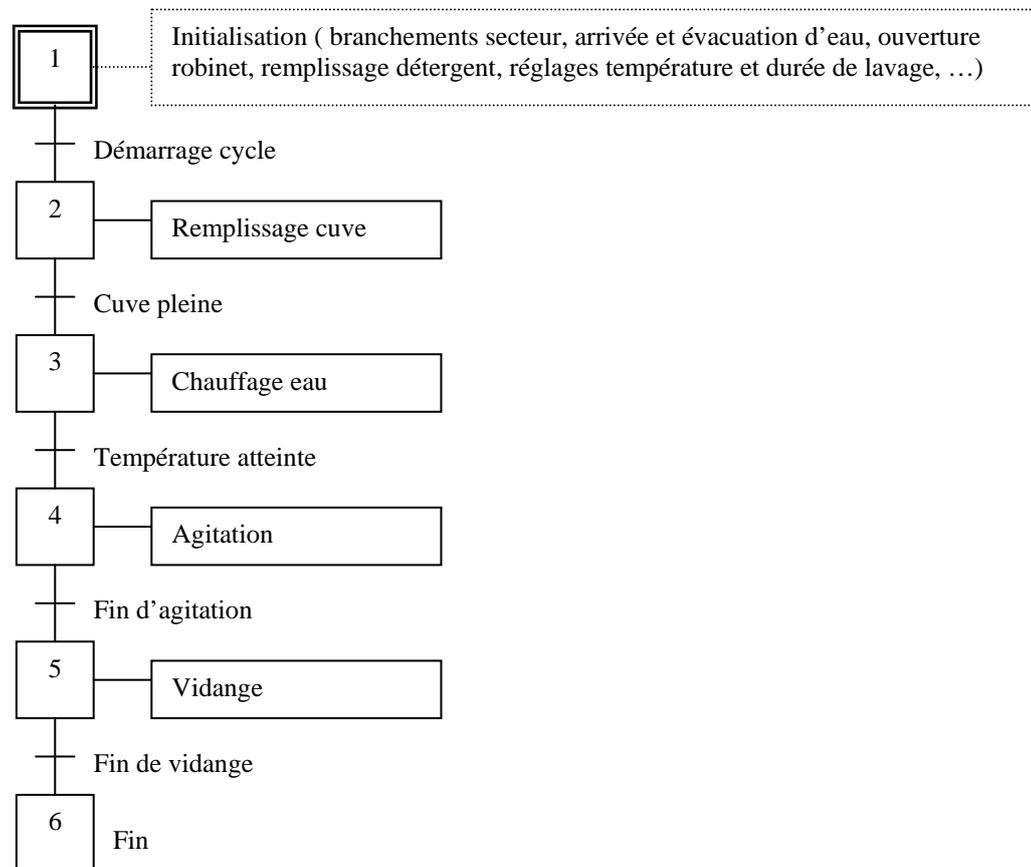
-2-Spécifications fonctionnelles

-2-a-Fonctions à automatiser

Mis à part la phase zéro qui est une étape d'initialisation et qui continuera à être manuelle, toutes les autres phases devront être automatisées. Ce sont les opérations de :

- 1° Ouverture et fermeture du robinet d'eau
- 2° Détection du trop plein
- 3° Chauffage de l'eau et arrêt du chauffage
- 4° Réglage et estimation automatiques de la température désirée
- 5° Agitation automatique dans les deux sens du linge et de l'eau
- 6° Réglage et estimation automatiques de la durée de l'agitation (du lavage)
- 7° Vidange automatique de la cuve

2-b-grafcet de niveau 1



-3-Spécifications technologiques et opérationnelles

-3-a-Contraintes imposées

- 1° Energie : 220 volts alternatif
- 2° Démarrage cycle par bouton poussoir à 2 positions marche/arrêt
- 3° Durées d'agitation : à droite une minute, à gauche deux minutes
- 4° Commande par programmeur à cames

-3-b-Solutions proposées

- 1° Ouverture et fermeture arrivée d'eau par électrovannes
- 2° Détection niveau d'eau par pressostat
- 3° Chauffage d'eau par thermoplongeur (résistance chauffante)
- 4° Détection de température d'eau par thermostat
- 5° Durées de lavage effectuées par temporisations mécaniques, pré-réglées par cames sur le programmeur
- 6° Détection de fin de vidange effectuée par le pressostat
- 7° Vidange par pompe
- 8° Agitation par un seul moteur à deux sens de marche, avec l'utilisation d'un tambour tournant à l'intérieur d'une cuve.

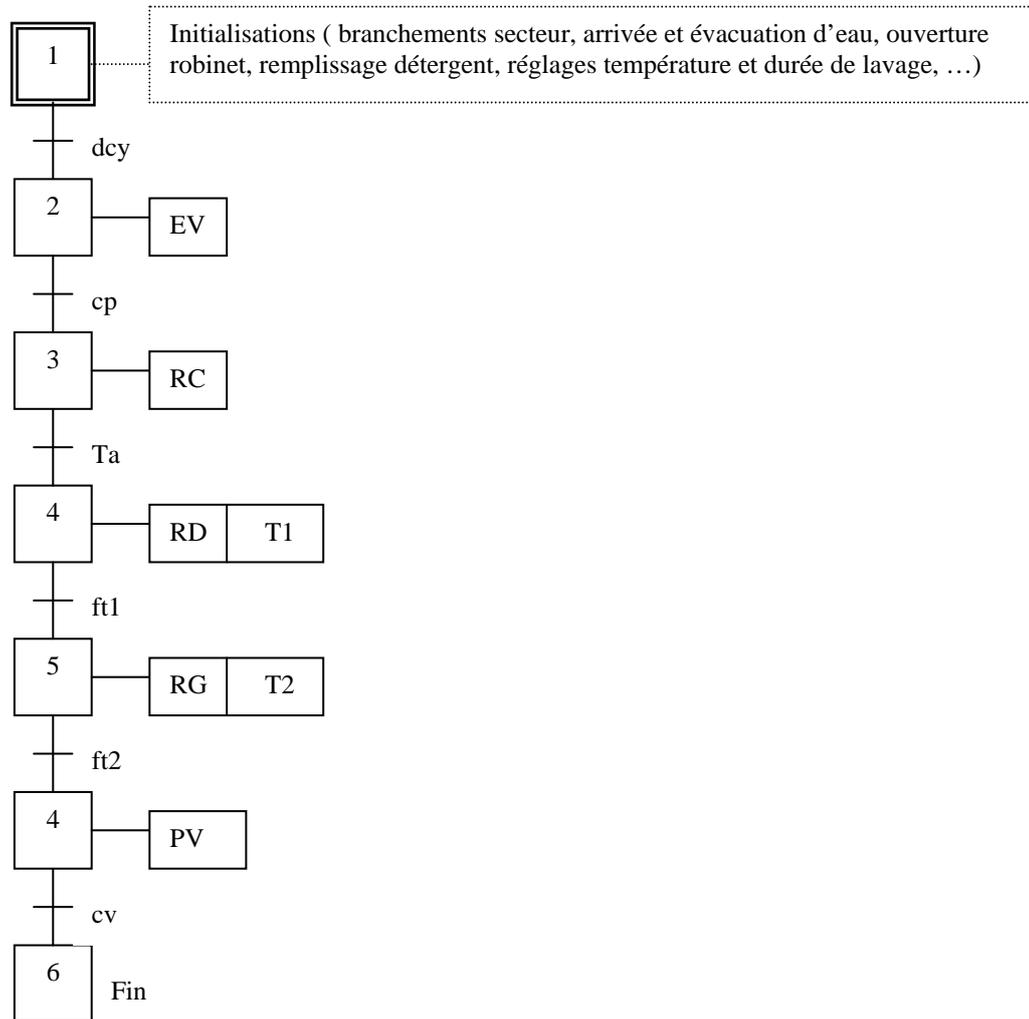
-3-c-Tableau des variables d'entrée/sortie

Tous les capteurs (délivrants des informations en entrée) ainsi que tous les actionneurs seront codifiés par des variables, rassemblées dans un tableau appelé tableau des variables d'entrée/sortie. On utilise généralement des minuscules pour les entrées et des majuscules pour les sorties.

Variables d'entrée et signification	Variables de sortie et signification
dcy : bouton de démarrage cycle	EV : commande ouverture électrovanne
cp : cuve d'eau pleine	MD : rotation droite moteur
cv : cuve d'eau vide	MG : rotation gauche moteur
Ta : température désirée atteinte	PV : commande pompe vidange
	RC : commande résistance chauffante

Variables internes d'entrée	Variables internes de sortie
ft1 : fin de la temporisation 1	T1 : lancement temporisation 1 (1 minute)
ft2 : fin de la temporisation 2	T2 : lancement temporisation 2 (2 minutes)

-3-d-Grafset de niveau 2 de la commande



-4-Organe de commande

Pour le concepteur il reste à choisir (ou concevoir au besoin) l'organe de commande ainsi que les interfaces de puissance nécessaires, et cela en fonction des contraintes engendrées par les choix précédents et/ou les spécifications imposées par le cahier de charges.

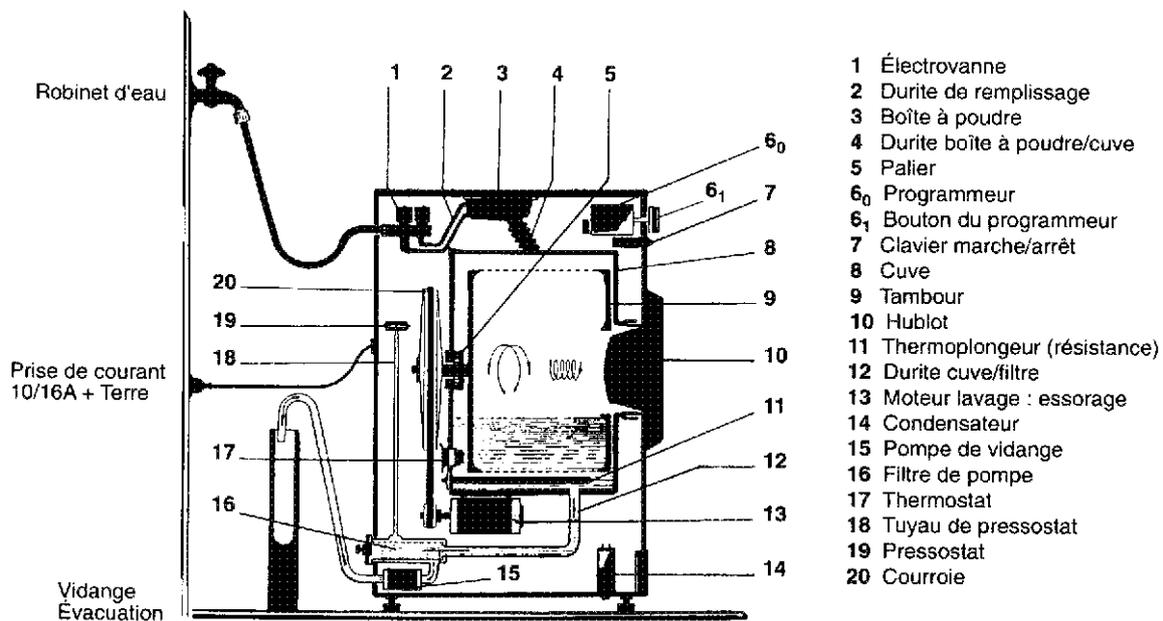
Compte tenu du choix du programmeur à cames comme organe de commande, il faut le dimensionner : temps pour faire un tour complet, déterminer le nombre de cames nécessaire, dimensionner les cames pour prendre en compte les durées de rotation de 1 et 2 minutes. Enfin connaissant la capacité de la cuve et le débit de la pompe, on peut en déduire la durée de vidange et prévoir pour cela une came supplémentaire.

Au niveau puissance, les seuls éléments nécessitant éventuellement une interface de puissance sont les deux moteurs. Pour la pompe de vidange le relais (ou le transformateur nécessaire) est intégré à la pompe. Pour le moteur de rotation on peut utiliser soit un relais soit une interface électronique (thyristor par exemple).

Si on veut utiliser une commande numérique (carte à microprocesseur) avec clavier de programmation, il faudra obligatoirement prévoir des interfaces de puissance, même si la commande des relais ne nécessite pas une grande puissance.

-5-Schéma de base de la machine à laver

[Haut de page](#)



Machine à laver à chargement frontal. Principaux éléments (doc. MERENTIER)

CHAPITRE 2 : LE GRAFCET

-I-INTRODUCTION	PAGE
-1-Rappel historique -----	22
-2-Généralités -----	
-II-ELEMENTS DU GRAFCET	
-1-Etape -----	23
-2-Transition et « réceptivité » -----	25
-3-Liaison orientée -----	25
-III-REGLES D'EVOLUTION	
-1-Règle 1 : initialisation -----	26
-2-Règle 2 : validation et franchissement d'une transition	
-3-Règle 3 : conséquences du franchissement d'une transition	
-4-Règle 4 : franchissements simultanés -----	27
-5-Règle 5 : activation et désactivation d'une même étape	
-IV-REPRESENTATION DES SEQUENCES MULTIPLES	
-1-Aiguillages -----	28
-2-Séquences simultanées -----	32
-3-Tâche ou séquence répétée ou sous programme -----	33
-4-Synchronisation entre plusieurs cycles-----	34
-V-COMPLEMENTS INTRODITS PAR LA NORME CEI-848	
-1-Hiérarchie de grafcets -----	36
-2-Modifications sur le graphisme -----	37
-3-Notion de macro étape ou représentation détaillée d'une étape -----	39
-VI-EXEMPLE D'APPLICATION -----	40

REFERENCES BIBLIOGRAPHIQUES :

-1-Norme française NF-C03-190 «Diagramme fonctionnel GRAFCET pour la description des systèmes logiques de commande», édit Afnor (Association Française de Normalisation), 1982

-2-Norme internationale IEC-848 «établissement des diagrammes fonctionnels pour systèmes de commande », édit CEI (Commission Electrotechnique Internationale), 1988

-3-R.Thomas « Les évolutions du grafcet et de sa normalisation », CETIM informations, n°113-décembre 1989, Cetim établissement de Saint-Etienne, France

-I-INTRODUCTION

-1-Rappel historique

Présenté par l'AFCEC en 1977, le grafcet, tout en s'implantant de façon très rapide dans l'industrie et dans l'enseignement, est resté quand même cantonné en France. Il a poursuivi son évolution grâce aux travaux des commissions de l'AFCEC, de l'ADEPA et des groupes de normalisation. Il a été normalisé en France en 1982 sous la référence NFC 03190. Ce n'est qu'en 1988 que l'IEC (Commission Electrotechnique Internationale) a publié la norme IEC 488.

Cette dernière reprend dans la forme et l'esprit l'essentiel de la norme française, tout en y apportant quelques modifications dans la forme et en y ajoutant quelques compléments. La terminologie « *diagramme fonctionnel* » a été conservée mais ce n'est pas le cas de l'appellation « *grafcet* » qui n'apparaît plus que dans la bibliographie française. Le terme de « réceptivité » disparaît au profit de « *condition de transition* », mais la notion de *graphe réceptif* est introduite (réceptif aux conditions de transition et aux conditions appliquées sur les actions). La *macro-représentation* disparaît au profit de la *représentation détaillée* d'une étape.

Si les termes introduits dans la norme française ne sont pas significatifs en anglais, par contre exprimés en français l'abréviation « grafcet » ainsi que les termes « réceptivité » et « macro-étape » sont très explicites, et permettent une assimilation rapide des concepts sous-jacents. C'est pourquoi nous continuerons à les utiliser bien qu'ils ne fassent pas partie de la norme internationale.

-2-Généralités

Le GRAFCET (Graphe de Commande Etapes Transitions) est un outil graphique de description du cahier de charges de la partie commande d'un système automatisé, utilisable au niveau 1 comme au niveau 2.

Le fonctionnement de l'automatisme peut être représenté graphiquement par un ensemble :

- d'*étapes* auxquelles sont associées des *actions*,
- de *transitions* auxquelles sont associées des *informations* (réceptivités),
- de *liaisons orientées* reliant les étapes aux transitions et les transitions aux étapes.

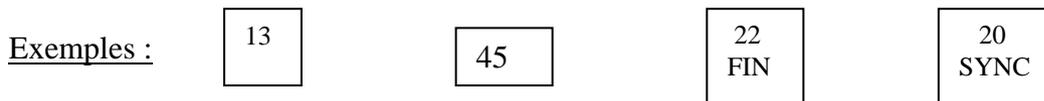
-II-ELEMENTS DU GRAFCET

-1-Etape

Elle correspond à une situation dans laquelle le comportement d'une partie ou de tout le système reste constant par rapport à ses entrées sorties. Une étape correspond donc à un état stable du système.

-1-a-Représentation

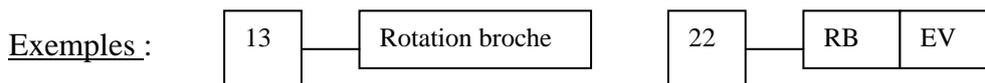
On représente graphiquement une étape par carré ou un rectangle avec un numéro à l'intérieur placé en haut ; on peut rajouter un nom symbolique en bas.



-1-b-Actions associées

Pour chaque étape on précise les actions à effectuer, caractéristiques de cette situation. Les actions sont représentées par des rectangles avec une description littérale ou symbolique.

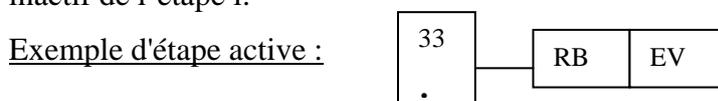
Ces actions ne sont effectives que lorsque l'étape est active.



-1-c-Etape active

Une étape peut être active ou inactive. On peut montrer une étape active à un instant précis en rajoutant le symbole point « . » en bas de l'étape. La situation du système automatisé à un instant donné est fournie par l'ensemble des étapes actives, donc celles pour lesquelles les actions associées s'exécutent.

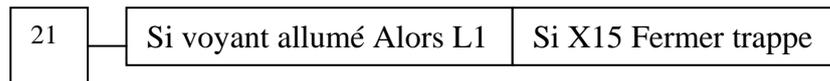
On note Xi (par exemple X33) la variable booléenne correspondant au caractère actif ou inactif de l'étape i.



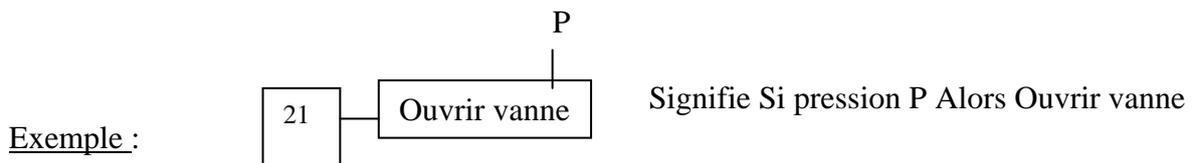
-1-d-Actions conditionnelles

L'exécution des actions peut être soumise à des conditions logiques, qui sont fonction des variables d'entrée ou de l'état actif ou inactif d'autres étapes. Par exemple l'expression « si X15 » signifie « si l'étape 15 est active » (c'est à dire $X15 = 1$).

Exemples :



On peut également noter une action conditionnelle en reliant la condition à l'action par un trait vertical disposé au-dessus de cette dernière.

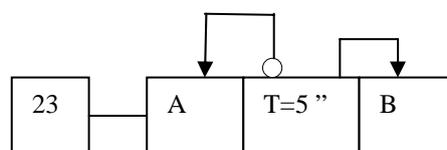


Lorsqu'il s'agit d'une variable interne qui est elle-même une action contrôlant d'autres actions au niveau d'une même étape, des liaisons orientées indiquent le sens de l'asservissement.

Par exemple si au niveau d'une étape 23 une temporisation contrôle deux actions A et B, et que l'on a le fonctionnement suivant lorsque l'étape 23 est active :

- temporisation de 5 secondes commandée systématiquement,
- pendant 5 secondes l'action A est validée et s'exécute, l'action B est inhibée,
- à l'expiration des 5 secondes, il y a déclenchement de la temporisation donc disponibilité de l'information fin de temporisation. Celle ci va stopper l'action A et autoriser l'action B.

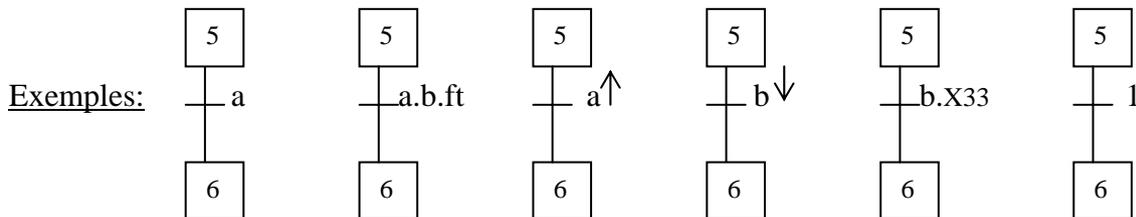
Le graphe décrivant ce fonctionnement sera alors le suivant :



-2-Transition et réceptivité

La transition indique la possibilité d'évolution d'une étape à une autre. Elle est représentée par un trait (généralement horizontal) perpendiculaire à la liaison orientée la reliant à l'étape.

- A chaque transition on associe une condition logique (appelée réceptivité) qui indique parmi toutes les informations disponibles, uniquement celles qui sont susceptibles à un instant donné de faire évoluer la partie commande d'une étape i vers une étape j, donc d'un état stable i vers un autre état stable j. Donc quand le système est dans l'état i, il est réceptif à cette information (il « écoute et l'attend ») et pas aux autres, d'où le terme de réceptivité pour désigner cette information particulière qui lui permet de changer d'état.
- Une réceptivité est une fonction combinatoire d'informations extérieures (états capteurs, consignes opérateur, état compteur ou temporisation), de variables auxiliaires ou de l'état actif ou inactif d'autres étapes.
- Une réceptivité toujours vraie est écrite égale à un.



-3-Liaisons orientées

Ce sont les voies d'évolution du grafcet. Deux liaisons orientées relient chaque transition à l'étape précédente et à l'étape suivante.

Les liaisons sont verticales (en général) ou horizontales. Quand le sens de la liaison n'est pas indiqué explicitement, le sens de la flèche est sous-entendu de haut en bas et de gauche à droite.

Lorsque plusieurs étapes sont reliées à une même transition, on représente le regroupement de plusieurs liaisons par deux traits parallèles.

-III-REGLES D'EVOLUTION

-1-Règle 1 : Initialisation

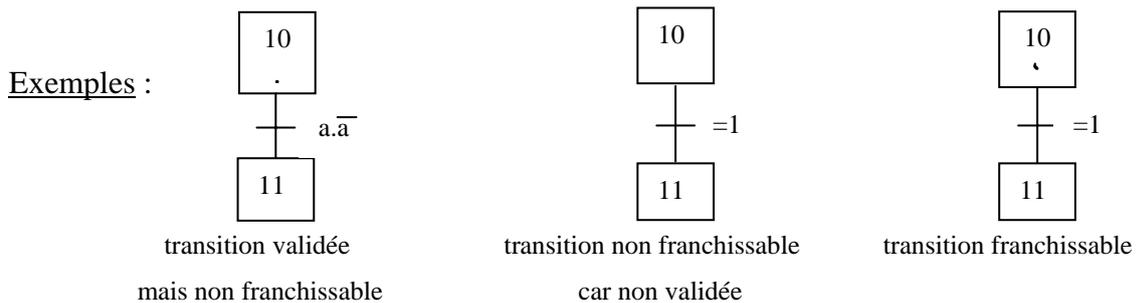
L'initialisation précise les étapes actives au début du fonctionnement. Elles sont activées inconditionnellement au début d'un fonctionnement cyclique. Après elles deviennent des étapes ordinaires, c'est à dire qu'elles sont activées et désactivées selon les mêmes règles que les autres étapes.

Elles sont repérées en doublant les côtés du carré représentant l'étape.



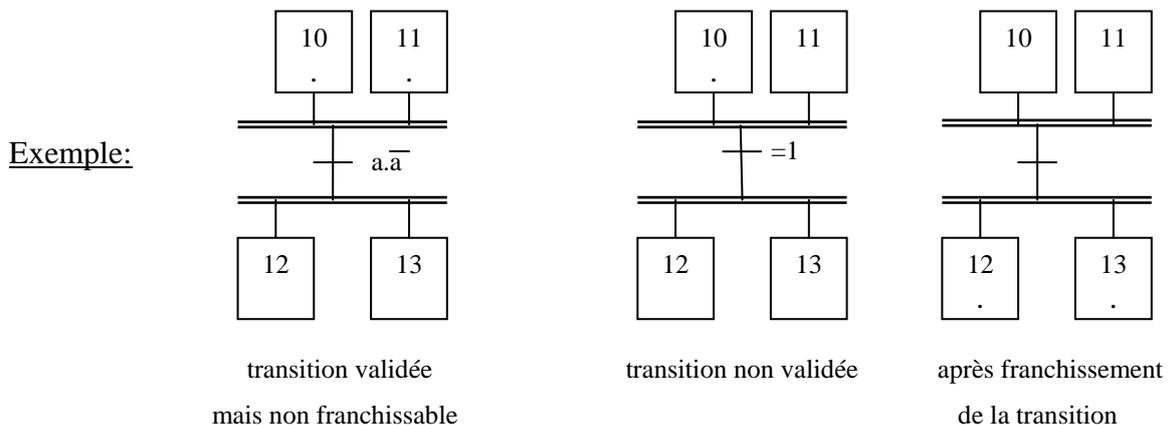
-2-Règle 2 : validation et franchissement d'une transition

Une transition est validée lorsque toutes les étapes immédiatement précédentes sont actives. Elle ne peut être franchie que si elle est validée et si la réceptivité associée est vraie. Elle est alors obligatoirement franchie.



-3-Règle 3 : conséquences du franchissement d'une transition

Le franchissement d'une transition entraîne l'activation de toutes les étapes immédiatement suivantes, et la désactivation de toutes les étapes immédiatement précédentes.



-4-Règle 4 : franchissements simultanés

Plusieurs transitions simultanément franchissables sont simultanément franchies (conséquence de la règle deux).

-5-Règle 5 : activation et désactivation d'une même étape

Si au cours du fonctionnement, une même étape doit être désactivée et activée en même temps, elle reste active.

Cette règle exprime en fait que le système ne possède pas d'état instable, et que la durée du passage d'un état stable vers un autre (franchissement d'une transition) est nulle. Il est à remarquer que dans la réalité (technologiquement parlant), cette durée de franchissement (ainsi que la durée d'activation d'une étape) ne peut être rigoureusement nulle, même si théoriquement (règles 3 et 4) elle peut être rendue aussi petite que l'on veut. Elle est fonction du temps de réponse des composants technologiques constituant l'automatisme.

-IV-REPRESENTATION DES SEQUENCES MULTIPLES

-1-Aiguillages

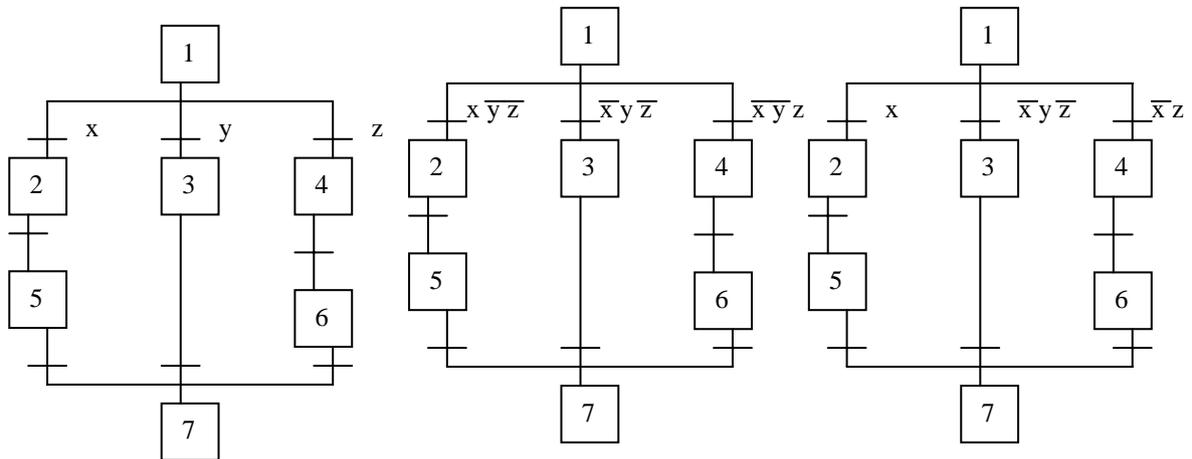
Une séquence est une suite d'étapes. Lorsque plusieurs séquences sont possibles, elle représentent différents fonctionnement possibles d'un même système. L'aiguillage correspond à une divergence en ou, et le retour au fonctionnement en une seule séquence (reprise en séquence) à une convergence en ou.

-1-a-Choix conditionnel entre plusieurs étapes

En conséquence de la règle 2, si les conditions de franchissement sont vérifiées, on peut rentrer en même temps dans les différentes séquences (cf figure 1-a si $x=y=z=1$), ce qui est une aberration car un système ne peut être simultanément dans deux états stables. Il faut donc rendre exclusives les réceptivités qui les commandent.

Si l'état logique 1 des différentes réceptivités correspond à un défaut de fonctionnement, l'exclusion aura pour conséquence de bloquer le système (cf figure 1-b si $x=y=z=1$), ce qui est normal en cas d'anomalie de fonctionnement.

Par contre si cet état est une possibilité de fonctionnement normal du système, il faut alors introduire une priorité dans les séquences (cf figure 1-c où l'ordre de priorité est: séquence 1, puis la 3 puis la 2).



1-a :aiguillage simple

1-b : séquences exclusives

1-c : séquences exclusives avec priorité

Figure 2.1 : aiguillages

-1-b-Saut d'étapes et reprise de séquence

Le saut d'étapes correspond à un morceau de séquence qui n'est pas effectué (sauté) si une certaine condition est vérifiée, puis le fonctionnement reprend normalement à partir d'une certaine étape. Ce genre de situation se rencontre souvent quand on a le choix entre deux (ou plusieurs) modes de fonctionnement, et que certaines opérations sont effectuées dans un mode et pas dans l'autre.

La reprise de séquence correspond au bouclage sur un morceau de séquence qui va se répéter tant qu'une certaine condition n'est pas vérifiée. Cette situation se rencontre souvent dans les opérations contrôlées par un compteur.

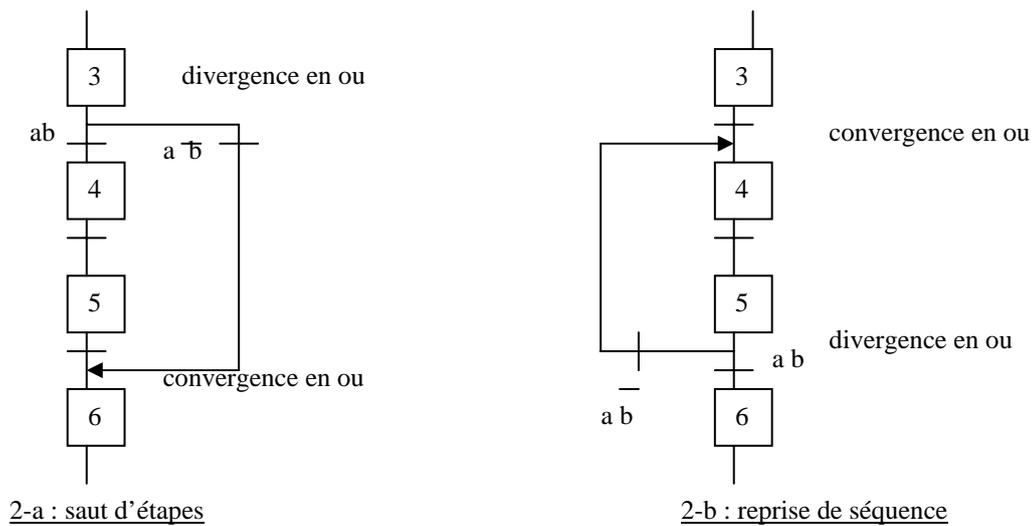
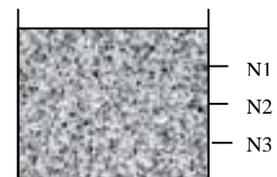


Figure 2.2 : saut d'étapes et reprise de séquence

1-c-Exemple d'aiguillage : permutation circulaire de pompes

Le niveau de liquide contenu dans un réservoir est contrôlé par 3 détecteurs N1, N2 et N3. L'alimentation de ce réservoir s'effectue par 3 pompes P1, P2 et P3. Chaque niveau découvert entraîne la mise en route d'une pompe.



Le nombre de pompes en service sera donc fonction du nombre de niveaux découverts. De plus afin d'équilibrer l'usure des pompes, celles-ci seront permutées à tour de rôle.

Le grafcet d'une telle commande peut être représenté ainsi :

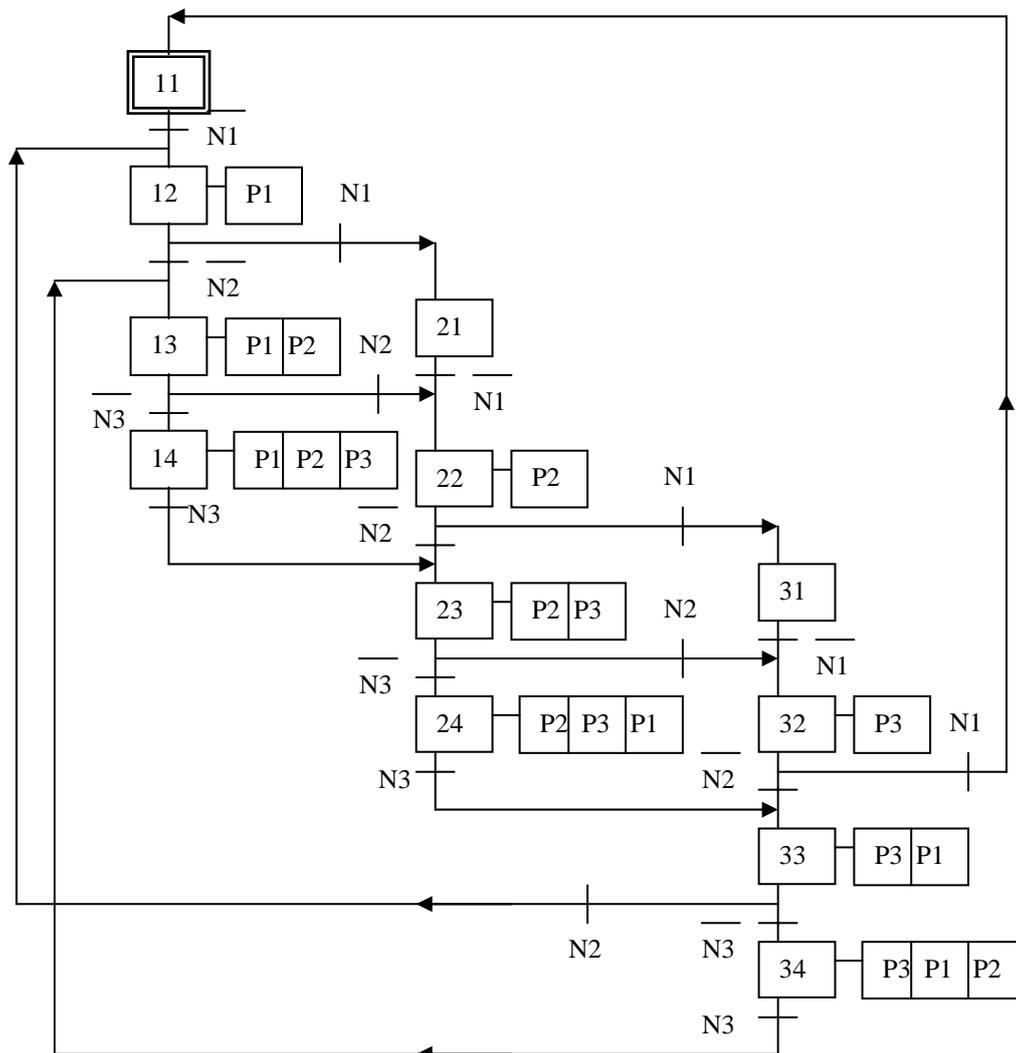
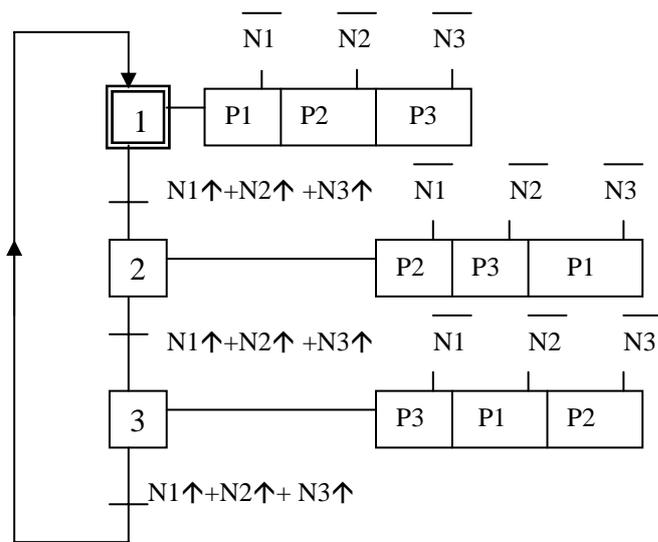


Figure 2.3 : graphe de la commande de la permutation circulaire de pompes

Remarques:

- Observons que le grafcet se décompose en trois parties (ou séquences) symétriques, dont les ordres correspondant à chaque séquence sont permutés circulairement.
- Dans une séquence déterminée (étapes 11, 12, 13, 14 par exemple), la progression dans les niveaux découverts correspond à la progression du nombre de pompes mises en action. Cette fonction est en fait combinatoire et pourrait aussi se représenter comme sur la figure 2.4 où chaque séquence sera représentée par une seule étape, et où on met également à profit la représentation graphique des actions conditionnelles.



Le passage d'une séquence à l'autre s'effectue dès qu'un niveau N1, N2 ou N3 est à nouveau recouvert. Comme ce niveau était découvert, cette information correspond à l'apparition d'un signal de recouvrement, ou à la somme logique des apparitions si on considère les trois niveaux.

Figure 2.4: représentation condensée du graphe de la figure 2.3

-2-Séquences simultanées

Un grafcet peut comporter plusieurs séquences simultanées, mais il faut que les *évolutions* des étapes actives dans chaque branche restent *indépendantes*. Le début des séquences simultanées est une divergence en et, et la fin une convergence en et.

Une transition unique (d'entrée) et deux traits parallèles indiquent le début des séquences simultanées. Elles correspondent en général au fonctionnement en parallèle de plusieurs machines qui coopèrent à la réalisation d'une tâche commune.

Une transition unique (de sortie) et deux traits parallèles indiquent également la fin des séquences simultanées. La condition de transition (réceptivité) n'est testée que si toutes les étapes de fins de séquences sont actives (conséquence de la règle 2). On parle alors d'attente réciproque vers une séquence commune. Pour illustrer cela on a pour habitude d'introduire une *étape d'attente* en fin de chaque séquence (cf figure 5-a), la réceptivité de la transition de sortie est alors mise à un. Si ce n'est pas le cas, par souci de clarté on peut noter entre parenthèses les informations particulières à chaque branche produites en fin de séquence ; et la réceptivité de la transition de sortie est le produit de ces informations (cf figure 5-b).

L'inconvénient quand on n'introduit pas d'étape d'attente, est que l'action associée à la dernière étape d'une branche continue (même si la branche a terminé son travail) jusqu'à ce que l'autre branche soit arrivée en fin de séquence. Cela n'est pas sans risque pour le matériel (échauffement d'un moteur de translation par exemple alors qu'on est en fin de course).

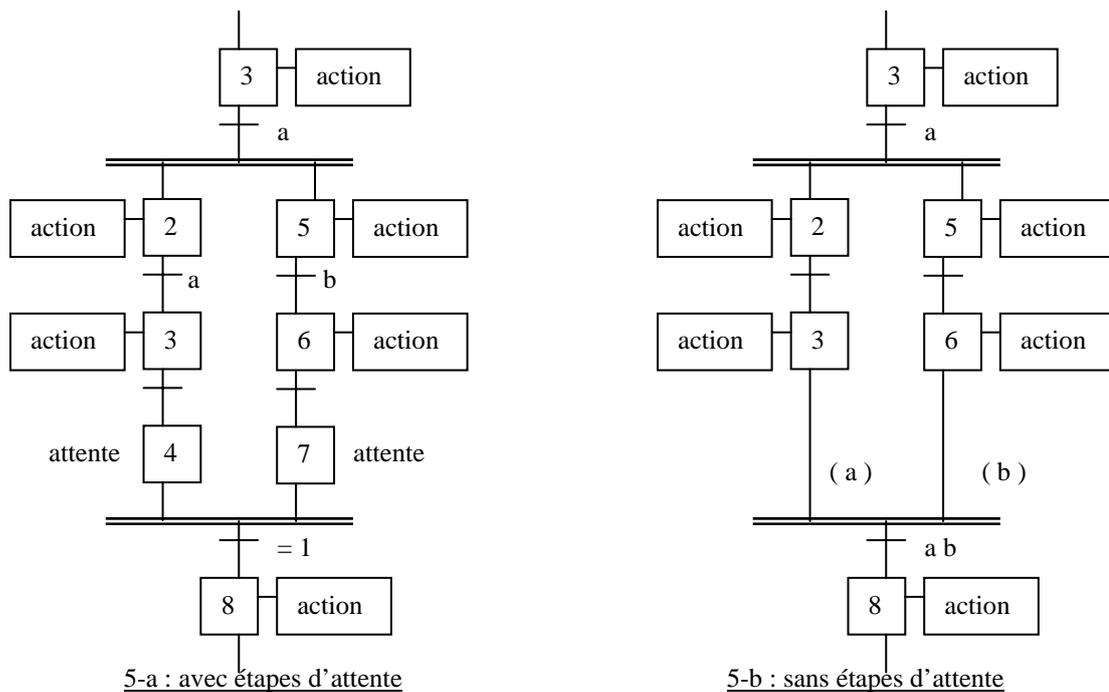


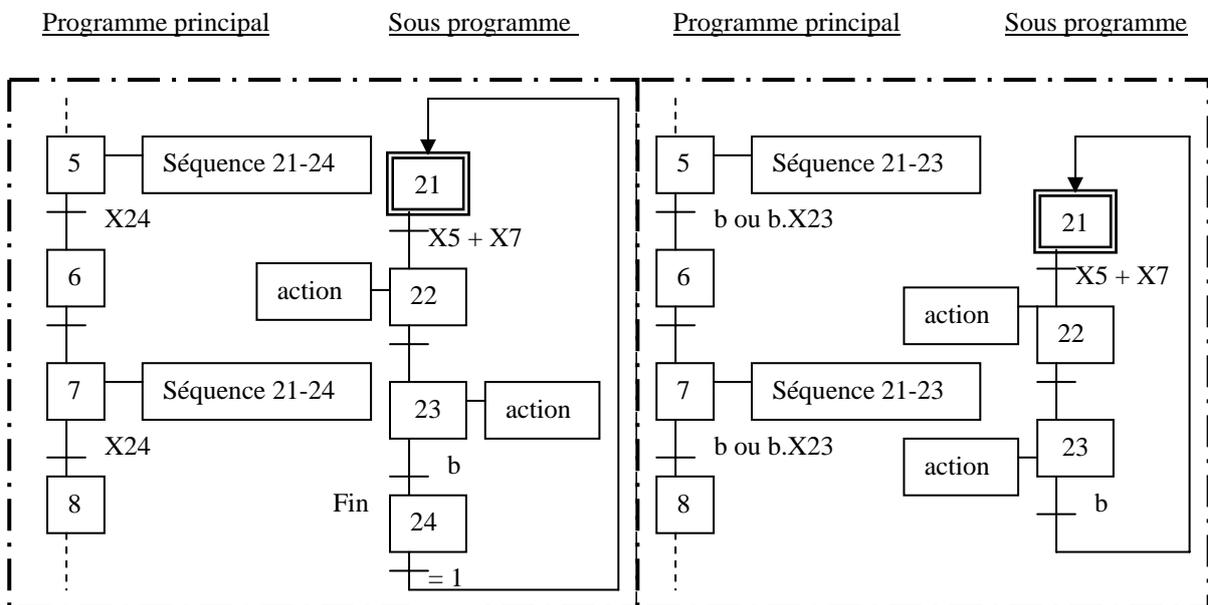
Figure 2.5 : séquences simultanées

-3-Tâche ou séquence répétée ou sous programme

Lorsqu’une séquence se répète plusieurs fois dans le cycle de fonctionnement, par souci de clarté et d’allègement du graphe, on l’extrait du grafcet pour en faire un *sous programme* (c’est à dire une *tâche* à part entière). On représente cette séquence par un grafcet séparé (graphe de la tâche) dont le fonctionnement est cyclique.

La première étape du sous programme est une étape initiale au même titre qu’une étape initiale ordinaire. L’entrée dans le sous programme se fait en introduisant comme conditions de franchissement de la première transition l’activité des étapes (du programme principal) d’appel du sous programme (cf figure 2.6). La fin du sous programme est marquée de préférence en introduisant une étape de fin, et le bouclage sur son étape initiale se fait par transition unitaire (cf figure 6-a).

L’appel du sous programme est symbolisé par une action associée à l’étape d’appel, dont la nature est décrite par le mot «séquence » suivi des numéros d’étapes de début et fin du sous programme. Le retour au programme principal se fait en introduisant dans la réceptivité de la transition (qui suit chaque étape d’appel du sous programme) l’activité de la dernière étape du sous programme (cf figure 6-a) et/ou l’information produite par la fin de ce dernier (cf fig. 6-b).



6-a : avec une étape de fin de la tâche

6-b : sans étape de fin de la tâche

Figure 2.6 : exemple de sous programme et représentations

-4-Synchronisation entre plusieurs cycles

Lorsque plusieurs cycles évoluent séparément, il est quelquefois nécessaire d’assurer des interverrouillages entre des séquences appartenant à des cycles différents, et ayant des fonctionnements simultanés incompatibles dans certaines parties des cycles (séquences exclusives), ou au contraire qui doivent se suivre séquentiellement et s’attendre mutuellement (séquences synchronisées).

-4-a-Séquences exclusives

Ce cas se produit lorsque les séquences «litigieuses » partagent une ressource commune, et doivent donc s’exclure mutuellement pour l’accès à cette ressource. On peut symboliser cela par la technique dite de la *boîte aux lettres* ou du *jeton* : une étape d’interverrouillage (symbolisant le jeton) est commune à l’entrée de toutes ces séquences, et reste activée (jeton disponible) tant que les séquences sont libres (cf figure 2.7). Cette étape est immédiatement désactivée (prise du jeton) dès que l’une d’entre elles est démarrée. La réactivation de cette étape commune ne sera effectuée (libération du jeton) que lorsque la séquence en cours sera complètement terminée. Les différents cycles conservent donc leur indépendance en dehors des séquences exclusives, chaque cycle pouvant être arrêté sans aucune conséquence pour les autres.

L’étape d’interverrouillage est une étape initiale, c’est à dire qu’elle doit être activée en même temps que les autres étapes initiales des cycles. Des étapes d’attente doivent être prévues dans chaque cycle.

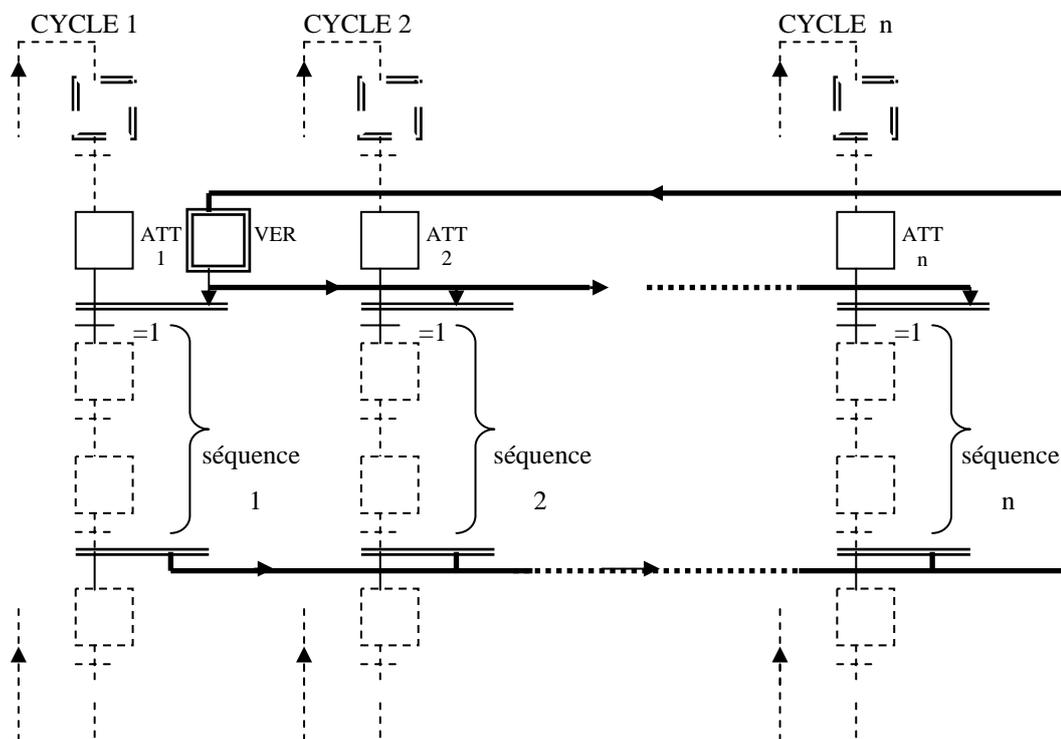


Figure 2.7 : synchronisation de cycles avec séquences exclusives

-4-b-Séquences synchronisées

Plusieurs séquences appartenant à des cycles différents peuvent devoir s'exécuter dans un ordre bien défini, tel que par exemple le partage d'une ressource avec interdiction d'accès à cette ressource deux fois de suite (c'est à dire tant que tous les autres cycles n'y ont pas accédé, comme dans le "polling" des structures "maître-esclave" dans les réseaux de télécommunication), ou bien chaque séquence a besoin du résultat de la séquence précédente pour effectuer sa tâche (postes de travail sur une chaîne de production en usine). Cela peut être schématisé par une technique de boîtes aux lettres multiples ou de "passage de témoin" (comme en athlétisme) : chaque cycle qui a terminé sa séquence exclusive passe le témoin à la séquence exclusive suivante.

On prévoit donc une étape d'attente dans chaque cycle ainsi qu'une étape de synchronisation. L'étape de synchronisation du premier cycle autorisé à accéder à cette ressource commune sera de type étape initiale.

Remarque :

Contrairement au cas des séquences exclusives où chaque cycle est indépendant de l'autre, dans les séquences synchronisées les cycles sont interdépendants : l'arrêt de l'un d'entre eux entraîne le blocage de tous les autres.

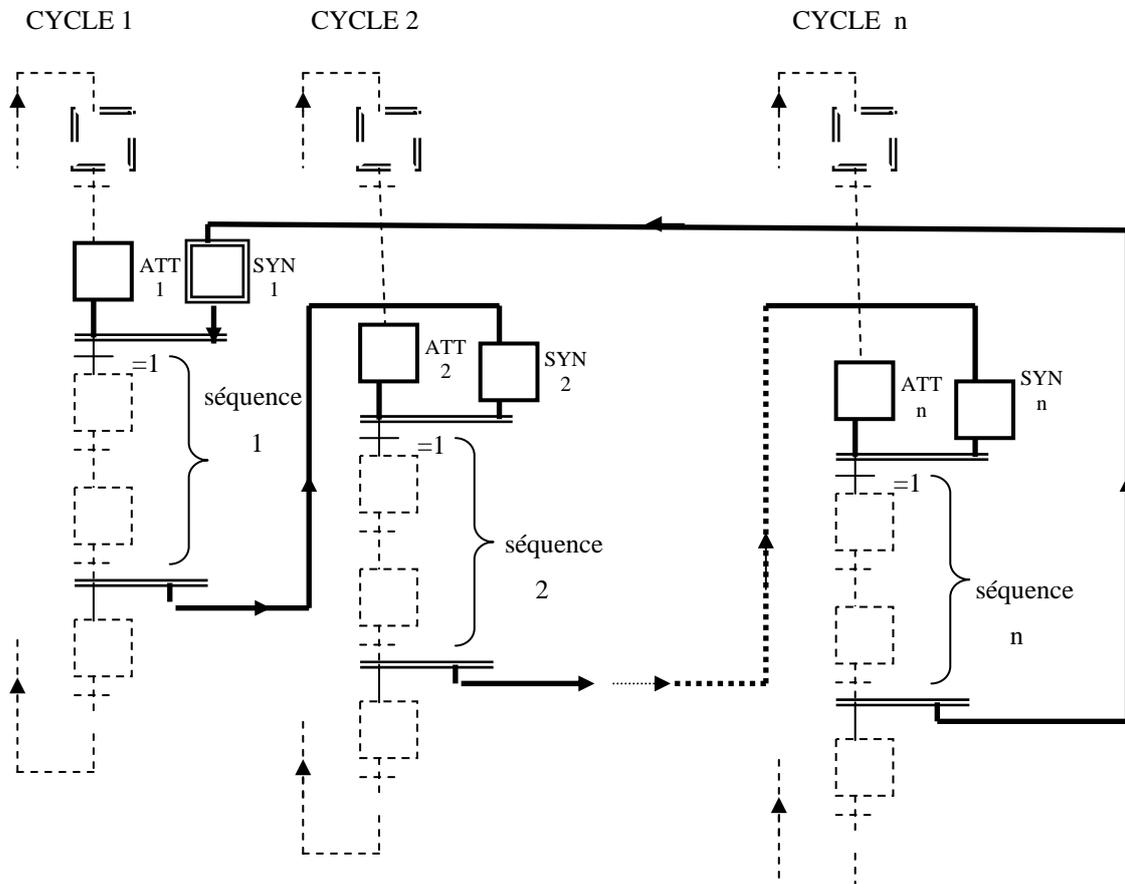


Figure 2.8 : synchronisation de cycles avec séquences synchronisées

-V-COMPLEMENTS INTRODUIITS PAR LA NORME CEI-848

Nous allons décrire dans ce qui suit des éléments du grafcet qui sont secondaires, c'est à dire non indispensables, et qui n'existaient pas dans la définition originelle du grafcet. Certains (parties 1 et 3) sont présents sous une autre forme dans la norme française, tandis que d'autres (partie 2) sont des ajouts introduits par la norme internationale.

Cette partie est un simple complément qui ne modifie en rien tout ce qui a été décrit jusqu'ici sur le grafcet.

-1-Hiérarchie de grafkets

L'évolution des commandes des processus vers des systèmes répartis, a conduit à proposer la notion de forçage. Cette notion sous-entend l'existence d'une hiérarchie dans les graphes, avec la notion de *maître* et d'*esclave*. On distingue en général dans la hiérarchie trois types de graphes : le *graphe de sûreté* (GS), le *graphe de commande* (GC), le *graphe de production* (GP) ou de *fonctionnement normal* (GFN).

Trois types de forçages ont été acceptés dans la norme CEI 848.

-1-a-Désactivation totale du ou des graphes esclaves : F/ Graphe_Esclave ()

Cette action met toutes les étapes du grafcet esclave à l'état inactif. Ce dernier ne pourra être réutilisé qu'après réactivation par le grafcet maître.

-1-b-Mise en situation initiale : F/ Graphe_esclave (I)

Cette action a pour effet d'activer uniquement les étapes initiales du grafcet esclave, et ce quelle que soit la situation antérieure du graphe (actif, inactif ou en fonctionnement).

-1-c-Mise en situation quelconque : F/ Graphe_esclave (Xi,Xj,...)

Le principe est le même que le précédent, mais les étapes activées ne sont pas des étapes initiales du grafcet esclave, mais des étapes dites initialisables. On les représente par un double trait dont la partie intérieure est en pointillés (X3 et X5 dans l'exemple de la figure 2.9).

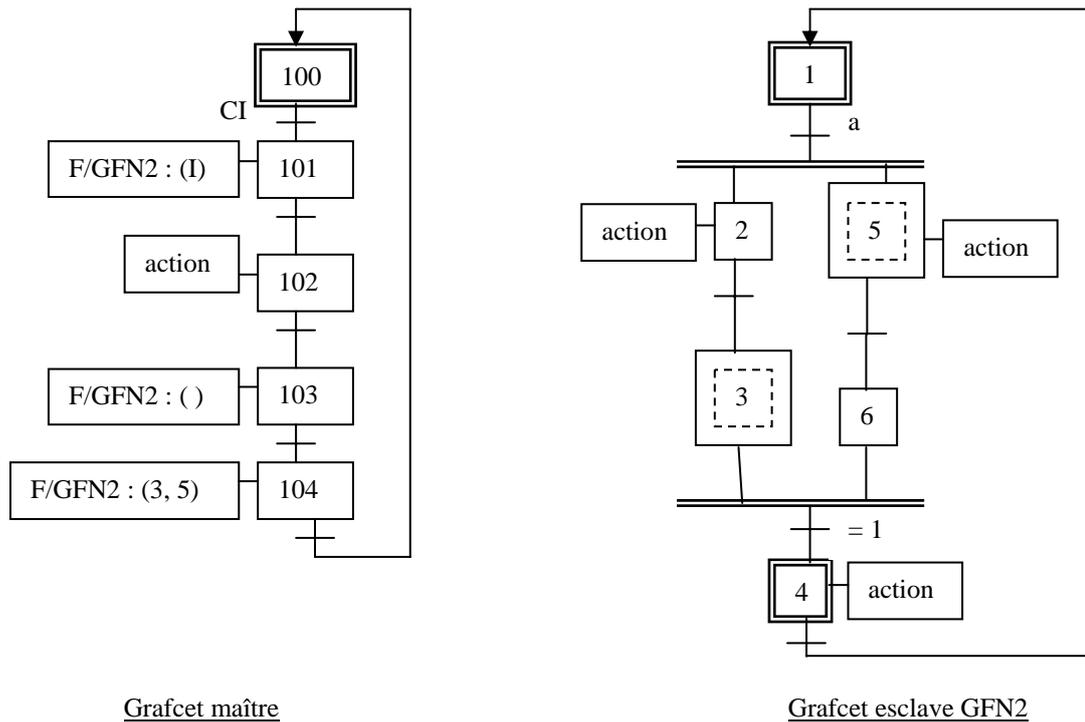


Figure 2.9 : grafquets hiérarchiques

2-Modifications sur le graphisme

Des modifications sur le graphisme ont été introduites pour se rapprocher des autres graphes déjà existant, par un rajout d'informations supplémentaires sur la nature des actions et des temporisations, l'innovation réellement notable étant une meilleure description de certains aspects séquentiels. Tout ceci bien entendu au prix d'une surcharge du graphe.

Les modifications importantes sont au nombre de trois :

-2-a-Durée de validité d'une information

Cette notion a été rajoutée pour tenir compte du temps de réponse des systèmes, et surtout pour tenir compte des systèmes de commande digitaux avec bus trois états (qui tendent à se généraliser), où une information présente sur le bus a une durée de vie limitée.

La notation $t1/a/t2$ de l'exemple figure 2.10 signifie que dès que l'information "a" est vraie, cette information n'est disponible qu'au bout d'un temps $t1$. De plus sa durée de validité est égale à $t2$.

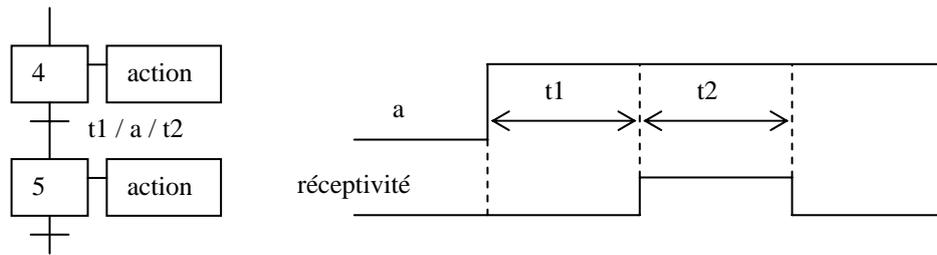


Figure 2.10 : limitation de la durée de validité d'une information

-2-b-Généralisation de l'information de transition

La condition de transition n'est plus limitée à des variables logiques, mais elle est généralisée à des schémas logiques ou logigrammes. Ainsi dans l'exemple de la figure 2.11 le logigramme et le schéma électrique produisent l'information $\overline{a(b+c)}$ qui est la réceptivité associée à la transition.

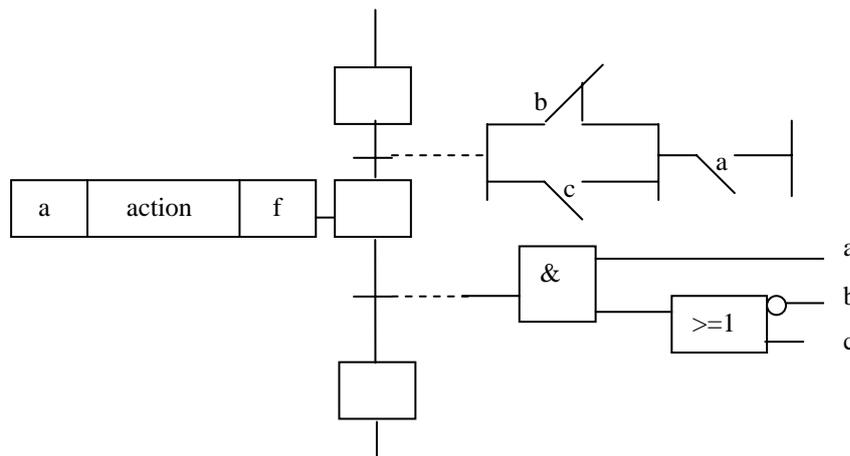


Figure 2.11 : extension de la représentation des réceptivités et description des actions

-2-c-Description séquentielle des actions

Le traitement des actions est remanié dans la forme : on rajoute une information de description du traitement binaire du signal d'activation de l'étape, ainsi qu'une référence du signal de fin d'exécution (respectivement « a » et « f » sur l'exemple de la figure 2.11).

Quatre types de traitements ont été retenus pour l'information « a », repérés par des lettres : S pour les actions mémorisées (Stored), D pour les actions retardées (Delayed), L pour les actions limitées dans le temps (Limited), C pour les actions conditionnelles (Conditional).

En cas de cumul des traitements, l'ordre d'écriture est important (par exemple SC est différent de CS).

-3-Notion de macro étape ou représentation détaillée d'une étape

Cette notion (présente dans la norme française) est différente de celle de sous programme. Il s'agit simplement de la représentation d'un ensemble d'actions qui se déroulent séquentiellement au niveau d'une même étape. Par souci de précision et de lisibilité, on éclate cette étape en plusieurs étapes et on parle alors de représentation détaillée d'une étape.

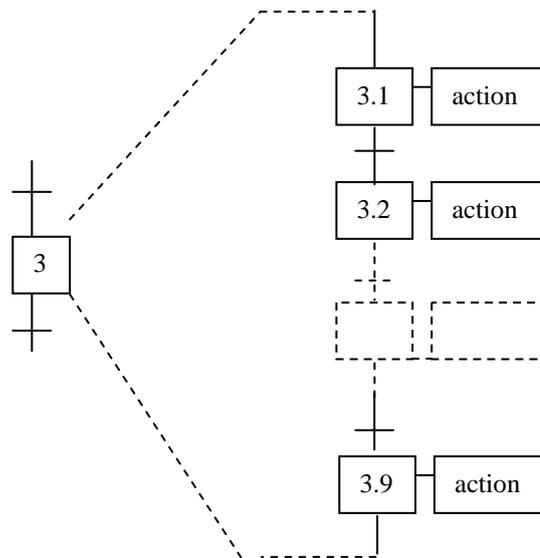


Figure 2.12 : macro étape

Remarque :

L'étape de sortie 3.9 participe à la validation de la transition de sortie de la macro étape 3.

-VI-EXEMPLE D'APPLICATION : Gestion d'une barrière de parking d'immeuble

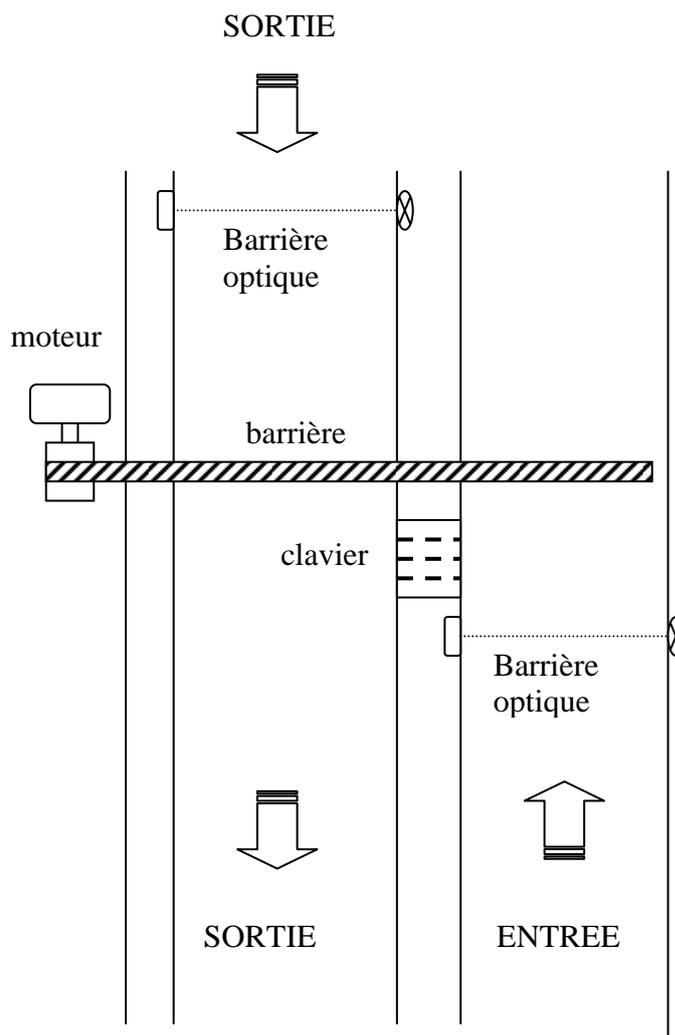
Deux barrières optiques contrôlent les accès à l'entrée (e) et à la sortie (s).

L'ouverture se fait soit après validation (v) du code confidentiel tapé au clavier par le conducteur entrant, soit après détection par la barrière optique (s) qu'un véhicule veut sortir.

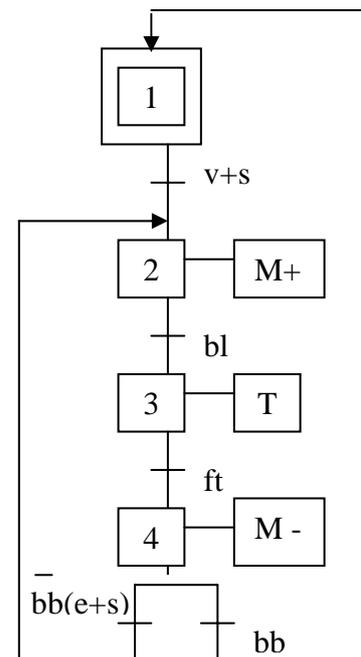
Une fois que la barrière est en position fin de course levée, la fermeture se fait au bout de 3 secondes (on utilise une temporisation externe). Pendant la descente de la barrière, si on détecte qu'un véhicule veut sortir (s) ou veut rentrer (e), la barrière se relève automatiquement.

La barrière est actionnée par un moteur à deux sens de marche, commandé par deux contacteurs : M+ (montée) et M- (descente). Les fins de courses sont détectées par 2 contacts : bl (barrière levée) et bb (barrière baissée).

Schéma de principe de l'installation



Grafcet modélisant la commande



CHAPITRE 3 : ELEMENTS DE TECHNOLOGIE

-I- RAPPEL SUR LA STRUCTURE D'UN AUTOMATISME	42
DANS LES TECHNOLOGIES CONCURRENTES	
-II- ACTIONNEUR PNEUMATIQUE : LE VERIN	
-1-Description	43
-2-Dimensionnement	44
-3-Capteurs de fin de course	46
-4-Différents types de vérins	47
-5-Exercice d'application	48
-III- PREACTIONNEUR POUR ACTIONNEUR PNEUMATIQUE: LE DISTRIBUTEUR	
-1-Moyens de pilotage ou de commande	51
-2-Symboles normalisés	
-3-Electrodistributeurs	52
-4-Auxiliaires de distribution	55
-5-Exemple d'application	57
-IV- LES INTERFACES	
-1-Interfaces modifiant les paramètres d'un signal (pneumatiques, électriques, électroniques)	59
-2-Interfaces modifiant la nature d'un signal (électrovanne, contact à pression, manostat & vacuostat, électrodistributeur)	67
-V- LA FONCTION MEMOIRE	
-1-Généralités	69
-2-Mémoire monostable (pneumatique, électrique)	
-3-Mémoire bistable (pneumatique, électrique, électronique)	70
-VI- ORGANE DE COMMANDE : LE PROGRAMMATEUR CYCLIQUE A CAMES	
-1-Principe	72
-2-Caractéristiques d'un programmeur à cames	73
-3-Exemple	74

REFERENCES BIBLIOGRAPHIQUES

- 1- M.Pinot & al « Du grafcet aux automates programmables », collection L.P., édit Foucher, Paris 1986
- 2- G.Boujat & J.P.Pesty « Automatismes », collection AGATI, série Bac Pro, édit Dunod, Paris 1993
- 3- F.Degoulange & al « Automatismes », classes de première et terminale E,F,BT- formation permanente, édit Dunod, Paris 1983.
- 4- Parker Pneumatic & Telepneumatic « Constituants pneumatiques », catalogue technique, juin 1995

-I- RAPPEL SUR LA STRUCTURE D'UN AUTOMATISME DANS LES TECHNOLOGIES CONCURRENTES

TECHNO-LOGIE	ACTIONNEUR	PREACTION-NEUR	CAPTEURS	LOGIQUE DE COMMANDE
Electrique	Moteur électrique (à CC, pas à pas, asynchrone...),	-Contacteur -Electroaimant (pour actionner électrovanne, serrure, serrage, levage, freinage...)	Electriques (fins de course, interrupteurs, boutons poussoirs, codeurs, potentiomètres etc...)	-Concevoir soi-même le boîtier de commande (électrique ou électronique), -Relais,-Séquenceurs, -API,-Ordinateur, -Programmateur à cames réglable à E/S électriques
Electro pneumatique ou Electro hydraulique	-Vérin pneumatique ou hydraulique -Moteur hydraulique	Electrodistributeur	Electriques	Idem à électrique (Pour le programmateur à cames, les E/S peuvent être électriques et/ou pneumatiques)
Tout pneumatique	Vérin pneumatique	Distributeur à commande pneumatique	Pneumatiques	-Cellules pneumatiques, -Relais & Séquenceurs pneumatiques, -Programmateur à cames réglable à E/S pneumatiques

Remarques

La logique de commande électrique a de plus en plus tendance à devenir une logique d'interfaçage, d'amplification et de sécurité. La logique de commande électronique est surtout utilisée pour le traitement de l'information à cause de sa faible consommation d'énergie, de son faible volume, de sa puissance de traitement, de la possibilité de programmation de certains composants et de son faible coût.

Excepté les préactionneurs les plus courants, nous avons volontairement occulté les interfaces dans ce tableau. Elles feront l'objet du paragraphe IV de ce même chapitre.

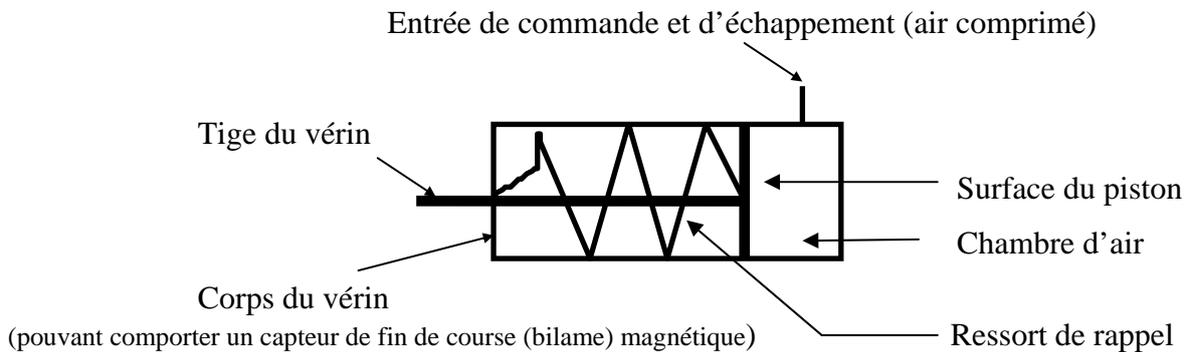
-II-LE VERIN

En technologie fluide, il peut être pneumatique ou hydraulique. On se limitera à l'étude du principe du vérin pneumatique linéaire à tige.

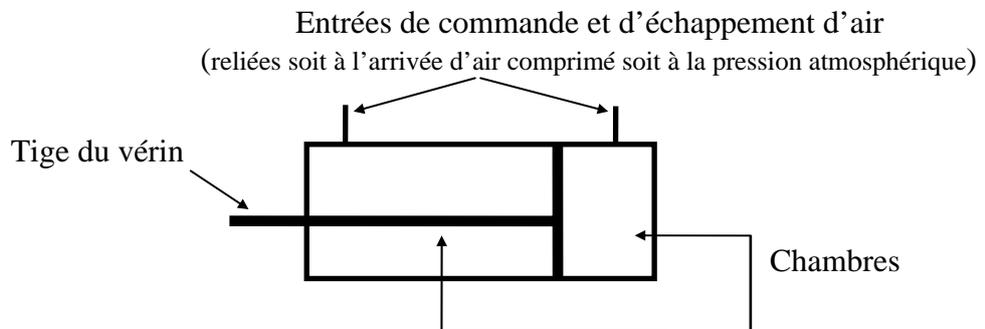
De l'air comprimé est envoyé dans la chambre du vérin. Sous l'effet de la pression exercée sur la surface du piston, la tige du piston sort du corps du vérin avec une force F . La rentrée de la tige s'effectue soit sous l'effet d'un ressort de rappel (simple effet), soit pas le biais d'une autre commande d'air comprimé sur la surface opposée du piston (double effet).

-1-Description

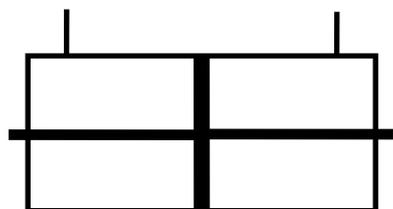
-Vérin simple effet (une entrée de commande et ressort de rappel) :



-Vérin double effet (2 entrées de commande) :



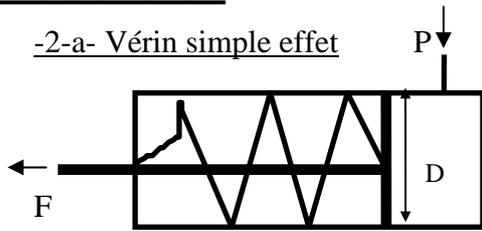
-Vérin double effet 2 tiges (1 ou 2 entrées de commande) :



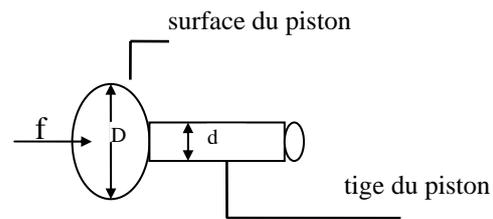
2 entrées de commande



1 entrée de commande et ressorts de rappel

-2-Dimensionnement

F : force théorique exercée par le vérin

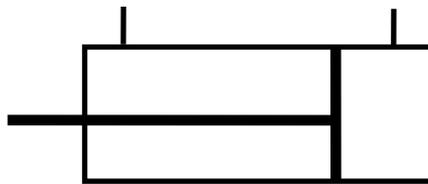


$f = P \times S$ (d'après la loi de Mariotte),

où S : section = $\Pi D^2 / 4$, P : pression exercée sur la surface

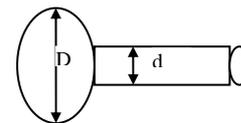
F_{th}	=	$P \times S$	-	$F_{ressort}$
Newton		$Mpa \ mm^2$		Newton

$(1 \text{ N/mm}^2 = 1 \text{ Mpa} = 10 \text{ bars})$

-2-b-Vérin double effet

S : section du piston

s : section de la tige



*Force théorique exercée par la tige en sortant :

$F_{ts} = P \times S = P \times \Pi D^2 / 4$
--

*Force théorique exercée par la tige en rentrant :

$F_{tr} = P \times (S - s) = P \times \Pi/4 (D^2 - d^2)$
--

-2-c-Taux de charge d'un vérin

$\tau = \text{Force utile} / \text{Force théorique}$
--

(en général $0,5 < \tau < 0,8$)

La force théorique correspond à l'effort statique : c'est l'effort développé par le vérin à l'arrêt (exemple bridage, serrage...).

La force utile correspond à l'effort dynamique : c'est l'effort disponible sur le vérin (à l'extrémité de la tige pour un vérin linéaire) pour déplacer une charge à vitesse convenable.

Le taux de charge communément utilisé est de 0,6.

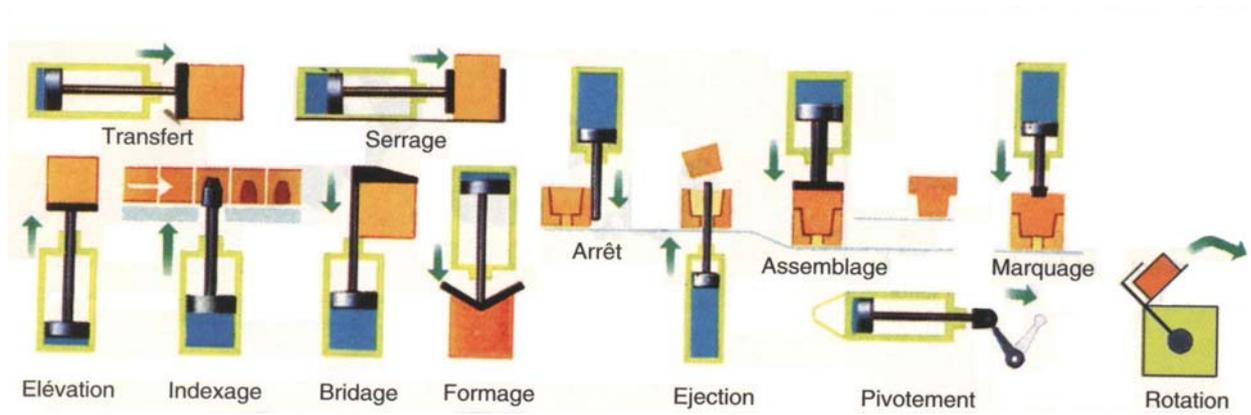


Figure 3.1 : Différentes utilisations d'un vérin

-2-d- Consommation d'un vérin

La consommation C est généralement donnée par les abaques des constructeurs. Elle est fonction de la section et de la pression de travail. Elle s'exprime en litre par centimètre de course du vérin. Le volume d'air (V) consommé pendant une certaine durée s'exprime en litres ou en litres/heure.

*Vérin simple effet :

$$V = C \times L = S \times L$$

(volume à exprimer en litres)

C : Consommation en l/cm ; L : course du vérin en cm ; S : section en cm²

*Vérin double effet :

$$V_{\text{sortie}} := V \text{ du vérin simple effet} = S \times L = C_{\text{sortie}} \times L = C_{\text{piston}} \times L$$

$$V_{\text{rentrée}} := (S - s) \times L = S \times L - s \times L = V_{\text{sortie}} - V_{\text{tige}}$$

$$V_{\text{tige}} := C_{\text{tige}} \times L$$

$$V_{\text{rentrée}} = (C_{\text{piston}} - C_{\text{tige}}) \times L = C_{\text{rentrée}} \times L$$

Pour un aller-retour du vérin : $C_{\text{totale}} = C_{\text{sortie}} + C_{\text{rentrée}}$

$$V_{\text{total}} = V_{\text{sortie}} + V_{\text{rentrée}} = 2 V_{\text{sortie}} - V_{\text{tige}} = C_{\text{totale}} \times \text{course } L$$

$$= L \times (2 \times S - s) = L \times (2 \times C_{\text{piston}} - C_{\text{tige}})$$

-3-Détection de la fin de course d'un vérin

Pour connaître la position exacte des vérins (tige sortie ou rentrée), les éléments de détection sont soit placés sur la partie opérative ou sur la machine (capteurs à action mécanique), soit implantés directement sur le vérin (capteurs à action pneumatique ou magnétique).

-3-a- Capteurs pneumatiques à seuil de pression

Ces capteurs utilisent la chute de pression dans la chambre d'échappement du vérin, et sont directement montés sur les orifices du vérin. Ils peuvent émettre un signal de sortie pneumatique, électrique ou électronique.

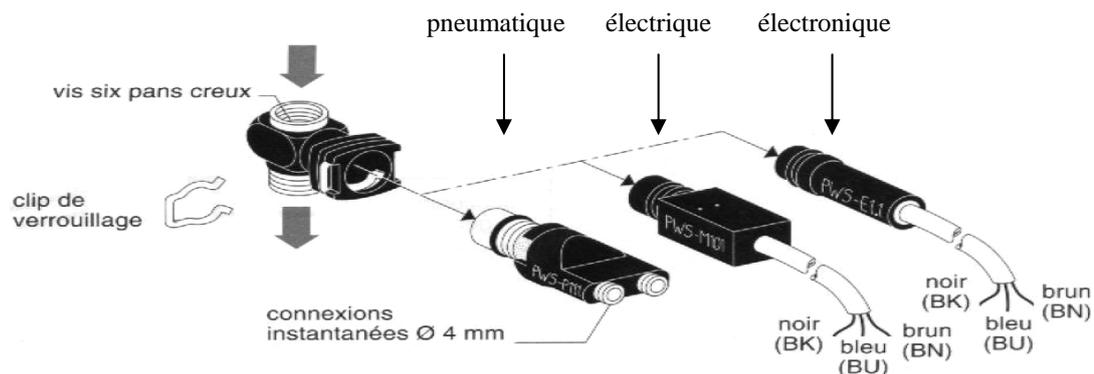


Figure 3.2 : Différents capteurs à seuil de pression

(pour le détail du capteur à sortie pneumatique, se référer au paragraphe IV-1-a-2)

-3-b- Capteurs électriques et électroniques à détection magnétique

Ces capteurs se placent directement sur le fût aimanté du vérin. Un aimant permanent disposé dans le piston crée un champ magnétique. Le piston en se déplaçant ferme le contact électrique ou déclenche le système de détection électronique du capteur.

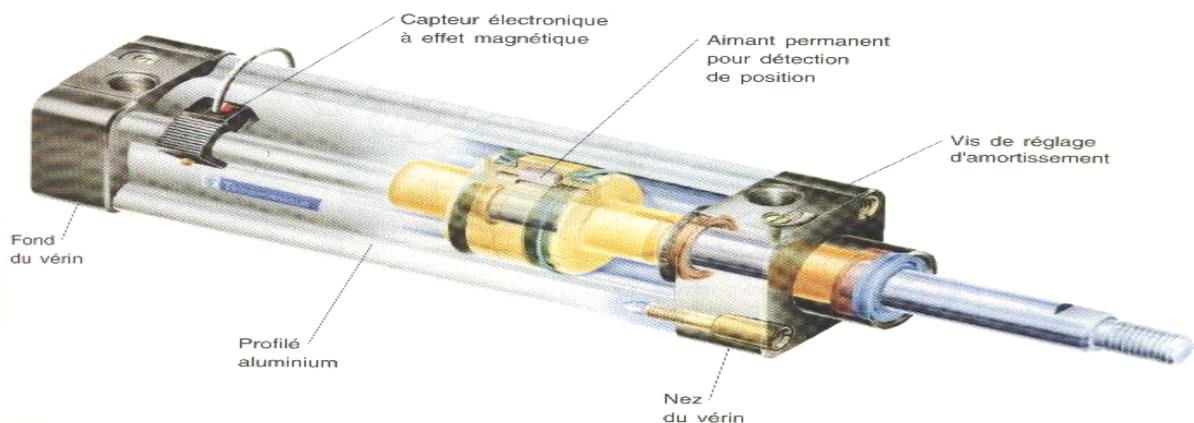


Figure 3.3 : Vérin muni d'un aimant et d'un capteur magnétique

-3-c- Capteurs pneumatiques à action mécanique

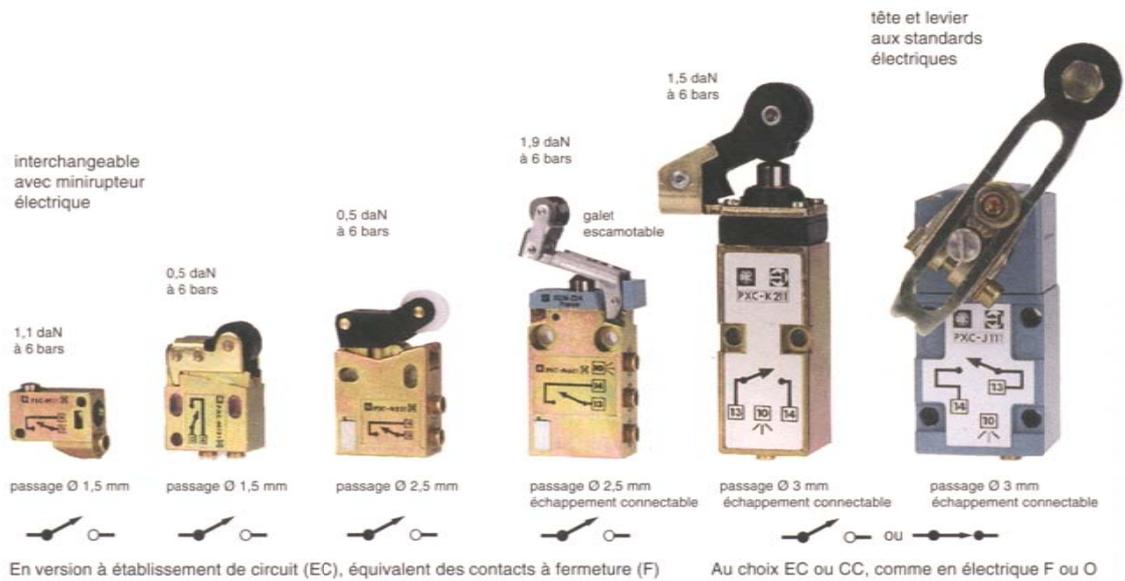


Figure 3.4 : Quelques capteurs pneumatiques à action mécanique
(d’après documentation Parker Pneumatic)

-4-Différents types de vérins

Bien que le vérin à tige et action linéaire soit le plus répandu (normalisé ISO), il existe cependant d’autres types de vérins, qu’ils soient sans tige ou à action rotative.

	Mini-vérins Vis	Mini-vérins ISO 6432	Vérins à tige ISO 6431	Vérins à tige Course courte	Vérins Sans tige	Vérins rotatifs
Technologie Spécificité	-Tige inox -Corps laiton	-Tige et tube inox -Nez et fond plastique ou métal	-Tige inox -Tube profilé aluminium ou -Version à tirants	Profilé aluminium	Profilé aluminium à tige fendue	-Corps aluminium extrudé ; -Extrémités et palette aluminium anodisé dur
Version	Simple effet	Simple et double effet	Double effet	Simple et double effet	Double effet	Double effet
Amortissement	Sans	Elastique ou pneumatique	Pneumatique réglable	Sans	Pneumatique	Elastique
Course standard en mm	5,10,15	De 10 à 200 suivant diamètre	De 25 à 800 suivant diamètre	De 5 à 100 suivant diamètre	De 0 à 8	Angle de rotation : - 0 à 280° simple palette - 0 à 100° double palette
Diamètre en mm (diamètres normalisés : 6, 10, 12, 16, 20, 25, 32, 40, 50, 63, 80, 100, 125, 160, 200)	6, 10, 16	10 à 25	32 à 200	12 à 63	25, 32, 40	Couple en N.m : - 0.5 à 15 en simple palette - 1.5 à 30 en double palette

(d’après documentation Parker Pneumatic)

-5-Exercice d'application

On considère 1 chariot (de masse à vide 110 kg) pouvant transporter des pièces (de masse 70 kg) entre 2 postes de travail distants de 1 mètre. Il est alimenté en pièces par un vérin de transfert simple effet, dont la tige a une longueur de 50 cm et le ressort de rappel une force de 200N. Ses déplacements sont effectués par le biais d'un vérin double effet, dont l'extrémité de la tige (de diamètre 40 mm) est solidaire du chariot.

Questions

Si le taux de charge des vérins est de 0.5, que l'on travaille à une pression de 3.14 bars, et que la cadence de travail des vérins est de un aller/retour toutes les 6 secondes, calculer:

- 1° Les diamètres des pistons des vérins simple effet et double effet.
- 2°-Les forces développées par les deux vérins en sortant et en rentrant.
- 3°) Les volumes d'air consommé par minute pour chaque vérin en sortant et en rentrant.
- 4°) Le volume horaire d'air global consommé par l'installation.

Pression de travail: 3,14 bars	
Diamètre	Consommation
mm	l/cm
40	0,04
65	0,1
70	0,125
80	0,16
100	0,24
120	0,3
125	0,4
140	0,5
150	0,6

(d'après documentation FESTO)

Solution**1° -a- Vérin simple effet**

$$F_{\text{utile}} = 70 \times 10 = 700\text{N}$$

$$F_{\text{théorique}} = F_{\text{ut}}/T = 700/0.5 = 1400\text{N}$$

$$F_{\text{th}} = P \times S - F_{\text{ress}} = P \times \pi \times D^2 / 4 - F_{\text{ress}} \Rightarrow D^2 = 4 (F_{\text{th}} + F_{\text{ress}}) / P \times \pi$$

$$P = 3.14 \text{ bars} = 0.314 \text{ N/mm}^2$$

$$D^2 = 4 \times 1600 \Rightarrow D = 2 \times 40 = 80 \text{ mm.}$$

-b- Vérin double effet

$$F_{\text{ut}} = (110+70) \times 10 = 1800\text{N} \Rightarrow F_{\text{th}} = F_{\text{ut}}/T = 1800/0.5 = 3600\text{N}$$

$$F_{\text{th}} = P \times S = P \times \pi \times D^2 / 4 \Rightarrow D^2 = 4 F_{\text{th}} / P \times \pi = 4 \times 3600 \Rightarrow D = 120 \text{ mm}$$

2° -a- Vérin simple effet

$$\text{en sortant : } F_{\text{th}} = 1400 \text{ N}$$

$$\text{en rentrant : } F_{\text{rent}} = F_{\text{ress}} = 200 \text{ N}$$

-b- Vérin double effet

$$\text{en sortant : } F_{\text{th}} = 3600 \text{ N}$$

$$\begin{aligned} \text{en rentrant : } F_{\text{rent}} &= P (S - s) = P \times (\pi \times D^2 / 4 - \pi \times d^2 / 4) = P \times \pi / 4 (D^2 - d^2) \\ &= \frac{1}{4} (4 \times 3600 - 1600) = \frac{1}{4} \times 4 (3600 - 400) \\ &= 3200 \text{ N} \end{aligned}$$

3° -a- Vérin simple effet

$$D = 80\text{mm}, P = 3.14 \text{ bars} \Rightarrow \text{consommation } C = 0.16 \text{ l/cm}$$

$$V_{\text{sortie}} = C \times L (\text{Course}) = 0.16 \text{ l/cm} \times 50 \text{ cm} = 8 \text{ litres.}$$

$$V_{\text{entrée}} = 0 \text{ car vérin simple effet et retour par ressort de rappel.}$$

$$V_{\text{total}} = V_{\text{sort}} + V_{\text{rent}} = 8 \text{ l}$$

$$\text{Cadence } 1 \text{ aller/retour en } 6 \text{ secondes} \Rightarrow 10 \text{ allers/retours en } 1 \text{ minute.}$$

$$V_{\text{tot/min}} = 8 \times 10 = 80 \text{ litres.}$$

-b- Vérin double effet

$$D = 120\text{mm}, P = 3.14 \text{ bars} \Rightarrow \text{consommation } C = 0.3 \text{ l/cm}$$

$$\phi_{\text{tige}} = 40\text{mm}, P = 3.14 \text{ bars} \Rightarrow \text{consommation tige } c = 0.04 \text{ l/cm}$$

1^o méthode :

$$V_{\text{sort}} = C \times L = 0.3 \times 100 = 30 \text{ litres}$$

$$V_{\text{tige}} = c \times L = 0.04 \times 100 = 4 \text{ litres}$$

$$V_{\text{rent}} = V_{\text{sort}} - V_{\text{tige}} = 30 \text{ l} - 4 \text{ l} = 26 \text{ litres}$$

$$V_{\text{tot}} = V_{\text{sort}} + V_{\text{rent}} = 30 \text{ l} + 26 \text{ l} = 56 \text{ litres}$$

$$V_{\text{tot/min}} = 56 \times 10 = 560 \text{ litres.}$$

2^o méthode :

$$C_{\text{sort}} = C = 0.3 \text{ l/cm}$$

$$C_{\text{rent}} = C_{\text{sort}} - C_{\text{tige}} = C - c = 0.3 \text{ l/cm} - 0.04 \text{ l/cm} = 0.26 \text{ l/cm}$$

$$C_{\text{tot}} = C_{\text{sort}} + C_{\text{rent}} = 0.56 \text{ l/cm}$$

$$V_{\text{tot}} = C_{\text{tot}} \times L = 0.56 \times 100 = 56 \text{ litres}$$

$$V_{\text{tot/min}} = 56 \times 10 = 560 \text{ litres.}$$

4^o Volume global horaire

$$V_{\text{global/min}} = V_{\text{simple effet}} + V_{\text{double effet}} = 80 + 560 = 640 \text{ litres}$$

$$V_{\text{horaire}} = V_{\text{min}} \times 60 = 640 \times 60 = 38400 \text{ litres} = 38.4 \text{ mètres cubes.}$$

Remarque :

On observe que la consommation horaire est trop élevée. Ceci est dû d'une part aux très grands diamètres des pistons, d'autre part à la cadence de travail très élevée (un aller retour toutes les 6 secondes).

Cette installation nécessite donc une immense centrale de production d'air comprimé uniquement pour 2 vérins. Par conséquent il y a eu de mauvais choix technologiques.

-III-LE DISTRIBUTEUR « PILOTE »

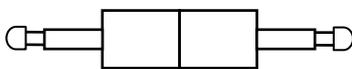
De même que le contacteur est associé au moteur électrique, le distributeur est le préactionneur associé au vérin pneumatique.

Le but d'un distributeur est de mettre en communication divers orifices deux à deux. Ceci peut être réalisé soit par rotation d'un disque distributeur ou d'un boisseau, soit par glissement d'un tiroir, soit par soulèvement d'un clapet. En pratique les distributeurs à tiroir sont les plus répandus : le tiroir piloté par les signaux pneumatiques de commande commute les canaux de pression et d'échappement, alternant ainsi les états des orifices de sortie raccordés au vérin.

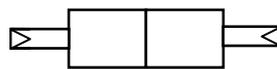
-1-Moyens de pilotage ou de commande

Fonctionnellement le distributeur a pour rôle de délivrer un débit d'air comprimé (pour commander un vérin) à la réception d'un signal de commande. La commande se fait soit par pilotage pneumatique, soit par électrovanne recevant le signal électrique de commande, soit par commande manuelle auxiliaire (pour les interventions de mise au point).

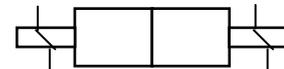
Bistable, un distributeur garde sa position (fonction mémoire) en l'absence de signal de pilotage. Monostable, il est rappelé à sa position stable de repos, soit par un ressort de rappel, soit par une pression permanente (retour différentiel).



commandes mécaniques



commandes pneumatiques



commandes électriques

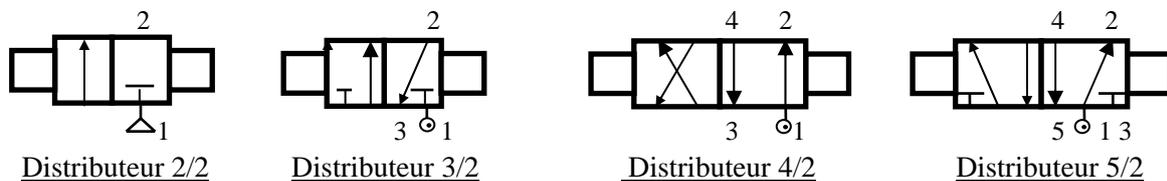
Le module de pilotage peut être intégré ou non au distributeur.

-2-Symboles normalisés :

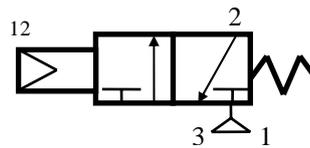
On parle de distributeur 2/2, 3/2, 4/2, 4/3, 5/2. Le premier chiffre correspond au nombre d'orifices et le second au nombre de positions de travail. Il existe un troisième élément qui caractérise un distributeur, c'est le nombre de voies (ou chemins internes) possibles de passage de l'air comprimé. Les numéros des orifices sont normalisés : 1 pour l'arrivée d'air, pairs (2 et 4) pour les sorties de commande, impairs (3 et 5) pour les sorties d'échappement.

Dans la représentation, le nombre de rectangles (ou de carrés) correspond au nombre de positions de travail du distributeur. Les orifices d'admission et d'échappement, ainsi que les numéros des orifices, sont représentés sur le rectangle correspondant à la position de repos.

Pour l'admission d'air, on utilise soit un triangle, soit un cercle avec un point au centre (le cercle est noir dans le cas de l'énergie hydraulique). Pour l'échappement on utilise un triangle.



Ce chiffre 12 indique que la commande met en communication les orifices 1 et 2



Distributeur 3/2 à commande pneumatique et retour par ressort de rappel

Généralement le distributeur 3/2 est utilisé pour la commande des vérins simple effet, et les distributeurs 4/2 et 5/2 pour les vérins double effet. Un étrangleur réglable par une vis est généralement disposé sur l'échappement pour régler le débit (et donc la vitesse du vérin).

-3-Electrodistributeurs

Si la commande est électrique on parle d'électrodistributeur (le courant pouvant être continu ou alternatif).

-3-a- Distributeur 3/2

Il est le plus souvent monostable, il possède donc un seul état stable qui est la position de repos. Il est alors commandé par une bobine E et un ressort de rappel, et il faut maintenir la commande sinon il revient automatiquement à la position de repos sous l'effet du ressort.

Quand la bobine est alimentée, l'électrovanne s'ouvre pour laisser passer l'air comprimé, mettant ainsi en communication les orifices 1 et 2 (d'où le chiffre 12 à côté de la bobine) d'admission et de commande. L'orifice 2 étant relié à l'entrée d'un vérin, ce dernier est alors commandé. Quand l'alimentation de la bobine est coupée, l'électrovanne se ferme (coupant l'admission d'air), le ressort de rappel rappelle le clapet (cf figure a), mettant en communication les orifices 2 et 3, permettant ainsi à l'air (emmagasiné dans le vérin) de s'échapper par l'orifice 3.

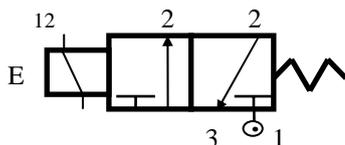


Figure a : monostable

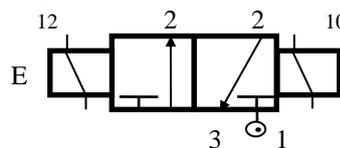
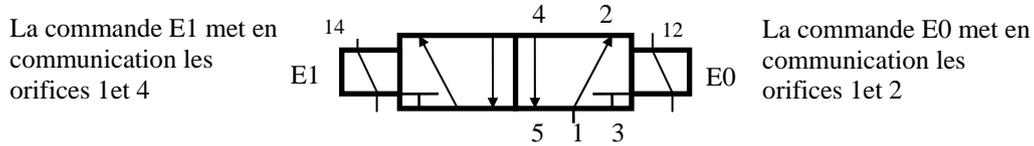


Figure b : bistable

Ce chiffre 10 indique que la commande coupe la liaison entre les orifices 1 et 2 (utilisé également sur le distributeur 2/2)

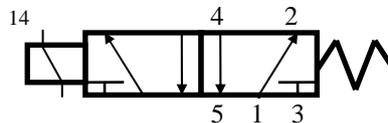
-3-b- Distributeur 5/2 : bistable

Il possède deux états stables correspondant aux deux positions de travail. Il est commandé par deux bobines E0 et E1, il a donc un fonctionnement à double pilotage. Même une impulsion sur E0 ou E1 suffit à le faire changer d'état.



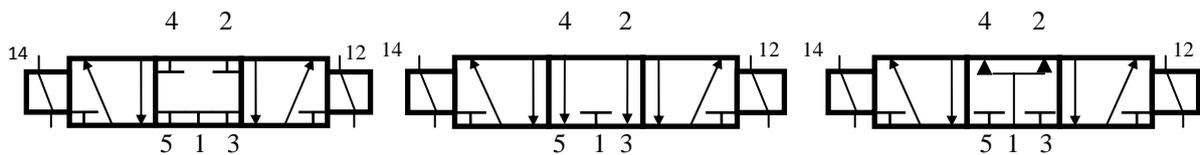
-3-c- Distributeur 5/2 : monostable

Il possède un seul état stable qui correspond à la position de repos. Les commandes s'effectuent (comme pour le 3/2) par une bobine et un ressort de rappel.



-3-d- Distributeur 5/3 monostable

La seule position stable est la position médiane : en l'absence de commande, deux ressorts de centrage (gauche et droit) ramènent le tiroir en position centrale. On parle soit de 5/3 à *point milieu bloqué* (entrée et sorties du distributeur bloquées), soit de 5/3 à *point milieu libre* (entrée bloquée et sorties reliées à l'échappement). Ainsi en l'absence de commande sur le distributeur, le vérin commandé sera soit bloqué soit libre.



5/3 à point milieu bloqué

5/3 à point milieu libre à l'échappement

5/3 à point milieu libre à l'admission

Remarque

(les 2 sorties reliées à l'échappement)

(les 2 sorties reliées à l'admission)

Un cas intéressant est de faire en sorte que toutes les positions du vérin soient des états stables. Pour l'obtenir on rajoute des *bloqueurs* (cf paragraphe III-4) sur les orifices des vérins pour figer l'état du matériel dès que la commande (E0 ou E1) du distributeur disparaît. Dès qu'elle réapparaît, le fonctionnement continue exactement de l'endroit où il s'est arrêté (utilisation en sécurité sur les asservissements, pour éviter le retour brusque à des positions de repos en cas de coupure des alimentations).

-3-e- Module d'interface électropneumatique

Chez certains constructeurs, la commande du distributeur est souvent autonome et indépendante du distributeur proprement dit. On utilise une interface modulaire à commande électrique (électrovanne) qu'on peut monter sur embase, dont la sortie pneumatique commande le distributeur. Ces modules d'interface offrent l'avantage, dans le cas des faibles puissances, de pouvoir commander directement un mini-vérin sans avoir recours à un distributeur. On rencontre des modules 3/2 à une sortie (monostable), et des modules à 2 sorties en 4/2 et 5/2 (monostable et bistable).

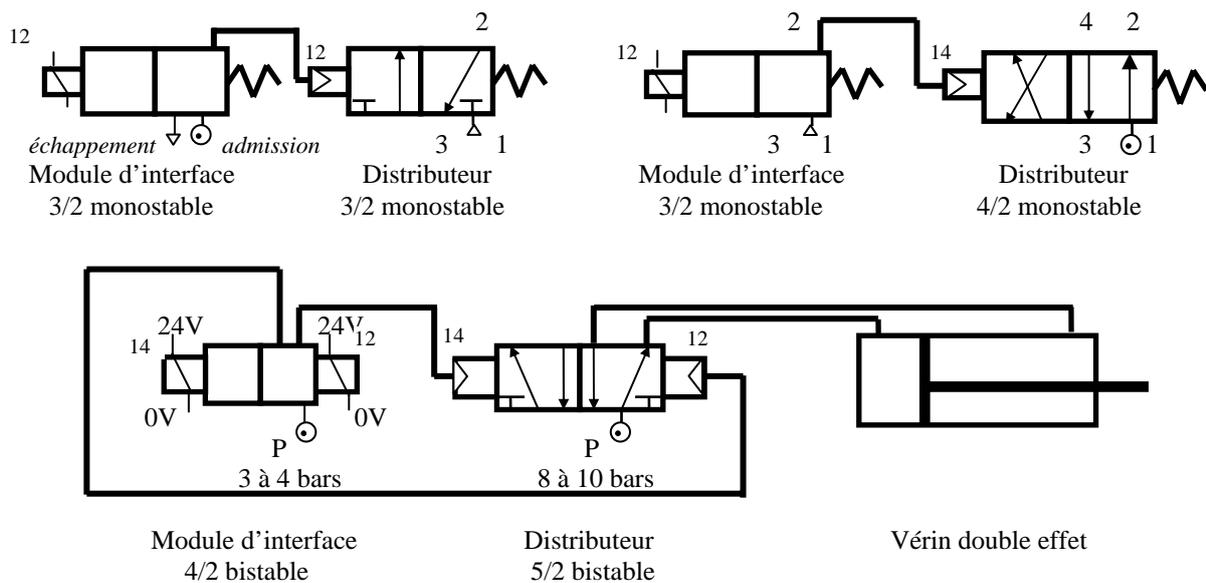


Figure 3.5-a : Connexions des modules d'interfaces aux distributeurs

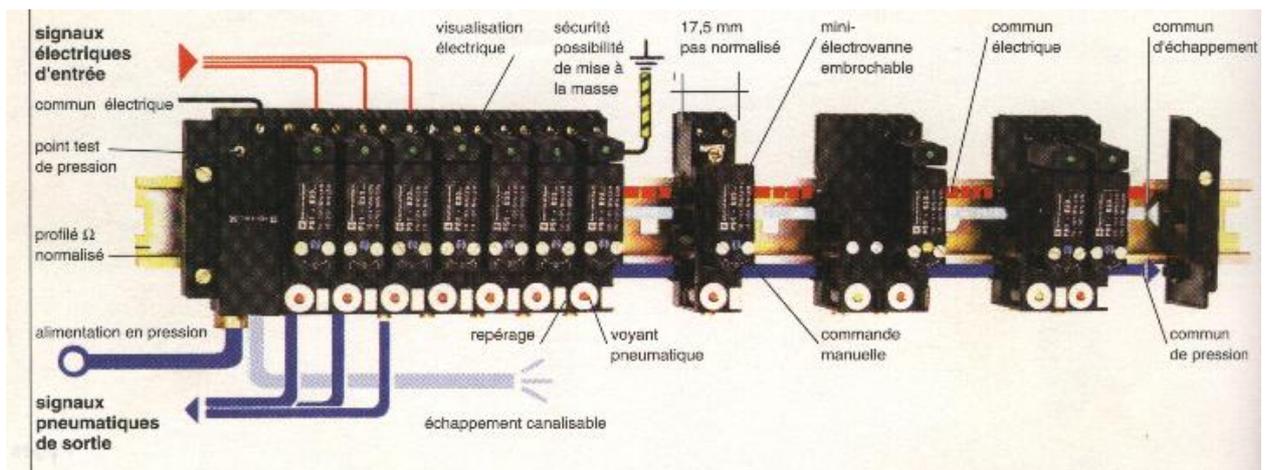


Figure 3.5-b : Interconnexion de modules d'interfaces montés sur rack

-4-Auxiliaires de distribution

Sur le circuit de puissance, entre distributeur et vérin, des auxiliaires (cf. figure 3.6) sont généralement nécessaires pour permettre :

- le réglage de la vitesse du vérin dans chacun des sens de déplacement,
- l'intégration de fonctions de sécurité, par blocage du vérin ou par purge de ses pressions d'air.

Pour toutes ces fonctions, l'implantation au plus près du vérin est la plus efficace. C'est pourquoi les auxiliaires sont prévus pour s'implanter, en général, directement sur les orifices de raccordement du vérin, en lieu et place des raccords qu'ils intègrent.

-4-a- Le régleur de vitesse

La tige du vérin pneumatique A se déplace dans le sens A+ sous l'action de la différence des pressions entre chambres amont (admission d'air) et aval (échappement d'air). La vitesse du mouvement A+ est déterminée par la vitesse de purge de l'air contenu dans la chambre aval.

Le régleur de vitesse laisse passer l'air à plein débit dans le sens admission, le clapet anti-retour étant ouvert. Il régule le débit d'air, et donc la vitesse du mouvement du vérin, dans le sens d'échappement (le clapet anti-retour est fermé) grâce à la restriction réglable .

La restriction réglable est généralement réalisée par une vis pointeau verticale (ou plus rarement par une bague de réglage à rampe annulaire). L'anti-retour est assuré soit par une jupe en élastomère soit par une bille.

Afin que le vérin ne « broute » pas et que la vitesse de déplacement soit constante, il est préférable que le régleur de vitesse soit au plus près du vérin.

-4-b- Le bloqueur 2/2

Particulièrement indiqués pour arrêter les vérins en cours de mouvement ou pour les maintenir en position, les bloqueurs 2/2 assurent un blocage efficace dès que le signal de pilotage (air comprimé) disparaît.

Deux bloqueurs (un sur chaque orifice du vérin) stoppent les débits d'air d'admission et d'échappement, immobilisant ainsi la tige du vérin et sa charge, par mesure de sécurité.

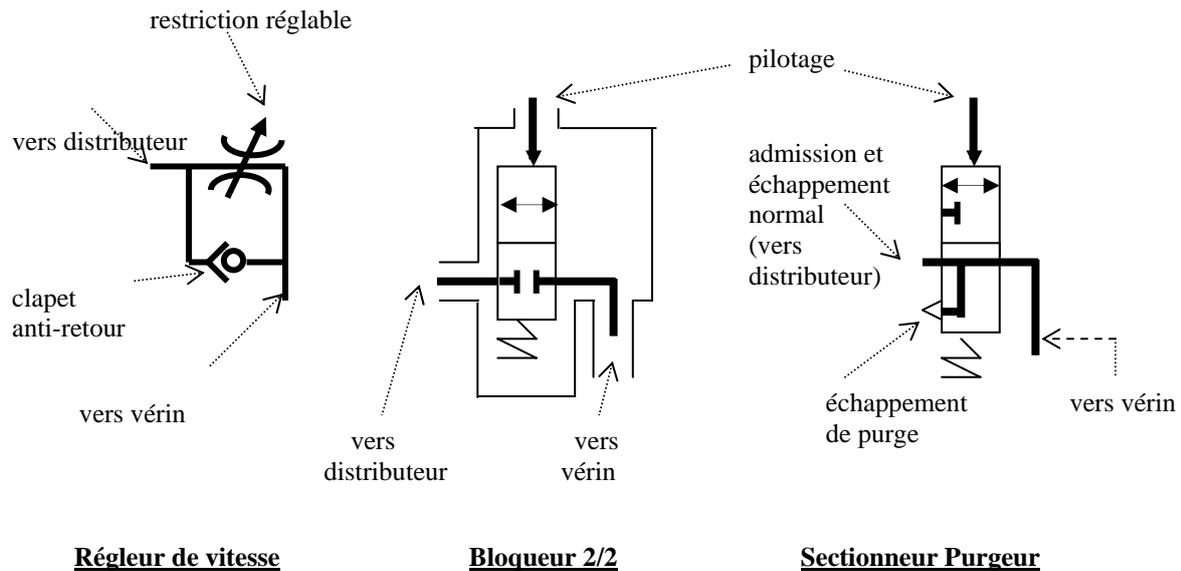


Figure 3.6 : Schémas symboliques de quelques auxiliaires de distribution

-4-c- Le sectionneur-purgeur

Le sectionneur général d'une installation peut ne pas suffire pour arrêter certains mouvements, la purge se trouvant ralentie par la restriction réglable du régleur de vitesse. Placé entre l'orifice du vérin et le régleur de vitesse, le sectionneur purgeur vide localement et rapidement la chambre du vérin. Il doit être commandé par le même signal (pneumatique) que le sectionneur général.

-4-d- Les multifonctions intégrées

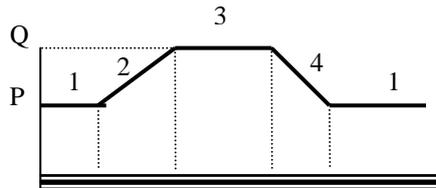
L'utilisation de bloqueurs ou de sectionneurs-purgeurs, pour répondre aux besoins de sécurité, ne dispense pas du besoin de réglage précis de la vitesse du vérin.

Les auxiliaires de distribution multifonctions répondent à cette nécessité. Ils sont conçus sous forme d'ensembles compacts et efficaces, sans augmenter l'encombrement, à monter directement sur les orifices taraudés des vérins. Ils assurent les doubles fonctions régleur de vitesse et bloqueur, ou régleur de vitesse et sectionneur-purgeur.

-5-Exemple d'application (pour le dessin et la schématisation, on utilisera un distributeur à clapet)

Un vérin commandé par un électrodistributeur doit effectuer un aller-retour complet (extrémités des fins de course P et Q) décomposé en cinq phases, selon le diagramme donné par la figure. Dans chacun des 4 cas exposés au paragraphe III-3 du chapitre 3 (distributeur monostable 3/2, bistable 5/2, monostable 5/2, monostable 5/3), donner le diagramme des phases, c'est à dire donner dans chaque cas les signaux de commande nécessaires sur les bobines (A0, A1), B, C, (D1, D2) pour réaliser dans l'ordre les 5 phases du dessin intitulées 1, 2, 3, 4, 1.

Diagramme des phases



Principe

Cas a

*Vérin double effet.

*Distributeur 5/2 bistable à 4 voies, à double pilotage, commandé par 2 bobines A0 et A1.

*Même une impulsion sur A1 provoque le changement d'état du distributeur, et l'enchaînement des étapes 2 et 3.

*Une simple impulsion sur A0 provoque le retour du vérin aux conditions initiales (passage par les phases 4 puis 1).

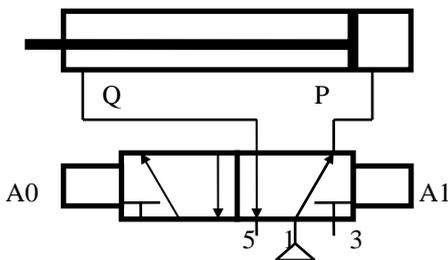
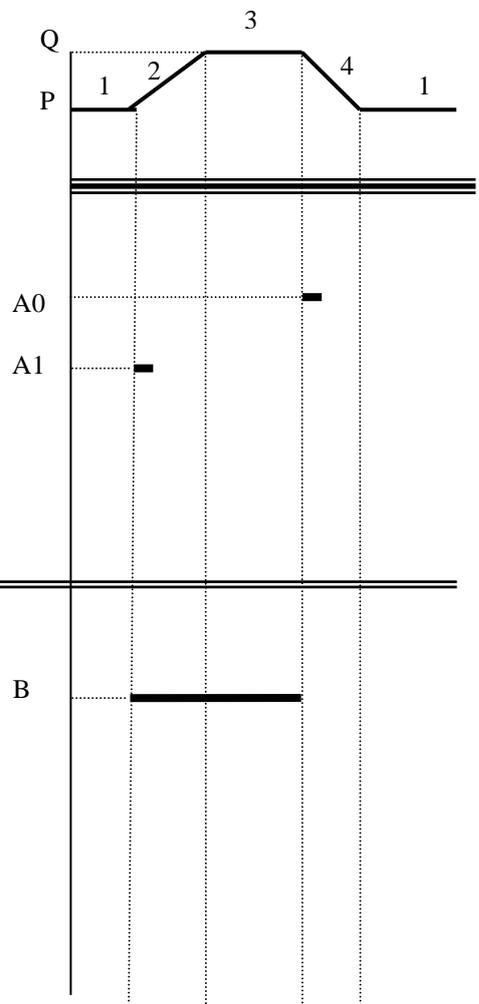


Diagramme des phases

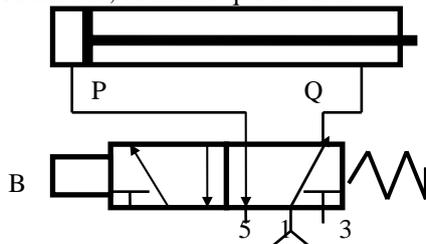


Cas b

*Vérin double effet.

*Distributeur 5/2 monostable à 4 voies, commandé par une bobine B et un ressort de rappel.

*Pour le déplacement du vérin, le maintien de B est nécessaire. Dès la suppression de B, le tiroir du distributeur revient à sa position initiale, de même que le vérin.



Principe

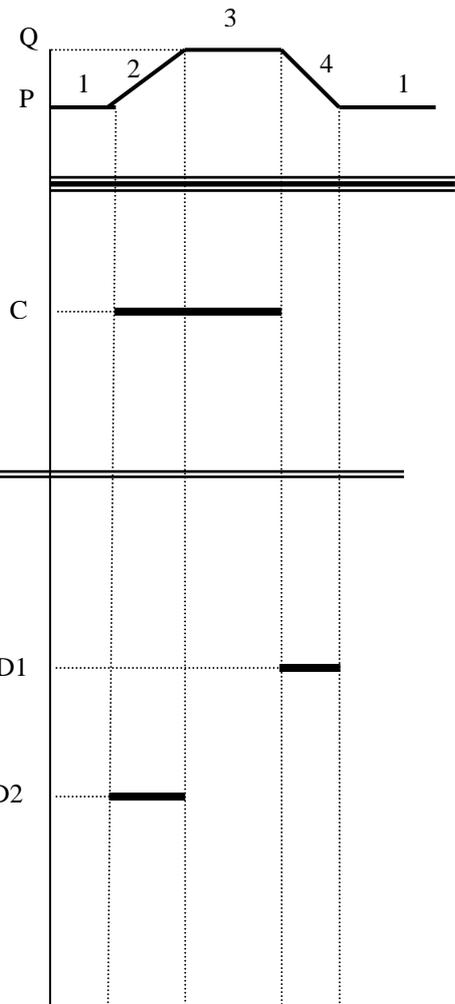
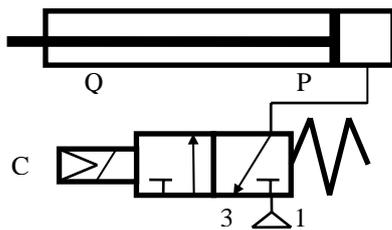
Diagramme des phases

Cas c

*Vérin simple effet et ressort de rappel.

*Distributeur 3/2, 2voies, commandé par une bobine C et un ressort de rappel.

*Même principe de fonctionnement que dans le cas b.



Cas d

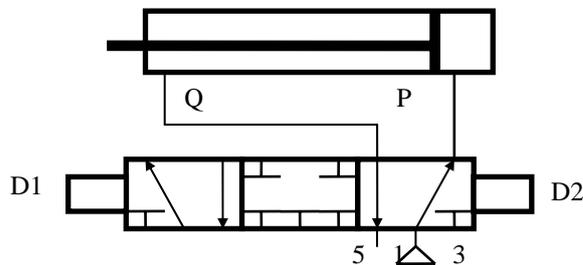
*Vérin double effet.

*Distributeur 5/3 monostable, 4 voies, avec **point milieu bloqué**; commandé par 2 bobines et deux ressorts de centrage.

*Dès la disparition du signal de commande, le tiroir du distributeur revient à la position médiane stable, et le vérin commandé reste figé dans la position à laquelle il est arrivé.

*Le maintien de la commande des bobines pour les phases 3 et 1 est inutile car le point milieu est bloqué.

*On aurait obtenu exactement le même résultat en remplaçant ce distributeur 5/3 par 2 distributeurs 3/2 monostables, et en rajoutant deux bloqueurs sur le vérin.



-IV- LES INTERFACES

On entend par interfaces les composants reliant les parties commande et opérative : soit la partie commande à l’actionneur, soit le capteur à la partie commande. Les préactionneurs, les coupleurs d’entrées/sorties, les modules d’entrées/sorties analogiques etc., font donc partie des interfaces.

Les interfaces de puissance ou préactionneurs (déjà cités dans le paragraphe I) qui peuvent être considérés comme une catégorie à part entière, s’insèrent naturellement entre l’interface de sortie et l’actionneur.

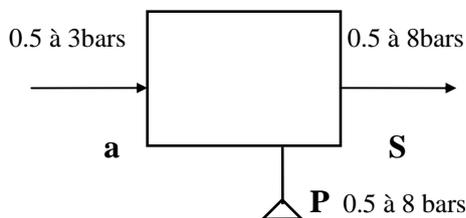
Les interfaces remplissent l’une des fonctions suivantes : conversion, adaptation, isolement. On les classera en deux catégories : celles qui modifient les paramètres d’un signal, et celles qui modifient la nature d’un signal.

-1-Interfaces modifiant les paramètres d’un signal

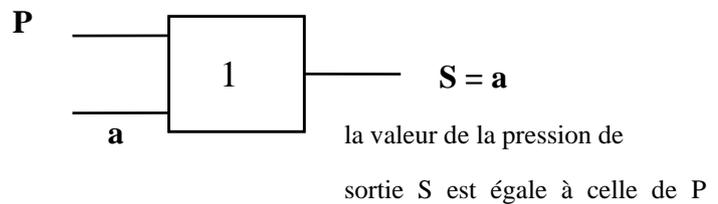
-1-a-Pneumatiques

-1-a-1-Cellule OUI

Synoptique



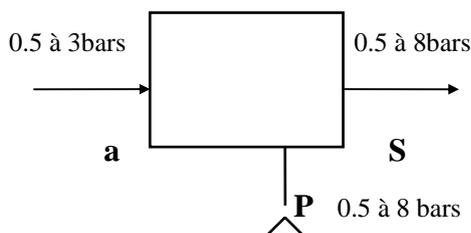
Symbole



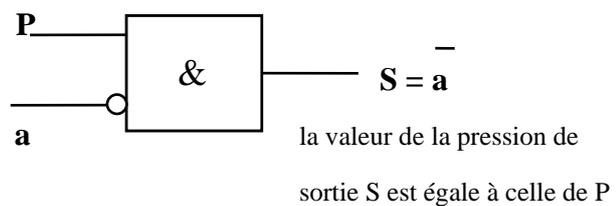
Suivant la valeur de P, la cellule peut être destinée à augmenter ou à réduire la pression d’un signal. Elle est surtout utilisée pour garantir une pression suffisante aux pilotes des préactionneurs (distributeurs).

-1-a-2-Cellule NON

Synoptique

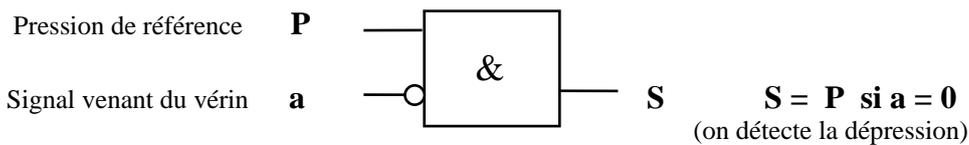


Symbole



Cette cellule est également appelée cellule inhibition, car on peut l'utiliser pour inhiber une entrée. Si on remplace la pression P par un signal b , la sortie correspondra alors à l'inhibition de l'entrée b par le signal a ($S = \bar{a} b$)

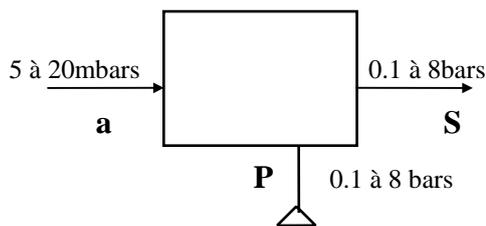
Montée sur un vérin, cette cellule est souvent utilisée comme capteur fin de course : elle détecte la chute de pression dans la chambre d'échappement quand le piston arrivant à l'extrémité a chassé tout l'air de la chambre. On l'appelle alors capteur à seuil de pression à sortie pneumatique.



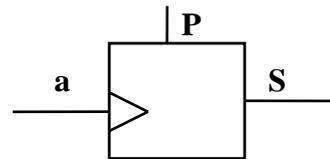
-1-a-3-Relais amplificateur

Il permet de transformer un signal de très basse pression en un signal de pression industrielle.

Synoptique



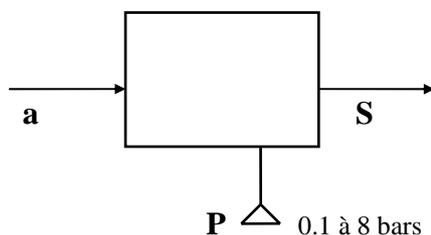
Symbole



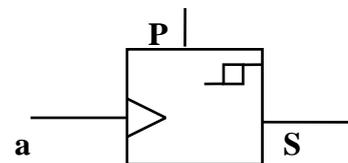
-1-a-4-Relais manostatique et vacuostatique

* Manostat ou pressostat

Synoptique



Symbole



P_2 : pression de pré réglage du mini-détendeur du manostat

Si $a < P_2 \rightarrow S = 0$

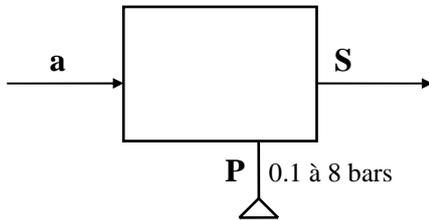
Si $a \geq P_2 \rightarrow S = P$

(Exemple d'utilisation : détection de serrage de pièces).

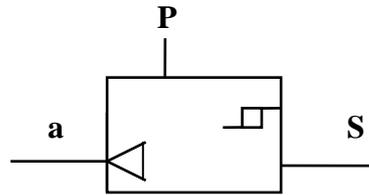
* Vacusostat

Détecte le seuil de vide pré réglé sur l'appareil, et autorise un signal de sortie.

Synoptique



Symbole



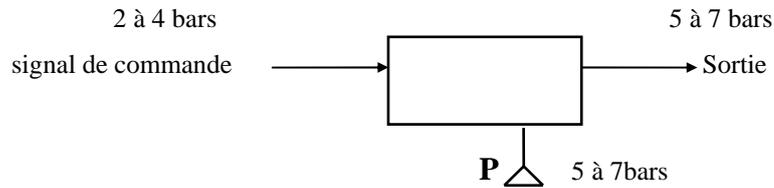
P_2 : pression de pré réglage du minidétendeur du manostat

Si $a > P_2 \rightarrow S = 0$

Si $a \leq P_2 \rightarrow S = P$

-1-a-5-Distributeur piloté (voir son étude au paragraphe III)

Dès qu'il reçoit un signal de commande la sortie S devient égale à P.



-1-b- Interfaces électriques

-1-b-1-Transformateurs

Leur utilisation principale est la transformation d'une tension alternative en courant alternatif.

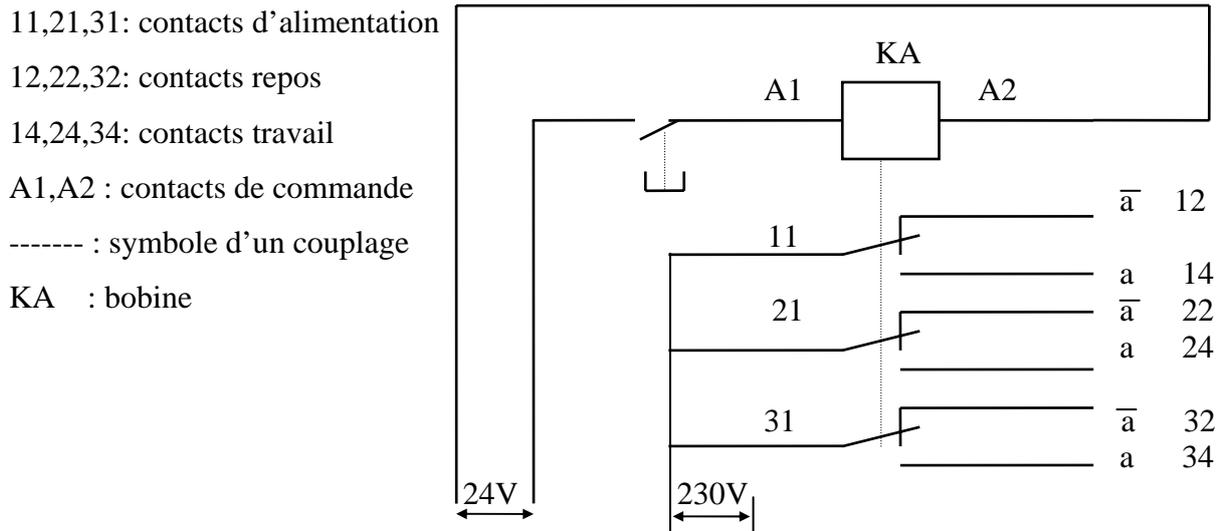


La puissance apparente s'exprime en VA (voltampères).

Remarque : dès qu'un transformateur est alimenté, il se comporte comme une résistance de faible valeur. Il est donc normal qu'il chauffe (même s'il travaille à vide).

-1-b-2-Les relais électromagnétiques• Relais instantané (RHN) ou mémoire monostable**Description

C'est un composant capable d'amplifier un signal, de l'inverser, ou de le mémoriser en câblant un circuit d'automaintien.



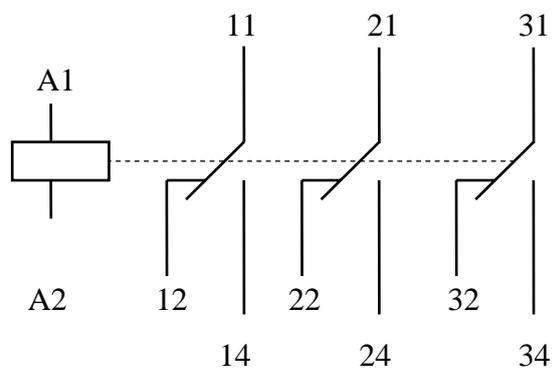
En général la tension de commande des bobines et la tension d'utilisation (sorties) sont différentes (de l'ordre de 24 V et 230 V respectivement).

**Principe de fonctionnement

Le passage du courant de commande dans la bobine (KA) crée, avec le noyau, un champ magnétique qui attire l'armature mobile qui vient s'appuyer contre le contact travail tant que la bobine est commandée. Si la bobine n'est plus alimentée, l'armature mobile est ramenée contre le contact repos grâce à un ressort.

Symbole :

11, 21, 31 : contacts d'alimentation
 12, 22, 32 : contacts repos
 14, 24, 34 : contacts travail
 A1, A2 : contacts de commande



****Caractéristiques**

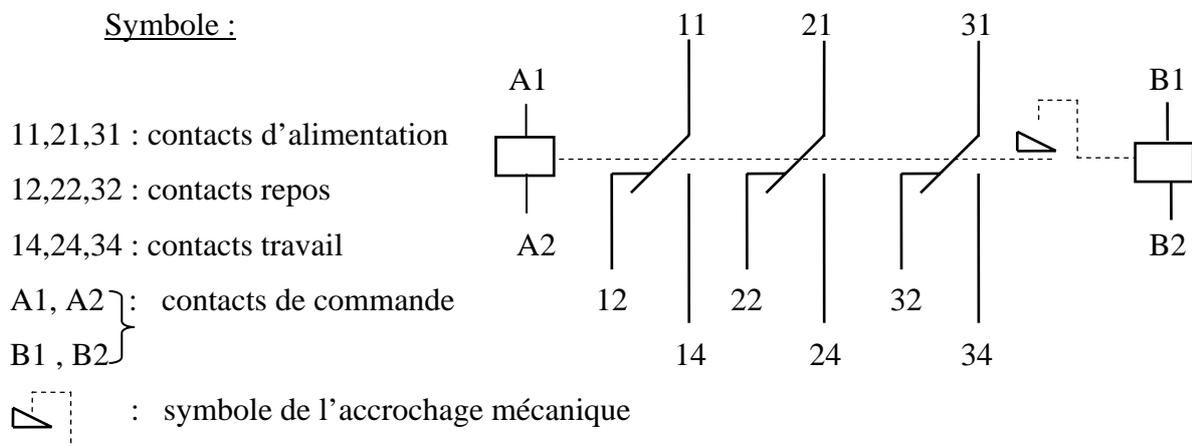
- Liées au circuit de commande (alimentation bobine) :
 - courant du circuit de commande : alternatif ou continu,
 - tensions d'alimentation : 24, 48, 110, 230 volts (6 et 12 volts pour les platines électroniques),
 - consommation moyenne : de 2.5 à 4.5 VA en alternatif, 1.6 VA en continu.
- Liées au circuit de puissance :
 - tension et intensité maximales en courant alternatif : 250 V, 5 A,
 - nombre de contacts : variable, par exemple 4 « FO » càd 4 contacts à fermeture (ouverts au repos) et 4 contacts à ouverture (fermés au repos).
- Liées au fonctionnement : (données par les abaques du constructeur)
 - cadence maximale : en nombre de manoeuvres (travail-repos) par seconde (10 par exemple),
 - endurance mécanique : en millions de manoeuvres (20 millions par exemple). Elle dépend de la tension et du courant.

- Relais électromagnétique à accrochage ou mémoire bistable (RHK)

C'est un relais instantané à deux états stables, commandé par deux bobines : l'une pour l'enclenchement (A) et l'autre pour le déclenchement (B).

Leur mémoire est assurée par un accrochage mécanique ou magnétique, donc même si on supprime le signal de commande de la bobine, le relais « reste accroché » et ne change pas d'état. Par conséquent une simple impulsion suffit à le faire changer d'état.

Symbole :

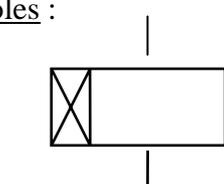


- Relais temporisés et relais clignotant

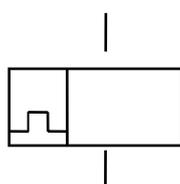
Les relais temporisés ont le même principe de fonctionnement qu'un relais instantané, mais leur action est différée dans le temps. Leur symbole est donc le même excepté pour la bobine qui comporte une information supplémentaire. On rencontre le relais temporisé au travail (RHT) et le relais temporisé au repos (RHR). La durée de la temporisation T varie de 0.1 à 300 s ou de 1 à 40 mn.

Le relais clignotant (RHC) change d'état (une fois qu'il est commandé) suivant une fréquence réglable. Le temps de réaction du relais varie de 2 à 40 ms, quant à sa période elle varie de 0.5 à 5 s, ou de 2 à 30 s.

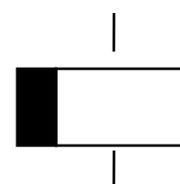
Symboles :



Relais temporisé
au travail (RHT)



Relais clignotant
RHC



Relais temporisé
au repos (RHR)

- Contacteur

C'est un préactionneur principalement destiné à la commande des moteurs.

Il est constitué d'un relais monostable (ou bistable) dit de puissance, car pouvant supporter de forts courants (utilisation courante : commande des machines asynchrones en triphasé).

Le principe de fonctionnement est le même que celui du relais : quand la bobine est alimentée, elle ferme les contacts de puissance (destinés à l'alimentation de l'actionneur) et les contacts auxiliaires (destinés à la partie commande de l'actionneur).

Le circuit de commande (alimentation bobines) est généralement alimenté en 24 V, et le circuit de puissance en 230, 400 ou 700 Volts.

On ajoute généralement avant et après le contacteur une protection des circuits commandés. Avant lui on utilise soit des fusibles coupe-circuit soit un relais magnétique (disjoncteur) pour la protection contre les court circuits. Après lui on utilise généralement un relais thermique pour la protection contre les surcharges faibles et prolongées. Pour protéger également contre les surcharges fortes on préfère parfois utiliser un relais magnétothermique.

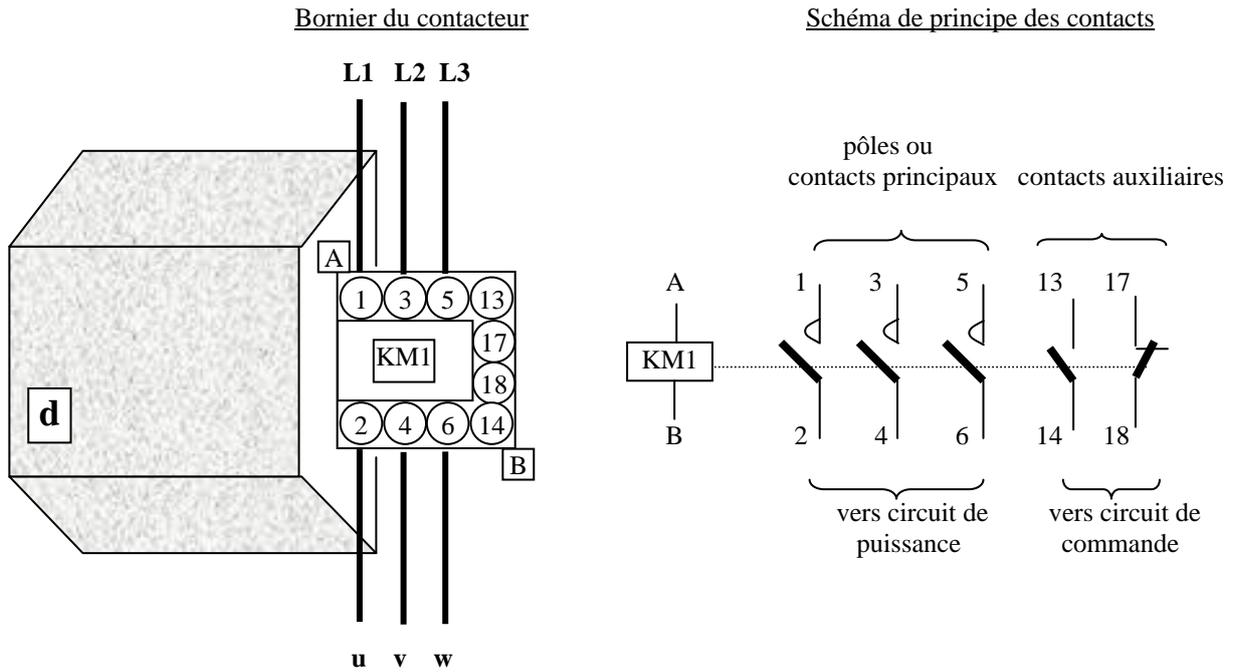


Figure 3.7 : Câblage d'un contacteur série D (doc. Telemecanique)

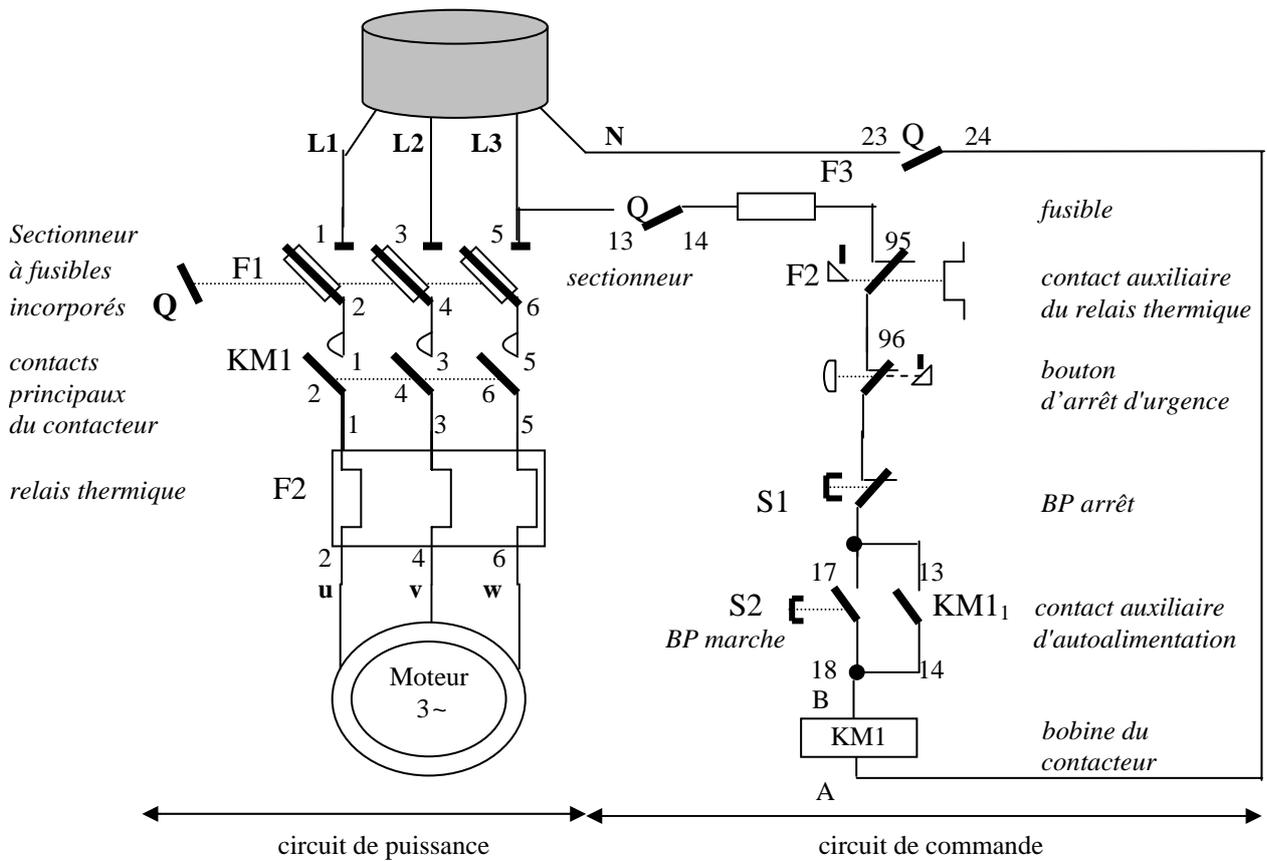


Figure 3.8 : Schéma de commande d'un moteur

-1-c- Interfaces électroniques-1-c-1-Découpleur optoélectronique

Il est utilisé en interface d'isolation galvanique (découplage) aussi bien en entrée qu'en sortie, pour isoler la partie commande de la partie opérative.

Il protège contre les perturbations électromagnétiques, les effets de rebondissement des contacts, les défaillances de câblage (courants de fuite, mauvaise mise à la terre, etc...).

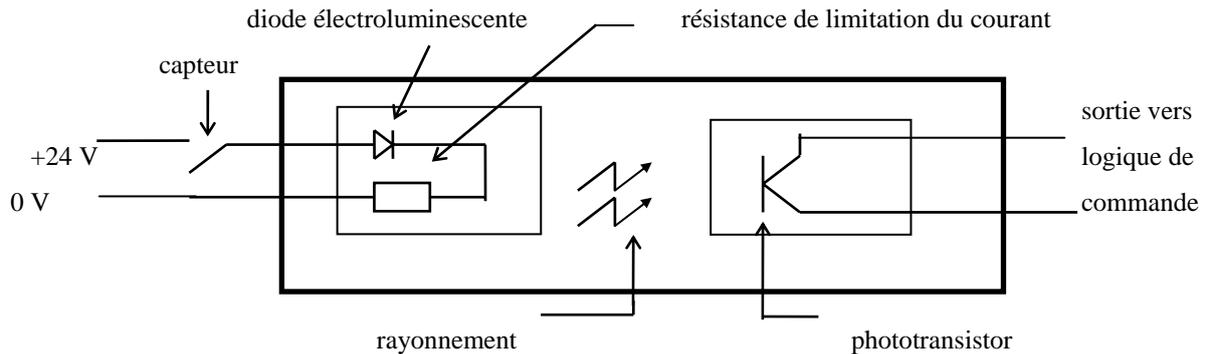


Figure 3.9 : Schéma de principe d'un optocoupleur et utilisation en interface d'entrée

-1-c-2-Les convertisseurs

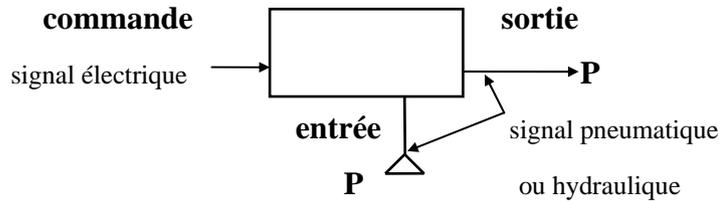
Ils sont principalement utilisés en électronique de puissance.

- Redresseur : courant alternatif \sim \longrightarrow courant continu =
- Gradateur : courant \sim sinusoïdal \longrightarrow courant \sim de forme quelconque
- Hacheur : courant = à faible ondulation \longrightarrow courant = à tension variable
- Onduleur : courant continu = \longrightarrow courant alternatif \sim

-2-Interfaces modifiant la nature d'un signal

-2-a- Les électrovannes

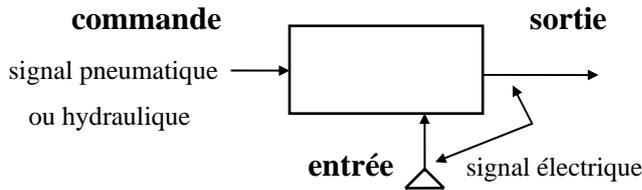
Synoptique



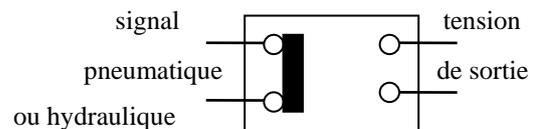
A la réception du signal de commande, le fluide en entrée se retrouve en sortie. Le signal électrique de commande est en 230, 110, 48 ou 24 volts.

-2-b- Les contacts à pression

Synoptique

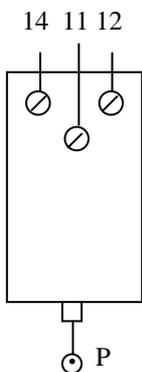


Symbole



Le signal de commande agit sur la membrane qui déplace le clapet et le rend solidaire des contacts, fermant ainsi le circuit électrique et provoquant l'apparition d'une tension en sortie.

Exemple : Contact à pression de Parker Pneumatic

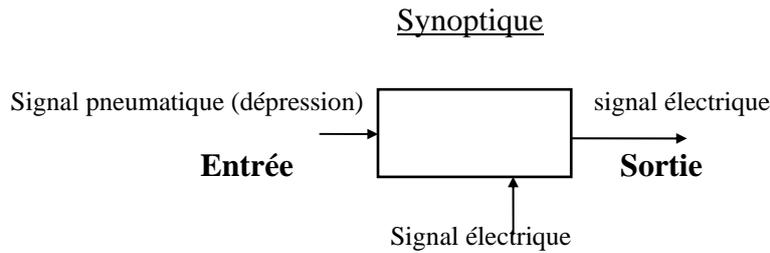


Il fonctionne comme un relais à trois lames dont la commande est pneumatique. La tension (généralement le +24V) qui est appliquée sur la borne 11 se retrouve à l'état de repos sur la borne 12 (contact repos). Quand une pression P est appliquée, les bornes 11 et 14 sont reliées et la tension se retrouve sur la sortie 14 (contact travail).

-2-c- Les capteurs à seuil de pression à sortie électrique

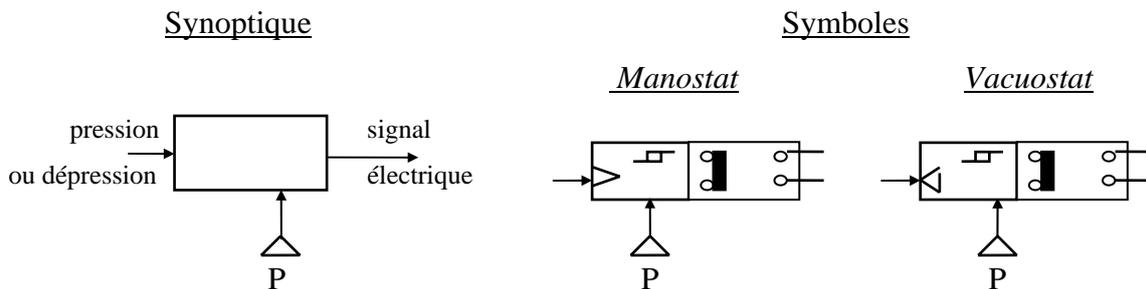
Le principe est le même que celui des contacts à pression, sauf que le signal de commande pneumatique n'est plus une pression mais une dépression (comme pour la cellule NON du paragraphe IV-1-a). On l'utilise monté sur un vérin comme capteur fin de course à sortie électrique.

Si on remplace les simples contacts de sortie par un transistor PNP, on parle alors de capteur à seuil de pression électronique.



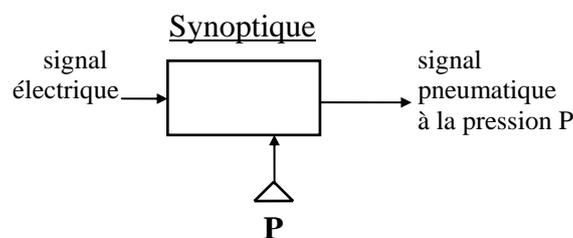
-2-d- Manostat et vacuostat à sortie électrique

Ils sont constitués par l'association d'un manostat (ou d'un vacuostat) et d'un contact à pression.



-2-e- Les électro distributeurs

Ils ont déjà fait l'objet du paragraphe III-3. Le signal électrique commande la bobine d'électroaimant d'une électrovanne, ce qui a pour effet de déplacer un clapet ou un tiroir et laisser ainsi passer l'air comprimé qui se retrouve en sortie.



-V-LA FONCTION MEMOIRE EN LOGIQUE CABLEE

-1-Généralités

Quelle que soit la technologie (pneumatique, électrique ou électronique), la fonction mémoire est matérialisée par deux entrées complémentaires X_1 (pour mise à un) et X_0 (pour mise à zéro), et par une ou deux sorties complémentaires.

En logique câblée, la mémoire correspond le plus souvent à un préactionneur.

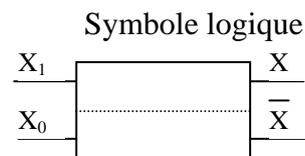
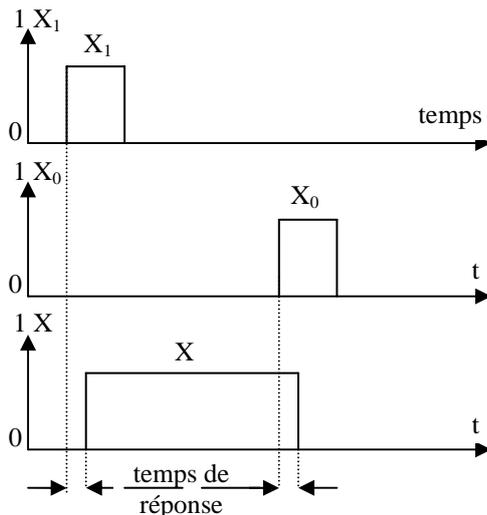


Table de vérité

X_1	X_0	X	\bar{X}
0	0	0	1
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1

Figure 3.10 : Temps de réponse d’1 mémoire

Remarques

La valeur du temps de réponse impose la durée minimale du signal de commande.

Si les deux entrées de la mémoire sont à un simultanément, la valeur de la sortie dépend de la technologie de fabrication de la mémoire: égale à 1 logique pour une mémoire à marche prioritaire et à zéro pour celle à arrêt prioritaire.

-2-Mémoire monostable

Elle se caractérise par le fait qu’elle a une seule entrée de commande. De plus le signal de commande doit être permanent; dès qu’il disparaît, un dispositif ramène la mémoire en position initiale.

-a- Pneumatique

Elle est réalisée par un distributeur monostable 3/2, ou 4/2, ou 5/2, ayant une seule commande et un retour par ressort de rappel.

Le schéma pneumatique de câblage permet de mettre en évidence la valeur importante des temps de réponse, qui sont de l’ordre de la seconde.

-b- Electrique

Elle est réalisée par un relais électromagnétique monostable, qu'il soit simple (relais instantané ou temporisé) ou intégré dans un contacteur.

-3-Mémoire bistable

Les deux signaux de commande peuvent être une simple impulsion. Cela suffit pour faire changer d'état au préactionneur.

-a- Pneumatique

On utilise un distributeur à double pilotage, donc à double commande. Les deux états du distributeur sont des états stables.

-b- Electrique

On utilise un relais électromagnétique à accrochage. Il possède donc deux bobines : l'une pour l'enclenchement et l'autre pour le déclenchement. Ce relais peut être simple ou monté sur embase pour réaliser un module d'étape de séquenceur (voir chapitre 4-IV).

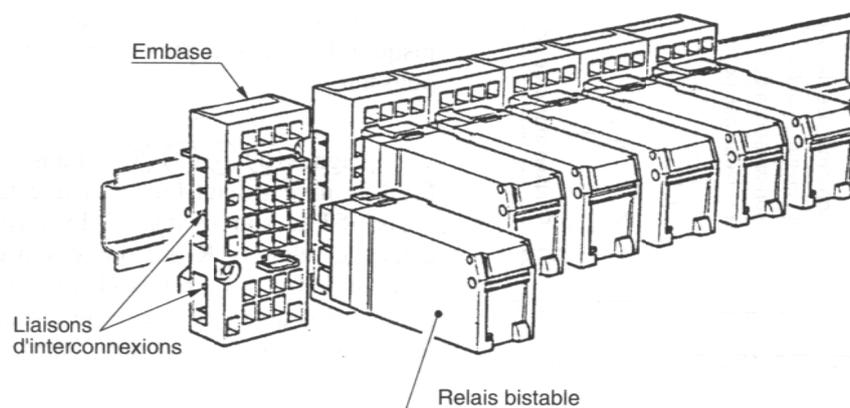


Figure 3.11 : Relais électriques montés sur embase

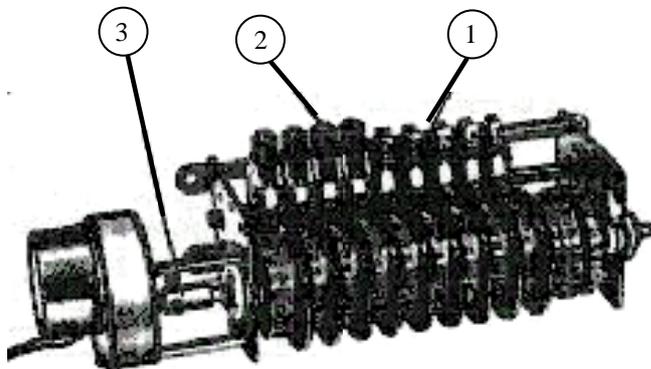
-c- Electronique

On utilise le plus souvent une bascule RS (reset, set) en circuit intégré.

-VI-LE PROGRAMMATEUR CYCLIQUE A CAMES

La machine à laver (le linge ou la vaisselle) et le distributeur de boissons sont des exemples types de systèmes automatisés, qui possèdent un programmeur cyclique à cames, comme composant principal de la partie commande.

Comme son nom l’indique, un programmeur cyclique est un dispositif programmé pour effectuer un certain travail durant un cycle de fonctionnement, ce programme pouvant changer d’un cycle à un autre. Il a pour rôle d’actionner, *au cours d’un cycle*, une série de contacts de sorties (ouverture électrovanne de remplissage de la cuve d’eau, ou commande de la résistance de chauffage par exemple) dans un ordre déterminé (notion de programme) en fonction du temps (durée de lavage) ou en fonction d’informations (cuve pleine, température atteinte).

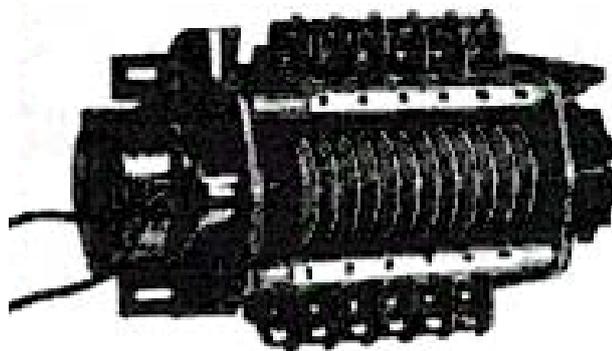


programmeur à cames
à minivannes pneumatiques et minirupteurs

- ① minirupteurs
- ② minivannes pneumatiques
- ③ accouplements rigides



programmeur avec bouton de réglage sur pupitre



Programmeur à cames fond d’armoire

Figure 3.12 : Différents programmeurs (doc.Crouzet)

-1-Principe

Un programmeur cyclique est constitué :

- d'un dispositif d'entraînement : motoréducteur,
- d'un support d'information : cames,
- d'éléments de sortie : minirupteurs ou minivannes.

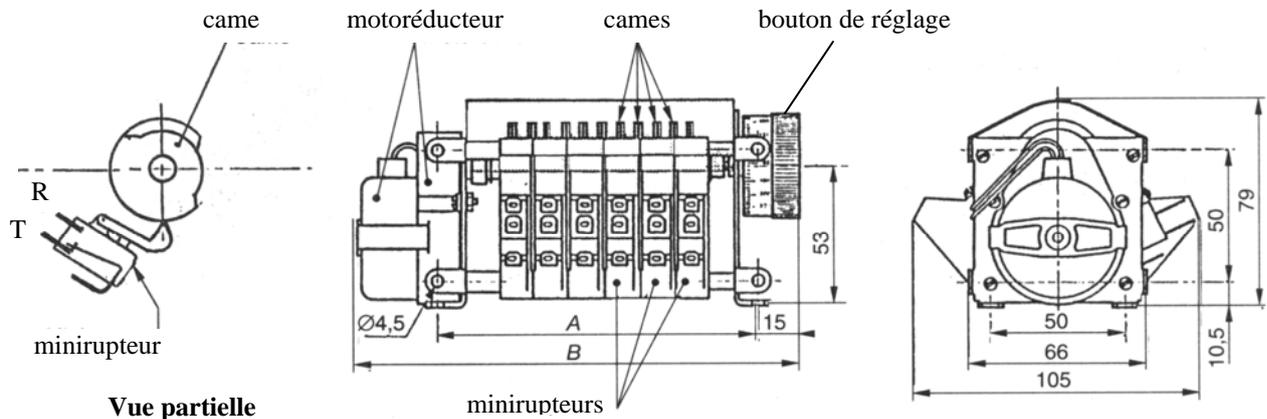


Figure 3.13 : Schéma d'un programmeur à bouton de réglage en façade (doc.Crouzet)

Un motoréducteur est l'association d'un moteur électrique et d'un réducteur. Ce dernier est constitué d'un ensemble d'engrenages (roues dentées) pour obtenir une réduction de la vitesse de rotation. La première roue est fixée sur l'axe du moteur, et sur la dernière roue est fixé un accouplement rigide permettant d'entraîner les cames à la vitesse désirée. Si par exemple on veut utiliser un programmeur avec un cycle de 20s (les cames font un tour en 20s), et que l'on dispose d'un moteur qui tourne à une vitesse de 3000tpm (soit un tour en 20ms), il faut définir le nombre de dents et de roues du réducteur pour obtenir un coefficient de réduction total de l'ordre de 1000.

Le programme est représenté par un diagramme linéaire. Celui-ci reproduit le profil développé théorique de chaque came en fonction de son contact repos ou travail, et de sa position en degrés. Un cycle correspond à un tour de came donc à 360 degrés.

Si par exemple la durée du cycle est de 1 mn pour 360 degrés, une encoche de 30 degrés représente un temps de $(1 \times 60s \times 30^\circ) / 360^\circ = 5$ secondes.

Inversement si l'on veut un temps de 17 secondes, il faudra une encoche (pour le contact repos R) ou une bosse (pour le contact travail T) de : $(360^\circ \times 17s) / 60s = 102^\circ$.

Certains minirupteurs ont un point intermédiaire, pour lequel aucun circuit n'est fermé.

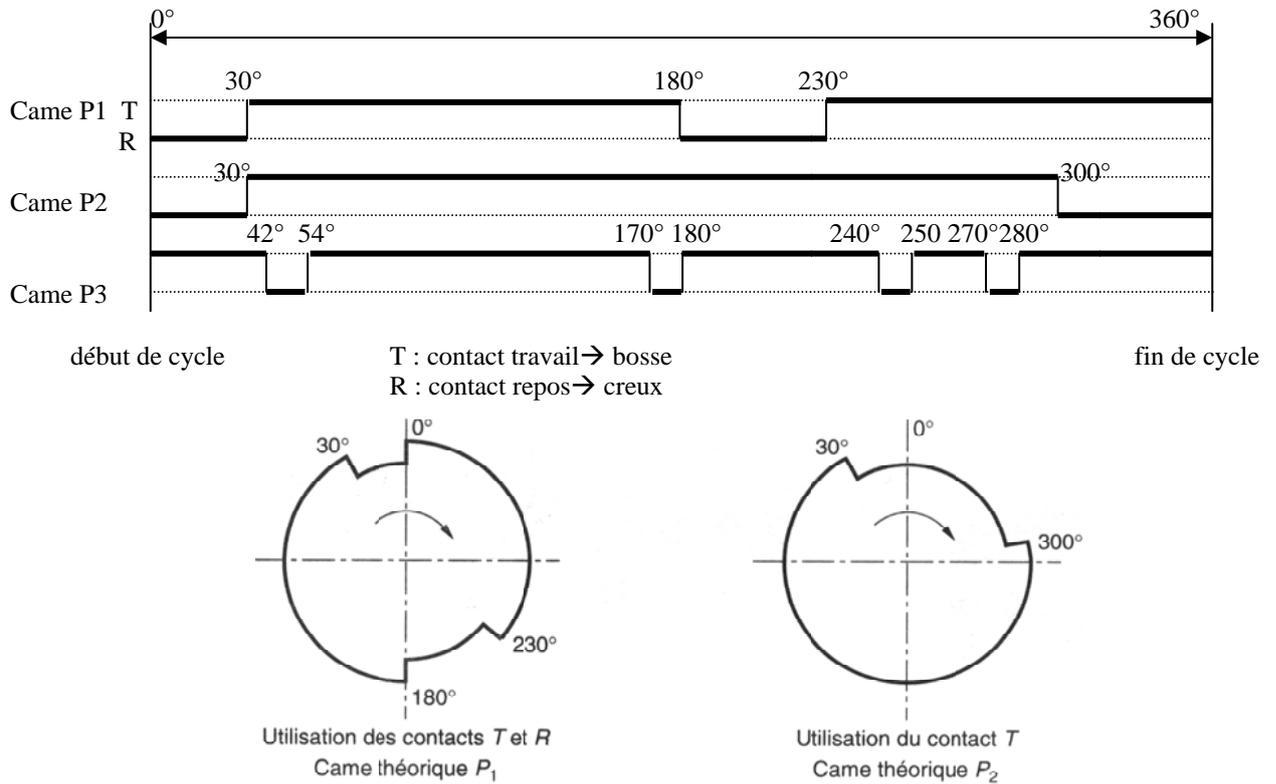


Figure 3.14 – Exemple de programmeur cyclique : programme linéaire et schémas théoriques des cames

-2- Caractéristiques d'un programmeur cyclique

- durées de cycle standards : de 2 secondes à 30 heures ;
- nombre de circuits de commande : de 1 à 44 circuits ;
- supports d'information : cames réglables taillées ou à taquets (cf. figure 3.15);
- sens de rotation : un ou deux sens de rotation ;
- nombre de moteurs : un ou deux ;
- nombre de vitesses : une ou deux ;
- bouton de réglage : en façade du pupitre ou en fond d'armoire.

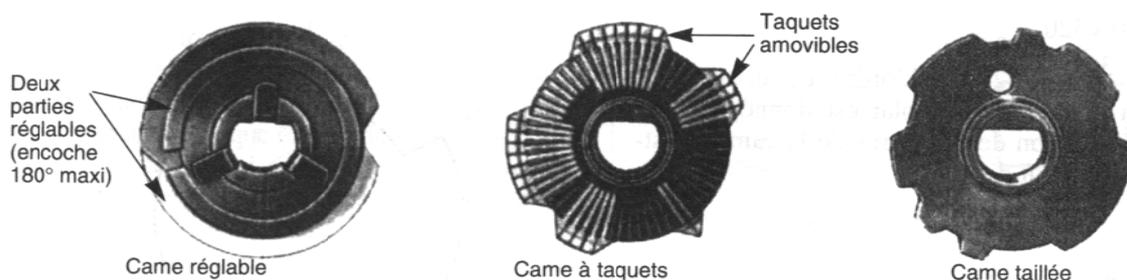


Figure 3.15 : différents types de cames (doc.Crouzet)

-3-Exemple

La figure 3.16 donne le diagramme incomplet du programmeur cyclique d'un distributeur de boissons chaudes. En fonction de ce diagramme :

Questions

- 1-Calculer le temps total de mixage eau + chocolat (came n°8).
- 2-A partir du démarrage du programmeur, calculer au bout de combien de temps sera distribuée la poudre de chocolat (came n°7).
- 3-Donner la forme théorique de la came n°7.
- 4-On donne la came théorique n°6 correspondant à l'eau pour café expresso, retrouver son tracé et ses temps sur le diagramme linéaire.

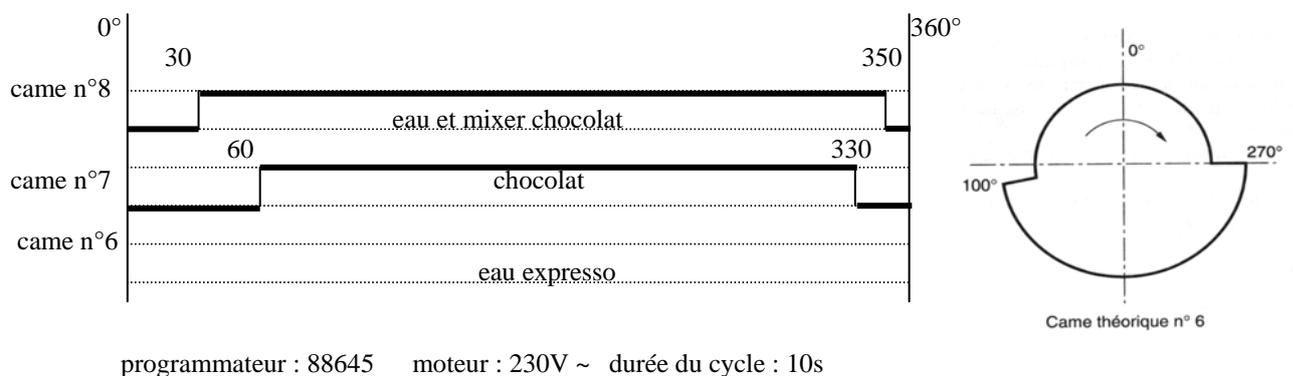


Figure 3.16 : doc Crouzet

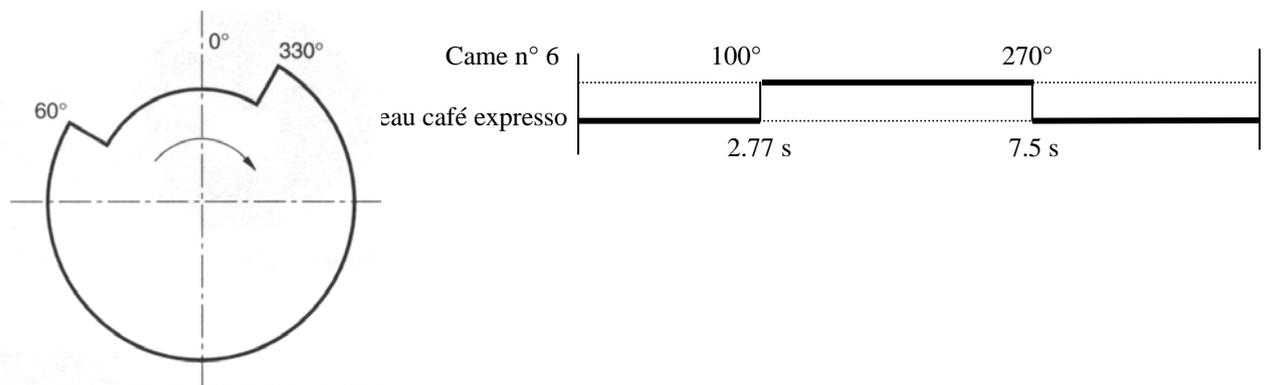
Réponses

1° Le temps de mixage eau + chocolat correspond au contact travail T activé pendant une rotation de 320°. Sachant que les comes font un tour complet en 10 secondes, la came 8 tourne de 320° en $(10 \times 320) / 360 = 8.88s$.

2° L'ordre de distribution de la poudre de chocolat est donné au bout d'une rotation de 60 degrés de la came n°7, c'est à dire au bout de : $(10 \times 60) / 320 = 1.66s$.

3° Came théorique n°7

4° Programme linéaire de la came n°6



CHAPITRE 4 : COMMANDE EN LOGIQUE CABLEE

-I-SYNTHESE DE LA PARTIE COMMANDE

Généralités et exemple	76
-1- Technologie pneumatique	77
-2- Technologie électrique	78
-3- Technologie électronique	80

-II- MATERIALISATION DE GRAFCET PAR DES SEQUENCEURS

-1- Correspondance grafcet-logigramme	82
-2- Cas des divergences et convergences	85
-a-Aiguillage	
-b-Séquences simultanées	

-III- LE SEQUENCEUR ELECTRONIQUE A BASCULES

-1- Grafcet linéaire ou à séquence unique	87
-2- Grafcets à séquences multiples	89
-3- Conclusions	92

-IV- LE SEQUENCEUR ELECTRIQUE

-1-Exemple et schéma de principe du module d'étape	93
-2-Représentation et disposition des E/S chez Télémécanique	94
-3-Séquenceur ou association de modules d'étapes	96
-4-Schéma de câblage électrique	96
-5-Réalisation de fonctions logiques dans le câblage de grafcet	99

-V- LE SEQUENCEUR PNEUMATIQUE

-1-Généralités	104
-2-Comparaison des techniques développées	105
-3-Mise en cascade des modules	106
-4-Câblage de grafcets à séquences multiples	108

REFERENCES BIBLIOGRAPHIQUES

- 1- F.Degoulange & al « Automatismes », classes de première et terminale E,F,BT- formation permanente, édit Dunod, Paris 1983.
- 2- J.Perrin & al «Automatique industrielle », édit Dunod, Paris 1982.
- 3- Y.Lecourtier & B.Saint-Jean « Introduction aux automatismes industriels », édit Masson 1989.
- 4- M.Pinot & al « Du grafcet au automates programmables », édit.Foucher, Paris 1986.

-I-SYNTHESE DE LA PARTIE COMMANDE

Effectuer la synthèse d'un automatisme, c'est passer du cahier des charges (généralement grâce au grafcet de niveau 2) au schéma de la partie commande. En logique câblée deux approches sont utilisées en fonction de la complexité du système automatisé et de la nature du matériel utilisé.

Dans la première on écrit d'abord les équations des sorties en utilisant les techniques classiques de la logique combinatoire (table de vérité, règles de Morgan, table de Karnaugh, etc...), puis on matérialise ces équations à l'aide de fonctions logiques dans la technologie utilisée. C'est cette technique qui sera développée dans ce paragraphe.

La deuxième est une méthode de synthèse directe basée sur le grafcet de niveau 2 et l'utilisation d'un séquenceur : on associe à chaque étape une fonction mémoire. La matérialisation du grafcet correspond donc à la réalisation pratique de l'automate, dans la technologie choisie (pneumatique, électrique ou électronique). Cette technique de matérialisation et de câblage automatiques sera développée au paragraphe II et utilisée dans les suivants.

Pour illustrer les différents modes de traitement de l'information en fonction de la technologie, on s'appuiera sur un exemple simple.

Exemple d'application

Un poste de montage comprend 3 magasins: le magasin V contient des vis, le magasin R des rondelles, le magasin E des écrous. L'opérateur dispose de 3 boutons poussoirs a, b, c:

- *- s'il appuie sur a, il obtient une vis;
- *- s'il appuie sur b, il obtient une vis et une rondelle;
- *- s'il appuie sur c, il obtient une vis, une rondelle et un écrou;
- *- toutes les autres combinaisons entraînent l'obtention d'une vis.

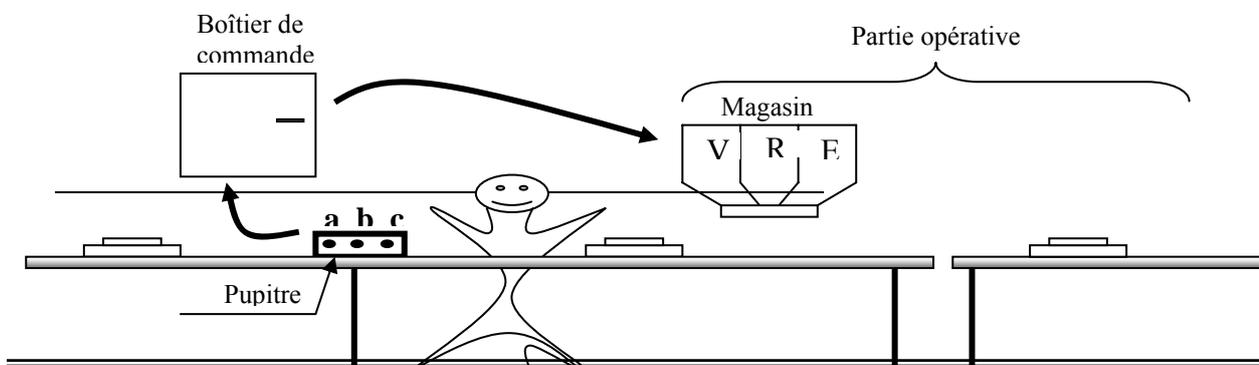


Figure 4.1 : Schéma synoptique de l'installation

Table de vérité

Equations des sorties

a	b	c	V	R	E
0	0	0	0	0	0
1	0	0	1	0	0
0	1	0	1	1	0
0	0	1	1	1	1
1	1	0	1	0	0
1	0	1	1	0	0
0	1	1	1	0	0
1	1	1	1	0	0

Cette table permet de déterminer les équations logiques des sorties :

(Vis) $V = a + b + c$

(Rondelle) $R = \bar{a} b \bar{c} + \bar{a} \bar{b} c$

(Ecroû) $E = \bar{a} \bar{b} c$

Partie opérative

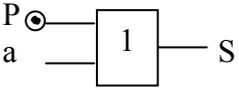
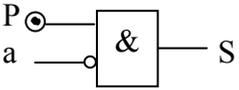
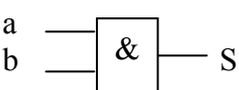
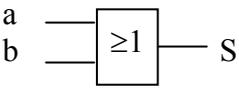
Trappes	V	R	E
Préactionneurs	M1	M2	M3
Actionneurs	C1	C2	C3

Remarque

On observe que dans les équations des sorties, les variables d'entrée apparaissent plusieurs fois. Par conséquent dans la partie commande d'un système automatisé on doit souvent multiplier les signaux qui sont issus des capteurs, qu'ils soient associés à la partie opérative ou à l'opérateur (pupitre). En technologie pneumatique cette condition est assurée par des piquages sur la conduite d'air comprimé dans le boîtier de commande. En technologie électrique la solution est l'utilisation de relais avec leurs multiples contacts.

-1-Technologie pneumatique

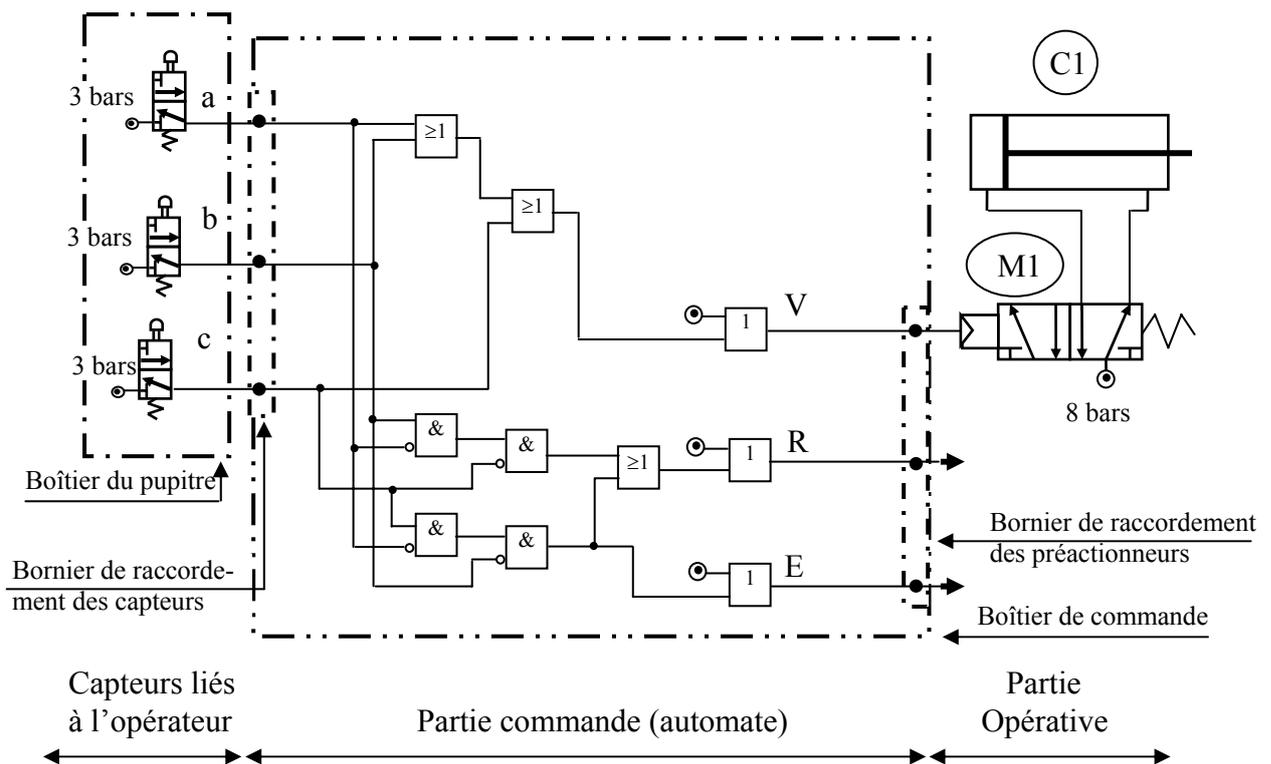
En technologie pneumatique le traitement de l'information se fait à base de modules appelés cellules. Elles sont au nombre de quatre : les cellules OUI, NON, ET et OU (cf. paragraphe IV chapitre 3).

Nom	Symbole	Sortie
OUI		$S = a$
NON		$S = \bar{a}$
ET		$S = a \cdot b$
OU		$S = a + b$

Application à l'exemple

- Partie pupitre : pour les boutons poussoirs ou capteurs liés à l'opérateur, on utilise des distributeurs 3/2 à commande manuelle et ressort de rappel.
- Partie opérative : les actionneurs sont des vérins double effet, et les préactionneurs des distributeurs 5/2 avec ressort de rappel.
- Partie commande : l'automate est réalisé à l'aide de cellules : 4 cellules NON (inhibition), 3 cellules OU, 3 cellules OUI.

Figure 4.2 : Schéma de câblage de l'automate en technologie pneumatique

-2-Technologie électrique

Cette technologie est basée sur les contacts électriques. Les fonctions OUI et NON sont assurées respectivement par des contacts ouvert ou fermé au repos. Les fonctions ET et OU sont réalisées par une mise en série ou en parallèle des contacts.

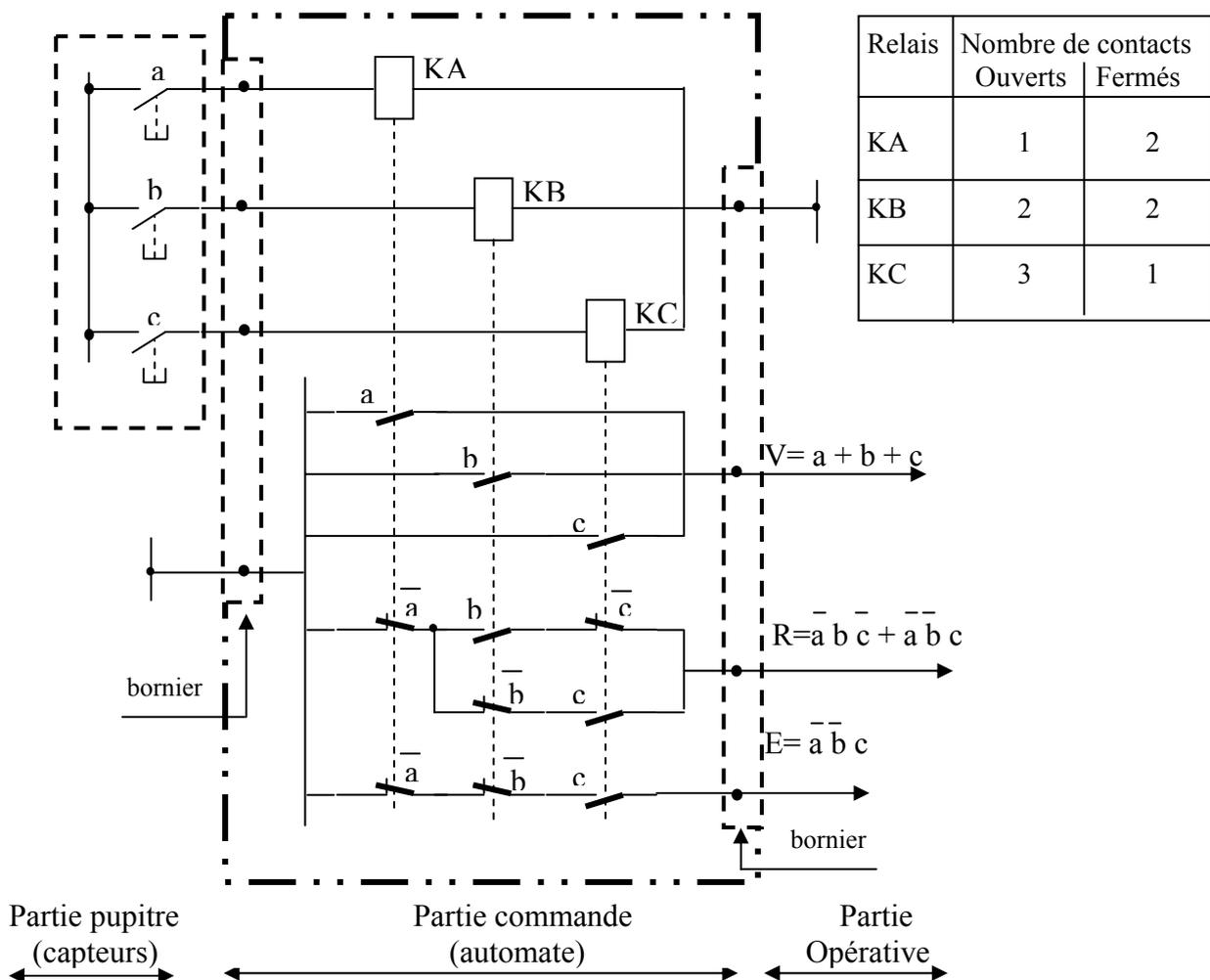
Nom	Symbole	Sortie
OUI		$L = a$
NON		$L = \bar{a}$

Nom	Symbole	Sortie
ET		$L = a \cdot b$
OU		$L = a + b$

Application à l'exemple

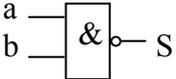
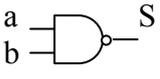
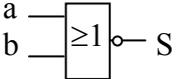
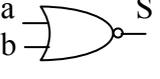
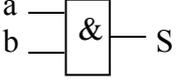
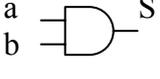
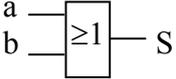
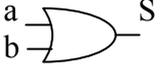
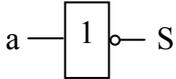
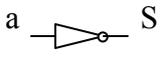
- Partie pupitre : pour les capteurs liés à l'opérateur, on utilise des boutons poussoirs électriques à contacts ouverts au repos.
- Partie opérative : les actionneurs sont des électro-aimants qui assurent directement l'ouverture des trappes.
- Partie commande : l'automate est réalisé à l'aide de relais instantanés, possédant chacun 4 contacts, et dont les bobines seront notées KA, KB et KC.

Figure 4.3 : Schéma de câblage de l'automate en technologie électrique



-3-Technologie électronique

En électronique les signaux sont traités par deux fonctions de base ET, OU, et par trois fonctions dérivées PAS, NOR, NAND. Le module de base est le transistor ou la porte logique en circuit intégré.

Nom	Symbole		Sortie
	Européen	Américain	
NAND			$S = \overline{a \cdot b}$
NOR			$S = \overline{a + b}$
ET			$S = a \cdot b$
OU			$S = a + b$
NON ou PAS ou INVERSEUR			$S = \overline{a}$

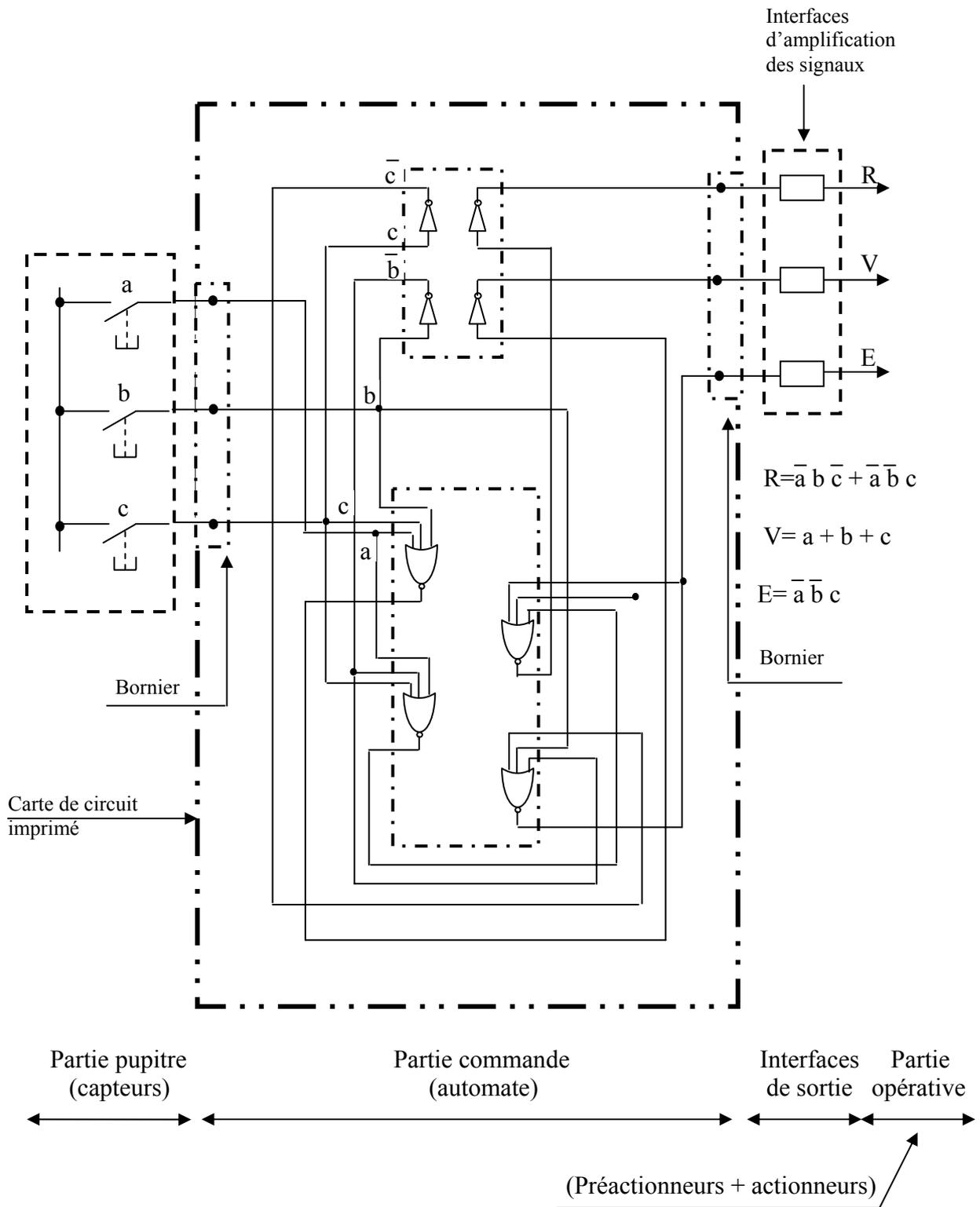
Application à l'exemple

- Partie pupitre : pour les capteurs liés à l'opérateur, on utilise des boutons poussoirs électriques à contacts ouverts au repos.
- Partie opérative : les actionneurs sont des électro-aimants qui assurent directement l'ouverture des trappes.
- Partie commande : l'automate est constitué d'une plaque de circuit imprimé, un circuit intégré quadruple porte NOR à trois entrées, un circuit intégré quadruple porte PAS (inverseur).

Remarque

Dans la technologie électronique, on a très souvent recours à des circuits d'amplification des signaux délivrés par l'automate (interfaces de sortie), ce qui n'est pas le cas dans la technologie électrique. En fonction de la nature des actionneurs, on peut utiliser en plus des préactionneurs.

Figure 4.4 : Schéma de câblage de l'automate en technologie électronique



-II- MATERIALISATION DE GRAFCET PAR DES SEQUENCEURS

C'est une méthode de synthèse directe basée sur le grafcet de niveau 2 : on associe à chaque étape une fonction mémoire, et à chaque séquence de grafcet un séquenceur. Même si le câblage est fonction de la technologie choisie (pneumatique, électrique ou électronique), le principe de réalisation pratique de l'automate reste le même.

-1-Correspondance grafcet-logigramme

Les règles de base de fonctionnement d'un grafcet indiquent que :

- une étape est activée par le franchissement de la transition précédente,
- une étape reste active tant que la transition suivante n'est pas franchie,
- l'action associée à une étape est effective tant que l'étape est active.

Une mémoire à deux entrées S (mise à un) et R (mise à zéro) est une image satisfaisante de l'étape. La mise à un de la mémoire est réalisée par la satisfaction simultanée (ET logique) de deux conditions : la mémoire précédente est active (à l'état logique 1) et la réceptivité associée à la transition entre elles est vraie. Ce qui est représenté sur la figure suivante par l'équation : $M_n(1) \equiv M_n \equiv S_n = M_{n-1} \times r_{n-1}$.

La remise à zéro de cette mémoire est réalisée par le franchissement de la transition suivante. Cela revient à dire que la mise à zéro de M_n n'a lieu que si la mémoire M_{n+1} et la réceptivité r_n sont à 1. On peut noter cela en écrivant $M_n(0) \equiv R_n = M_{n+1} \times r_n$.

Comme r_n est une condition de l'activité de M_{n+1} (M_{n+1} est mis à un ssi r_n est à 1), on peut simplifier l'expression de la remise à zéro de M_n par : $M_n(0) \equiv R_n = M_{n+1}$.

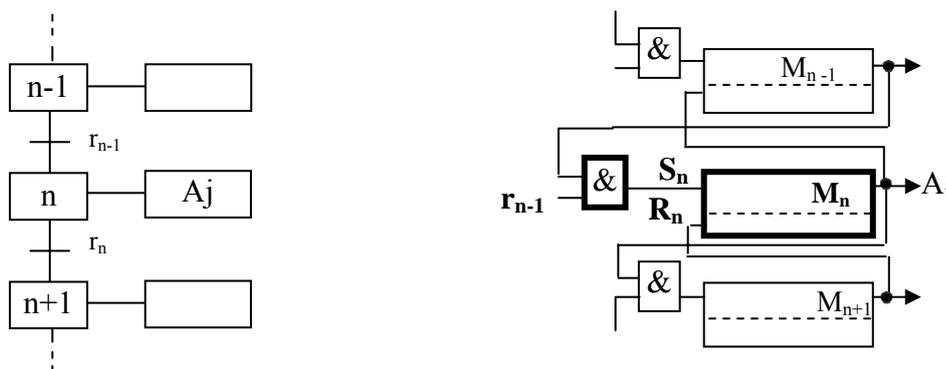


Figure 4.5 : Correspondance mémoire-étape

La simplification précédente n'est pas acceptable si dans le grafcet ainsi matérialisé on rencontre une boucle comportant moins de 3 étapes, car une même mémoire d'étape M_{n+1} est

à la fois étape d'entrée et étape de sortie de la mémoire M_n (cf Figure 4.6-a). Ceci rend la commande aléatoire car M_{n+1} intervient comme condition aussi bien dans la mise à un que la mise à zéro de M_n . Pour remédier à ce problème, deux possibilités sont à envisager pour la remise à zéro d'une mémoire:

- soit câbler la relation générale $M_n(0) \equiv R_n = M_{n+1} \times r_n$, (cf. Figure 4.6-c)
- soit utiliser la relation simplifiée $M_n(0) \equiv R_n = M_{n+1}$ et introduire une mémoire (donc une étape) auxiliaire « fictive » entre M_n et M_{n+1} . Cette mémoire auxiliaire (cf. figures 4.6-b & 4.6d) est identique à M_n avec comme unique différence une réceptivité d'activation égale à 1. Ceci ramène à trois le nombre d'étapes et interdit ainsi à une mémoire d'être présente à la fois sur la mise à un (Si) et la mise à zéro (Ri) d'une autre mémoire.

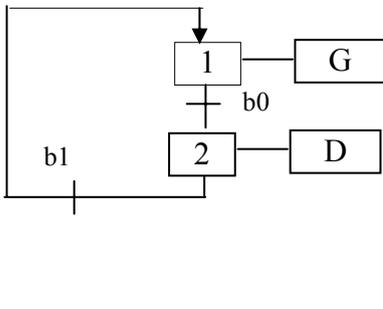


Figure 4.6-a : Boucle avec 2 étapes

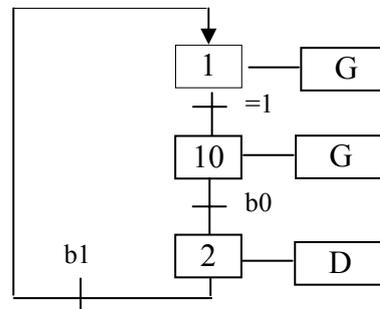


Figure 4.6-b : Introduction d'une étape fictive

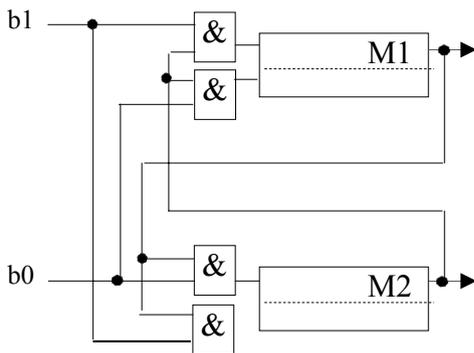


Figure 4.6-c : Réalisation générale

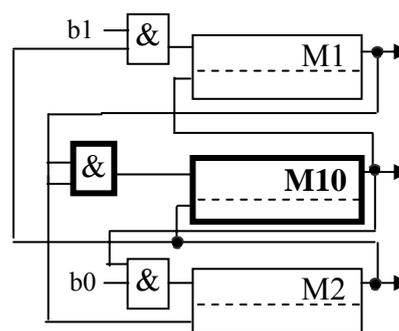


Figure 4.6-d : Réalisation par étape « fictive »

Le problème qui reste à traiter est celui des étapes initiales : en effet comme leur nom l'indique, à la mise sous tension elles sont mises à un de manière inconditionnelle, alors que

toutes les autres étapes sont mises à zéro. On utilisera pour cela une variable secondaire d'initialisation I , qui apparaîtra dans les variables S_i (mise à 1) des étapes initiales, et dans les variables R_i (mise à zéro) des autres étapes.

En résumé les règles de matérialisation d'un grafcet linéaire à l'aide de mémoires d'étapes peuvent s'énoncer ainsi :

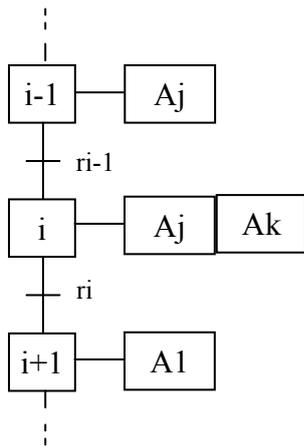


Figure 4.7 : Séquence de grafcets à implanter

sortie X_i . Si l'étape i est active, la sortie X_i est égale à 1, si l'étape i est inactive $X_i=0$.

- *Règle2* : La mise à 1 de X_i est assurée par l'équation logique $S_i = X_{i-1} \cdot r_{i-1}$, où r_{i-1} désigne la réceptivité associée à la transition reliant l'étape $i-1$ à l'étape i .

- *Règle3* : La remise à zéro de X_i est assurée par l'équation logique $R_i = X_{i+1} + I$, où I désigne la commande servant à l'initialisation du « séquenceur » (donc du grafcet correspondant).

- *Règle4* : Si M_j est une mémoire correspondant à une étape initiale, I sera appliqué sur S_j et non sur R_j :

$$S_i = X_{i-1} \cdot r_{i-1} + I.$$

- *Règle5* : Une sortie de l'automate pour commander une action A_j sera réalisée à l'aide de la somme logique des sorties X_k des mémoires correspondant aux étapes où A_j doit être active (ainsi dans l'exemple de la Figure 4.7, $A_j = X_{i-1} + X_i$).

- *Règle1* : A chaque étape i on associe une mémoire ayant pour

-2-Cas des divergences et convergences

Que l'on ait des séquences simultanées ou un aiguillage (sélection de séquence, saut d'étape, reprise en séquence), la situation est toujours la même : une étape peut avoir plusieurs étapes d'entrée ou de sortie.

-2-a-Aiguillage

Dans ce cas les étapes communes d'entrée ou de sortie d'une étape possèdent des réceptivités différentes. On aura une divergence et une convergence en OU. Les règles 2 et 3 sont modifiées ainsi:

- Règle2b : En considérant toutes les étapes i-1 d'entrée de l'étape i, la mise à 1 de Xi est assurée par l'équation logique $S_i = \sum(X_{i-1} \cdot r_{i-1})$, où r_{i-1} désigne la réceptivité associée à la transition reliant chaque étape d'entrée i-1 à l'étape i.
- Règle3b : En considérant toutes les étapes i+1 de sortie de l'étape i, la remise à zéro de Xi est assurée par l'équation logique $R_i = \sum X_{i+1} + I$, où I désigne la commande servant à l'initialisation du « séquenceur » (donc du grafctet correspondant).

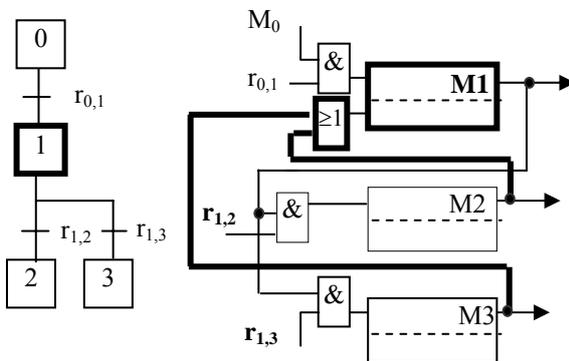


Figure 4.8-a : Divergence en OU

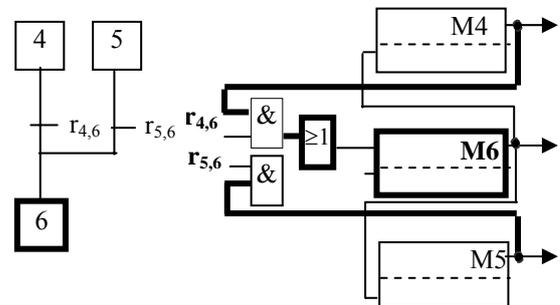


Figure 4.8-b : Convergence en OU

-2-b-Séquences simultanées

Comme pour l'aiguillage, ici également on a une étape (l'entrée des séquences simultanées) avec des sorties multiples et une étape (la sortie des séquences simultanées) avec des entrées multiples.

On continuera donc d'utiliser les règles 2b et 3b précédemment définies pour l'aiguillage, avec cependant un produit à la place de la somme, car nous avons une convergence et une divergence en ET à la place d'une convergence et une divergence en OU.

Pour les étapes d'entrée/sortie des séquences simultanées, les règles 2 et 3 se réécrivent ainsi :

- *Règle2c* : Pour l'étape i d'entrée des séquences simultanées, la mise à 0 de X_i est assurée par l'équation logique $S_i = \prod(X_{i+1} \cdot r_i) + I$, où r_i désigne la réceptivité associée à la transition reliant l'étape i à chacune de ses étapes de sortie $i+1$ et I la commande servant à l'initialisation du « séquenceur ».
- *Règle3c* : Pour l'étape i de sortie des séquences simultanées, la mise à un de X_i est assurée par l'équation logique $R_i = \prod X_{i-1} \cdot r_{i-1}$, où r_{i-1} désigne la réceptivité associée à la transition reliant l'étape i à ses étapes d'entrée X_{i-1} .

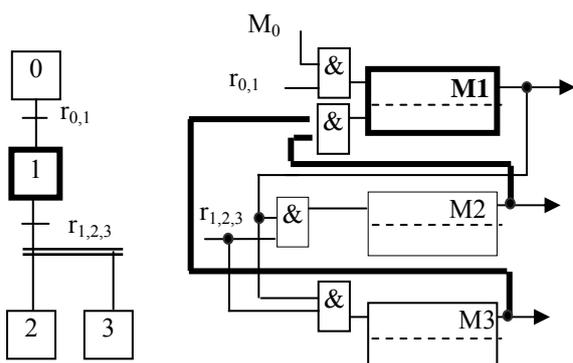


Figure 4.9-a : Divergence en ET

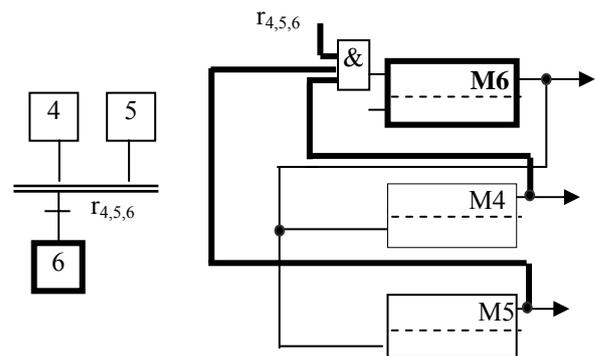


Figure 4.9-b : Convergence en ET

-III-LE SEQUENCEUR ELECTRONIQUE CABLE A BASCULES

Dans les chapitres précédents, on a étudié la théorie du grafcet permettant de modéliser la commande d'un automate grâce à un graphe basé sur les notions d'étape et de transition.

Ensuite on a vu une méthode de synthèse de l'automate basée sur l'association étape-mémoire : on a associé à chaque étape du grafcet une variable X_i qui est égale à 1 si l'étape est active, et à 0 si elle est inactive. Cette valeur est mémorisée dans une mémoire d'étape.

Ici on s'intéresse à la réalisation pratique de l'automate. On commence par réaliser la mémoire d'étape à l'aide d'une bascule: on associe la sortie Q_i d'une bascule SR (ou JK) à chaque variable X_i symbolisant une étape. Puis on effectue la synthèse directe de l'automate en matérialisant le grafcet par un « séquenceur » câblé. Cela consiste essentiellement à calculer et câbler les entrées (S_i, R_i) de ces bascules en utilisant les règles énoncées précédemment. Comme les actions associées à une étape s'exécutent dès que l'étape est active, il est alors aisé de déduire les sorties de l'automate (actions) en fonction des X_i .

-1-Cas du grafcet linéaire ou à séquence unique

Pour préciser la méthode on se basera sur l'exemple classique d'une perceuse munie d'un moteur de rotation de broche à deux vitesses (PV et GV), et d'un moteur de translation verticale à deux sens (D et M). Le grafcet de niveau 2 de la commande est donné par la figure 4.10-b.

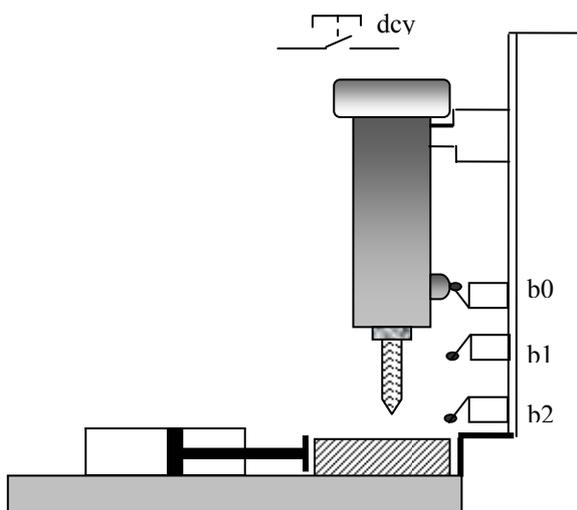


Figure 4.10-a : Tête de perçage

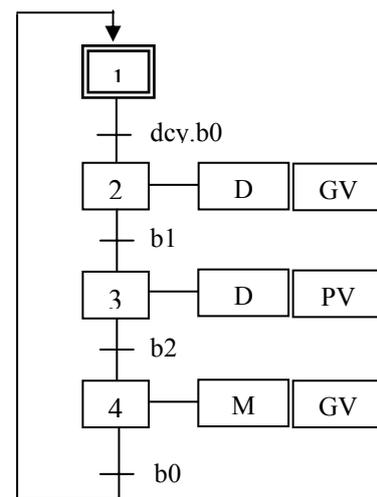


Figure 4.10-b : Grafcet du mouvement de la tête

Comme le grafcet comprend 4 étapes, il faudra 4 bascules pour le matérialiser. On utilisera des bascules RS. Au lieu de détailler le raisonnement pour toutes les étapes, on se contentera d'étudier le fonctionnement de la bascule associée à l'étape 2. Ce raisonnement est basé sur les règles d'évolution du grafcet et le principe de fonctionnement de la bascule RS.

La sortie X2 est à 1 (étape 2 active) si X1 = 1 (étape 1 active) et si dcy.b0 = 1 (réceptivité vraie). L'entrée S2 de mise à 1 de la bascule B2 est donc commandée par le produit X1.dcy.b0 : $S2 = X1 \cdot dcy \cdot b0$ (ce qui est conforme à la règle n°2 du §2 page 81).

L'étape 2 est désactivée (bascule remise à zéro) lorsque l'étape 3 devient active, c'est à dire par le passage à 1 de X3. L'entrée R2 de remise à zéro de la bascule B2 est donc commandée par X3 : $R2 = X3$ (ce qui est conforme à la règle n°3 du §2 page 81).

Pour obtenir les équations des entrées Si et Ri des autres bascules, on peut soit répéter le raisonnement précédent, soit **appliquer directement les règles numéros 2 et 3** (de la page 81).

Le problème délicat est celui des étapes initiales (ici il y en a une seule). En effet à la mise sous tension, les sorties des bascules vont prendre des valeurs aléatoires. On désire mettre à un l'étape initiale, et à zéro toutes les autres étapes. On utilisera pour cela une variable secondaire d'initialisation I, qui apparaîtra dans la variable S1 (mise à 1) de l'étape initiale, et dans les variables Ri (mise à zéro) des autres étapes. Cette variable sera câblée par le biais d'une porte OU de façon à effectuer manuellement ce forçage à un ou à zéro des différentes bascules. Pour être sûr de la remise à zéro, on prendra des bascules à déclenchement prioritaire. Actuellement dans les circuits évolués la commande I est assurée automatiquement.

On obtient finalement le tableau suivant qui résume les fonctions de commande des bascules, et le schéma de câblage correspondant de la Figure 4.10.

$$\begin{array}{ll} S1 = X4 \cdot b0 + I & R1 = X2 \\ S2 = X1 \cdot dcy \cdot b0 & R2 = X3 + I \\ S3 = X2 \cdot b1 & R3 = X4 + I \\ S4 = X3 \cdot b2 & R4 = X1 + I \end{array}$$

Il reste à étudier la matérialisation des sorties de l'automate (actions). Pour chaque action, on relie par des portes OU les sorties Xi associées aux étapes pour lesquelles ces actions sont lieu. Ainsi par exemple la commande du préactionneur D a lieu aux étapes 2 et 3. Par conséquent l'équation de D est : $D = X2 + X3$.

En procédant de la même manière on trouve : $PV = X3$, $M = X4$, $GV = X2 + X4$.

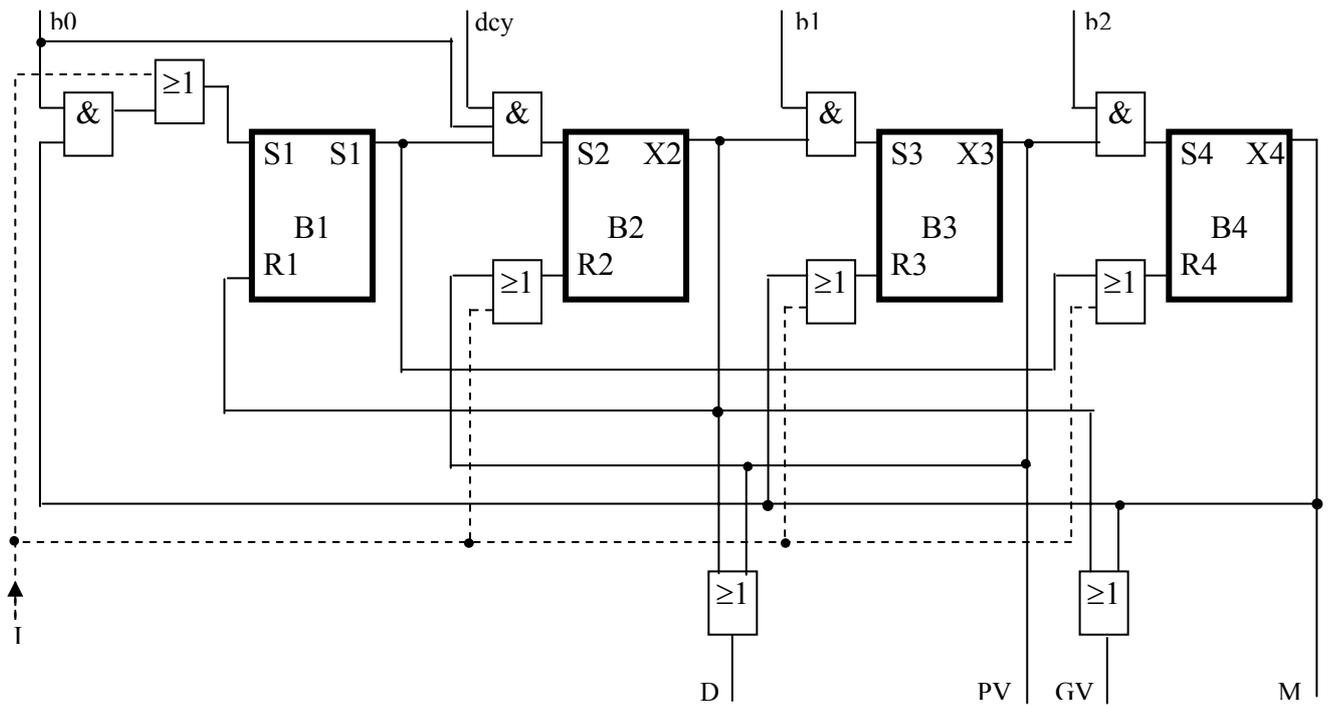


Figure 4.10-c : Schéma d'implantation du grafcet de la figure 4.10

-2-Cas du grafcet à séquences multiples

-2-a-Aiguillage

Dans ce cas les étapes communes d'entrée ou de sortie d'une étape possèdent des réceptivités différentes.

L'étape 1 de l'exemple de la Figure 4.11 possède deux étapes d'entrée: 0 et 3 (convergence en OU). Par conséquent, conformément à la règle 2b du §2 page 82, dans l'expression S1 de mise à 1 de la bascule B1 doivent apparaître X0 et X3 avec leurs réceptivités respectives, reliées par la fonction logique OU. De même, conformément à la règle 3b, X1 apparaît dans les expressions Ri de mise à zéro des bascules B0 et B3.

$$S1 = X0.m.d0 + X3.b0.d0 \quad R0 = X1 + \dots \quad R3 = X1 + \dots$$

L'étape 3 possède deux étapes de sortie : les étapes 0 et 1 (divergence en OU). Par conséquent dans les expressions Si de mise à 1 des bascules B0 et B1 doit apparaître X3 avec les réceptivités respectives à chaque bascule (règle 2b). De même dans l'expression R3 de mise à zéro de la bascule B3 doivent apparaître X0 et X1, reliées par la fonction logique OU (règle 3b).

$$S1 = X3.b0.d0 + \dots \quad S0 = X3.b0.d0 + \dots \quad R3 = X0 + X1$$

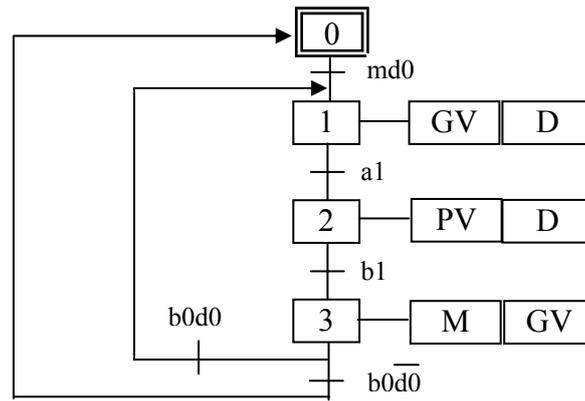


Figure 4.11-a : Exemple de grafcet avec reprise en séquence

Les équations des actions et des entrées des bascules sont les suivantes :

$$\begin{aligned}
 S0 &= X3.b0.d0 + I & R0 &= X1 \\
 S1 &= X0.m.d0 + X3.b0.d0 & R1 &= X2 + I \\
 S2 &= X1.a1 & R2 &= X3 + I \\
 S3 &= X2.b1 & R3 &= X0 + X1 + I \\
 D &= X1 + X2 & M &= X3 & GV &= X1 + X3 & PV &= X2
 \end{aligned}$$

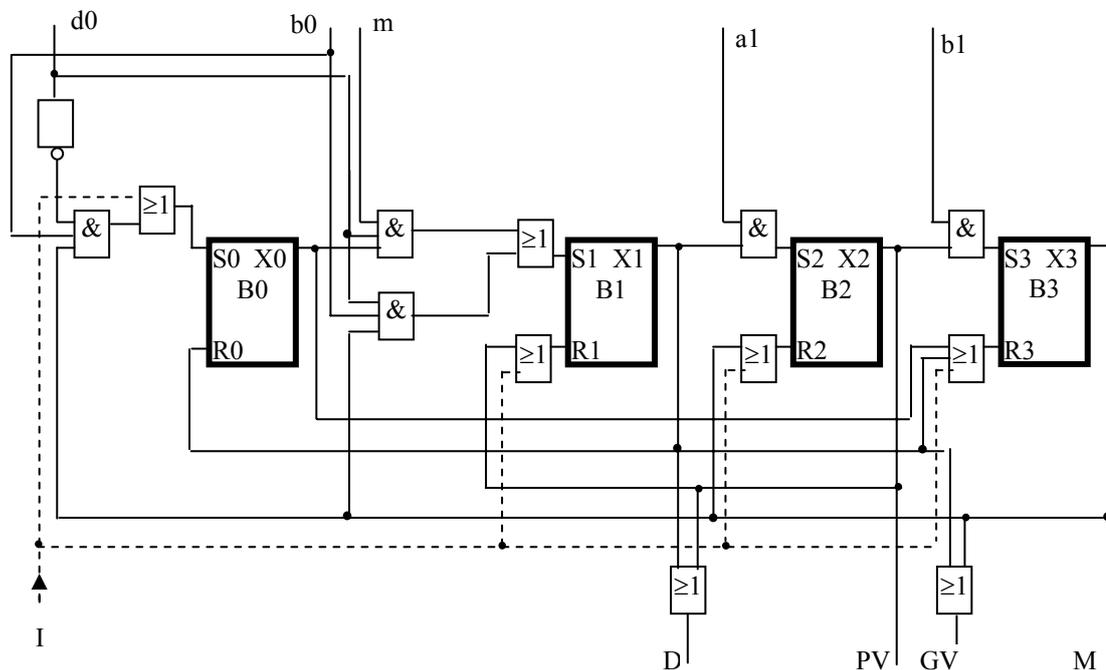


Figure 4.11-b : Schéma d'implantation du grafcet avec aiguillage

-2-b-Séquences simultanées

Par définition on entre dans des séquences simultanées et on en sort par une étape, une transition et une réceptivité uniques. Par conséquent, contrairement au cas de l'aiguillage, l'étape d'entrée des séquences simultanées possède des étapes de sortie qui ont toutes la même réceptivité d'activation. De même l'étape de sortie des séquences simultanées possède des étapes d'entrée qui ont toutes la même réceptivité de désactivation.

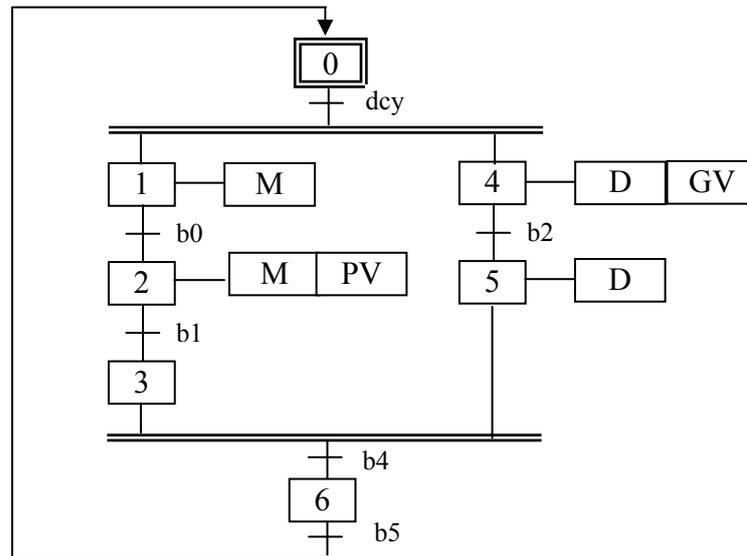


Figure 4.12-a : Exemple de grafcet avec séquences simultanées

Pour le câblage, la convergence en ET se traduit par le fait que la bascule B0 associée à l'étape 0 n'est désactivée que si les deux bascules B1 et B4 sont actives toutes les deux (fonction ET). De même la divergence en ET se traduit par le fait que la bascule B6 associée à l'étape 6 n'est activée que si les deux bascules B3 et B5 sont actives toutes les deux (fonction ET).

Ainsi les équations des bascules B0 et B6, associées aux étapes 0 et 6 de l'exemple de la Figure 4.91, s'écrivent comme suit (conformément aux règles 2c et 3c du §2 page 83) :

$$S0= X6.b5 + I \quad R0= X1.X4 \quad S6= X3.X5.b4 \quad R6= X0 + I$$

On remarque sur le grafcet que l'on a trois séquences : la séquence principale qui comprend les étapes 0 et 6 et les deux séquences simultanées. Par conséquent on réalisera trois séquenceurs qui seront câblés séparément, puis on réalisera les connexions entre le séquenceur principal et ceux des séquences simultanées. Cette connexion se fera notamment par

l'introduction de deux portes ET : l'une dans la désactivation de la bascule B0 (divergence en ET) et l'autre dans l'activation de la bascule B6 (convergence en ET).

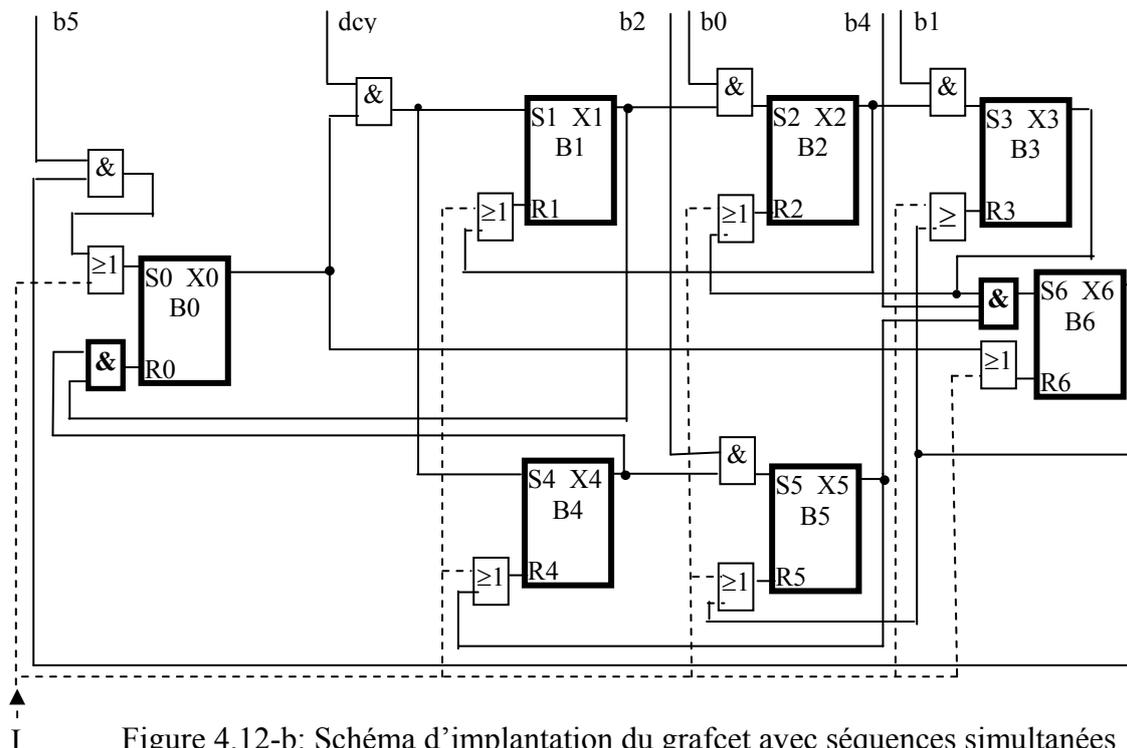


Figure 4.12-b: Schéma d'implantation du grafset avec séquences simultanées

(pour ne pas charger le dessin les actions ne sont pas représentées)

3-Conclusions

L'intérêt principal de la méthode de matérialisation d'un grafset par un séquenceur électronique câblé à base de bascules, est la grande facilité de conception et de maintenance de l'automate réalisé, car elle aboutit à un câblage qui traduit fidèlement la structure du grafset.

Son inconvénient découle de son principe même : le fait d'utiliser une bascule par étape rend très délicats le câblage et la maintenance pour des grafset complexes comportant plusieurs séquences multiples.

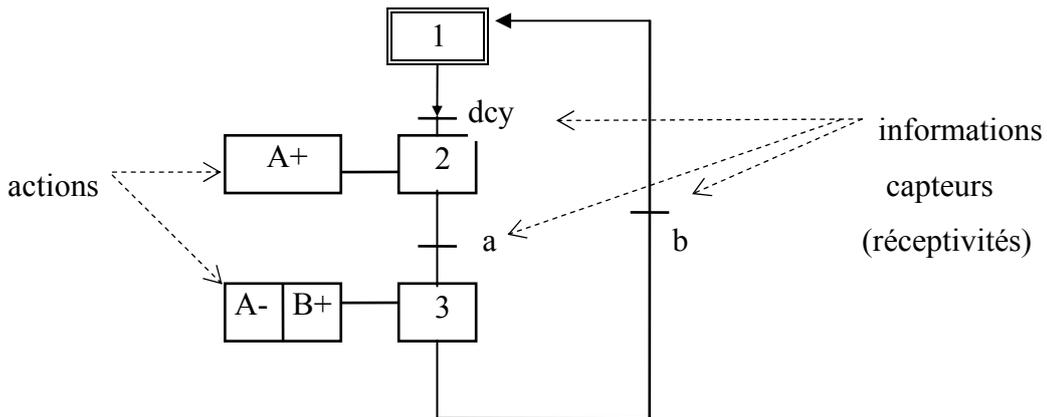
Pour des installations de taille moyenne, des composants électriques ou pneumatiques ont été développés industriellement. Leur principe de fonctionnement est identique à celui de la bascule RS : ils réalisent la fonction mémoire d'étape (bascule RS) par le biais d'une fonction logique OU sur les entrées de remise à zéro, et une fonction ET sur les entrées Si de mise à 1. Des embases particulières facilitent les connexions entre ces composants et la réduction du câblage. Il s'agit des modules d'étapes pour séquenceurs électriques ou pneumatiques, qui feront l'objet des paragraphes suivants.

-IV-LE SEQUENCEUR ELECTRIQUE

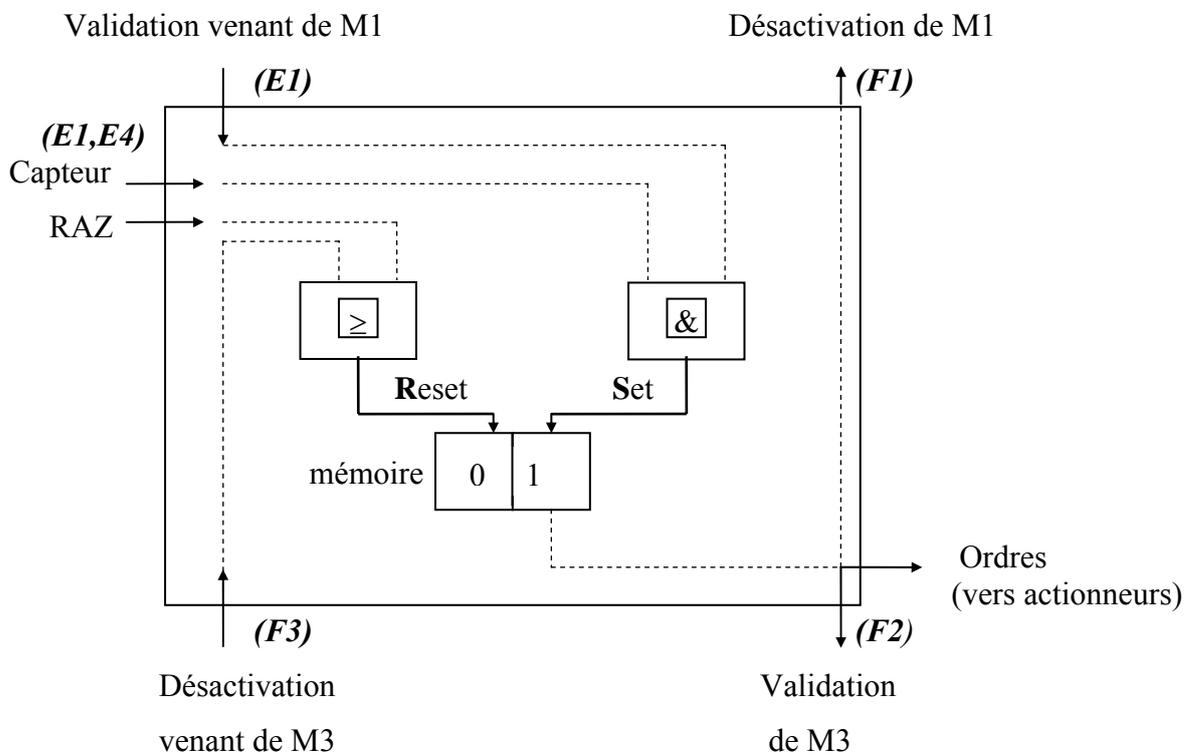
C'est un câblage séquentiel de dispositifs de commande appelés mémoires de phases ou modules de phase ou également « modules d'étapes » (car à chaque étape numéro i d'un grafcet est associé un module de phase Mi du séquenceur).

-1-Exemple et schéma de principe du module d'étape

-1-a-Exemple de grafcet



-1-b-Schématisation du module d'étape numéro 2 (M2)



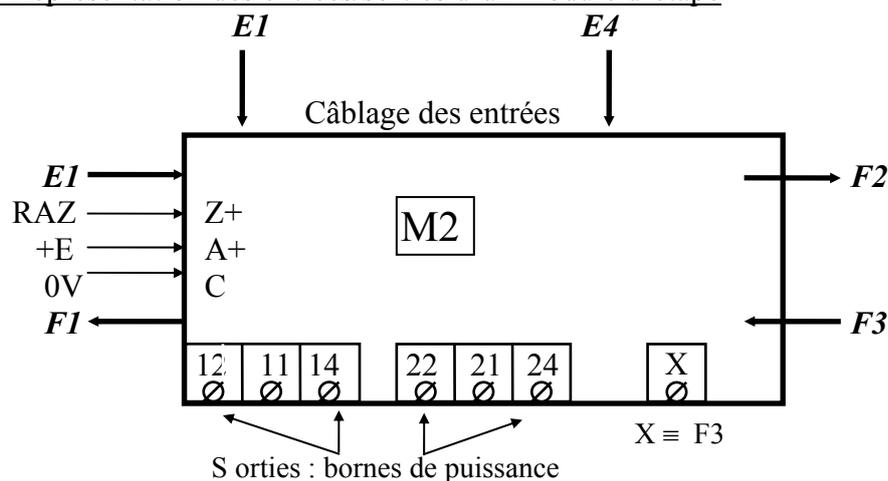
Remarque: La mémoire correspond à une bascule RS symbolique, dont les entrées de commande R et S correspondent aux sorties des portes logiques OU et ET. Quant à ses deux sorties \bar{Q} et Q, elles correspondent aux deux rectangles qui donnent les états 0 et 1 de la mémoire.

-1-c-Cas du séquenceur électrique

Le schéma de principe précédent est valable pour les séquenceurs pneumatique et électrique. Dans ce dernier cas, la mémoire de la bascule RS est réalisée à l'aide de deux bobines d'électroaimant pour la commande d'un relais bistable. Les sorties logiques \bar{Q} et Q correspondent aux états des contacts repos et travail du relais (voir le détail du relais au paragraphe IV-1-b-2 du chapitre 3)

-2- Représentation et disposition des E/S chez Télémécanique :

-2-a- Représentation des entrées/sorties d'un module d'étape



Remarque : Les bornes d'entrée 11 et 21 sont alimentées uniquement si on utilise les sorties.

-2-b- Schéma simplifié de fonctionnement interne d'un module d'étape

Le ET logique du schéma de principe sera réalisé par une mise en série du signal de validation venant du module précédent et des informations venant des capteurs. Le OU logique sera réalisé par une mise en parallèle de l'information de remise à zéro et du signal de désactivation venant du module suivant. On insérera des diodes de protection là où il sera nécessaire de protéger les sorties contre les courants de retour.

Les sorties des fonctions logiques ET et OU du schéma de principe attaqueront deux bobines d'électroaimant d'un relais bistable (mémoire logique du module). Les contacts repos correspondent aux sorties 12 et 22, quant aux contacts travail (qui permettent de commander les préactionneurs) ils correspondent aux sorties 14 et 24.

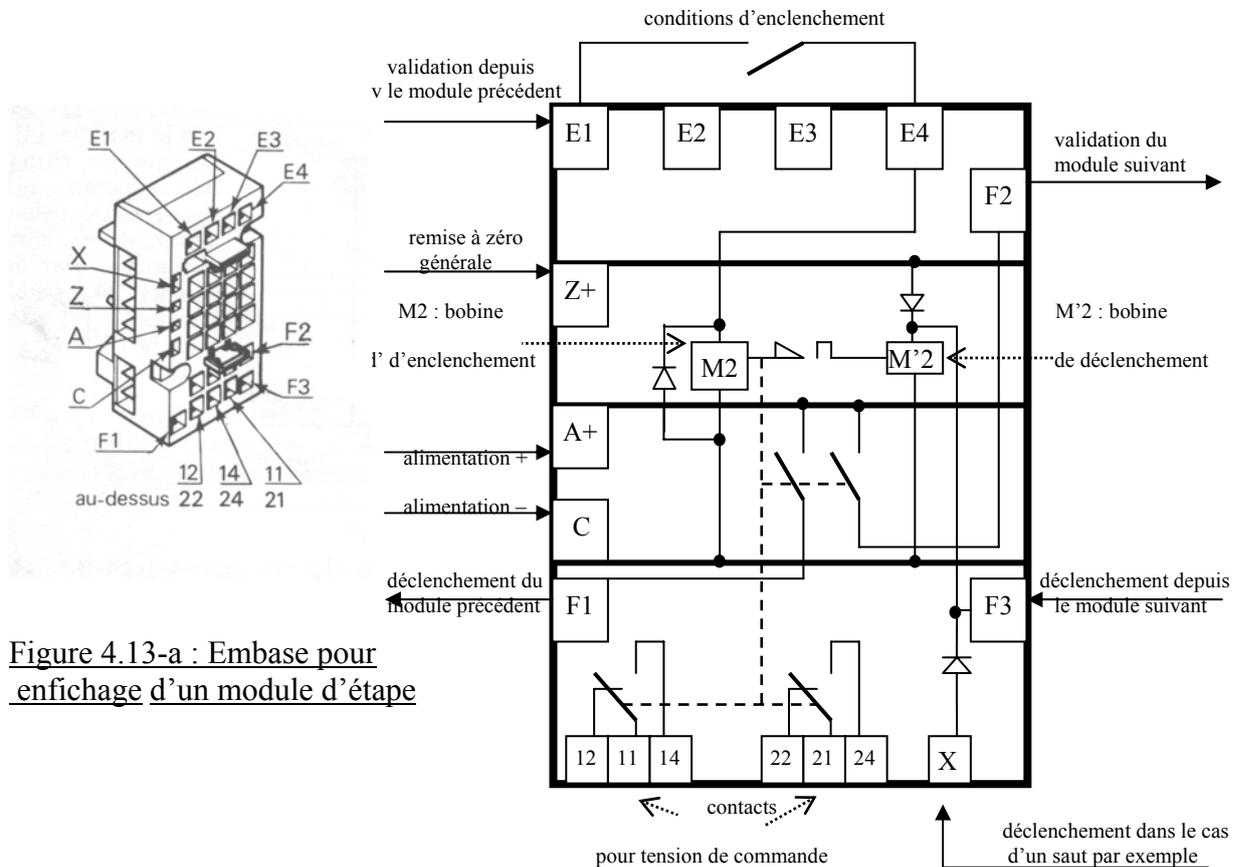


Figure 4.13-a : Embase pour enfichage d'un module d'étape

Figure 4.13-b : Schéma interne d'un module d'étape (doc Télémécanique)

Principe de fonctionnement

Quand la bobine est alimentée par le signal entrant par la borne E4, elle commande la fermeture des contacts du relais. Les jonctions 11-14 et 21-24 sont alors réalisées : le signal entrant par la borne 11 (ou 21) se retrouve en sortie sur la borne 14 (ou 24).

Le module d'étape est alimenté (en A.C. ou D.C.) sous une tension de 24, 48, 110 ou 220 volts. Comme pour un relais classique il ne faut surtout pas confondre la valeur de cette tension qui est généralement de 24V (présente en A+ et sur les entrées/ sorties Z+, E1, E4, F3, X, F2,

F1) avec la tension d'utilisation qui est généralement de 220V ou 380 V (présente sur les entrées 11 et 21 et par conséquent sur les sorties 12,14,22 et 24).

-3-Séquenceur ou association de modules d'étapes

Le séquenceur est conçu sous forme de modules relais (ou modules d'étapes) montés sur des embases fixées sur des racks, et le fonctionnement est le même que si on les enfichait les uns dans les autres. Ainsi seul le premier module a besoin d'être alimenté, et les connexions entre modules sont automatiquement réalisées (par « enfichage interne »). Seul le câblage des entrées (informations capteurs) et des sorties (commandes des préactionneurs) a besoin d'être réalisé.

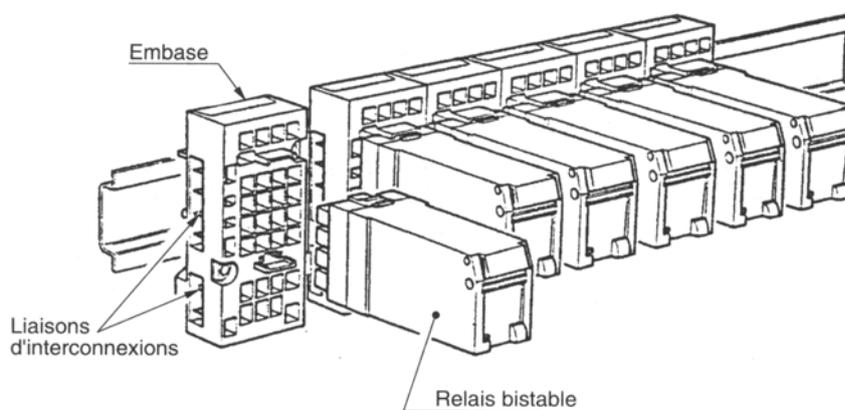


Figure 4.14 : Séquenceur électrique câblé

-4-Schéma de câblage électrique

La représentation du schéma électrique de commande à base de séquenceurs, est facilitée par l'utilisation du grafcet. Ainsi à chaque étape d'un grafcet, il suffit de faire correspondre un module d'étape. Les actions seront câblées à partir des sorties du module, et les informations capteurs (« réceptivités ») seront câblées aux entrées du module. Les modules d'étapes seront reliés directement les uns aux autres (par enfichage) pour constituer le séquenceur. Pour le premier et le dernier module du séquenceur, une boucle de recyclage assure la validation du premier module par le dernier, et une boucle de déclenchement assure la désactivation du dernier module par le premier.

On réalisera donc deux schémas : sur l'un on fera apparaître la connexion des modules d'étapes et le câblage des entrées (capteurs), ce qui représente le séquencement du grafcet. Sur l'autre on fera apparaître les sorties (préactionneurs) ainsi que les alimentations.

Les figures des pages suivantes donnent ces deux schémas pour l'exemple du grafcet donné au paragraphe V-1.

Remarques

-1-En début d'utilisation du séquenceur, il faut initialiser manuellement le premier module. Il faut donc prévoir un bouton poussoir d'initialisation.

-2-Un bouton poussoir de remise à zéro de tous les modules du séquenceur peut également être prévu, en insérant ce bouton entre la borne plus (+) de l'alimentation et la borne Z+ du séquenceur.

-3-Un séquenceur doit avoir au minimum trois modules d'étapes pour être utilisable. Sinon avec deux modules, chacun serait en même temps activé et désactivé par l'autre.

-4-Le nombre de séquenceurs à utiliser est égal au nombre de branches du grafcet, chacun comprenant autant de modules qu'il y a d'étapes dans la séquence.

Figure 4.15-a : Schéma détaillé de câblage du séquenceur de l'exemple page 82

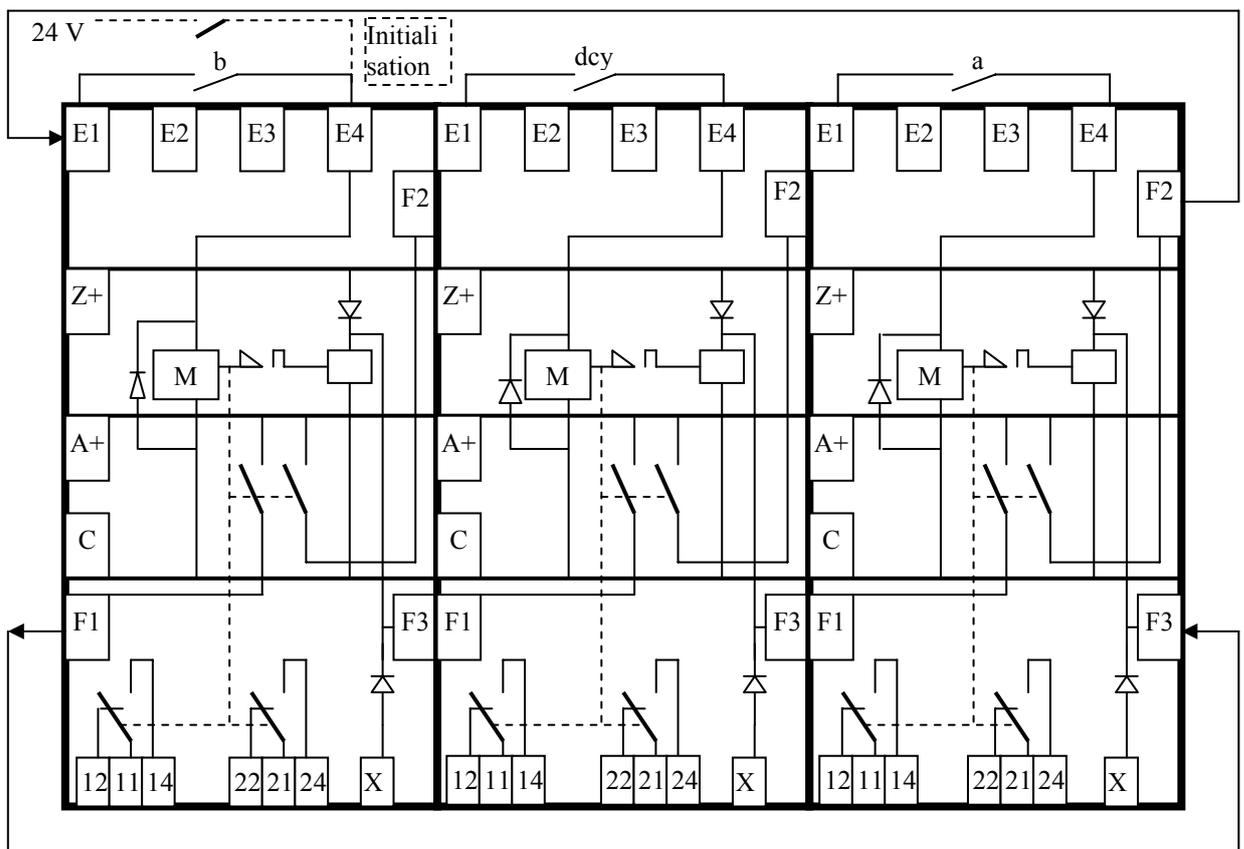
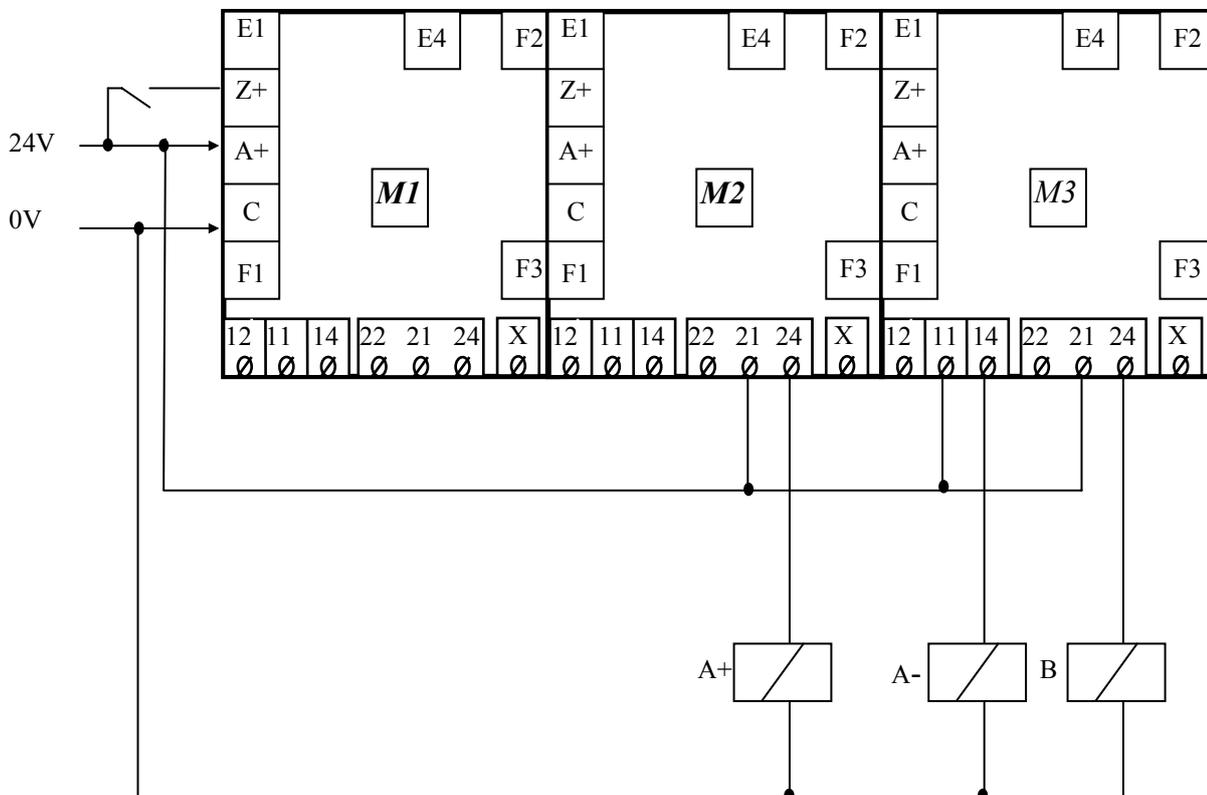


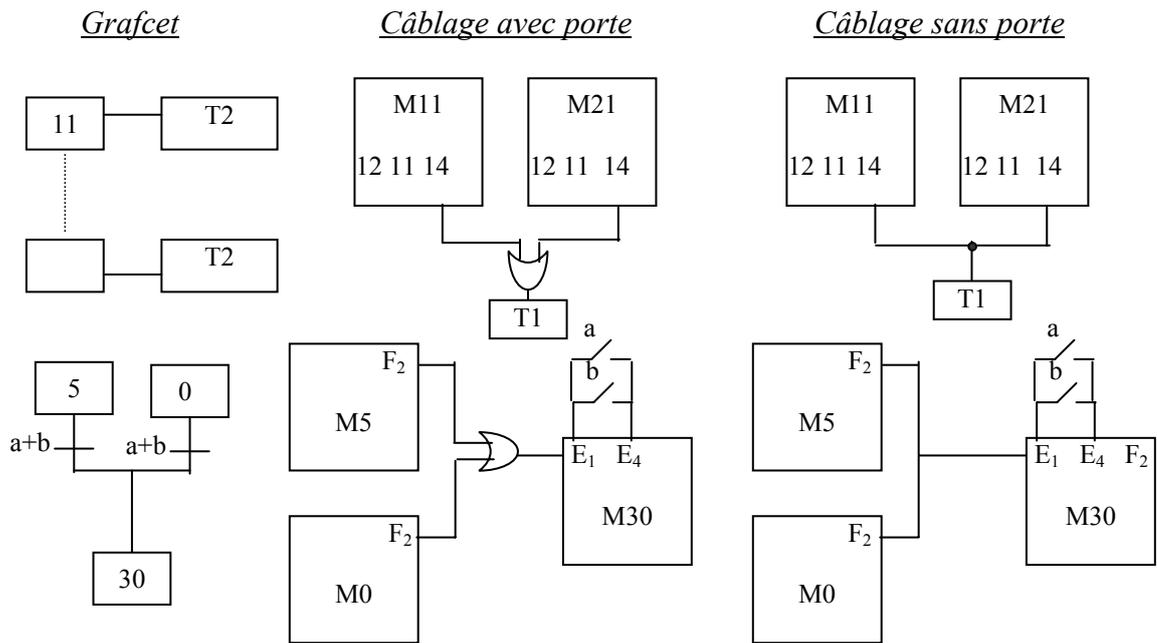
Figure 4.15-b : Alimentation et commande des préactionneurs



-5-Réalisation de fonctions logiques dans le câblage de séquenceurs à partir du grafcet

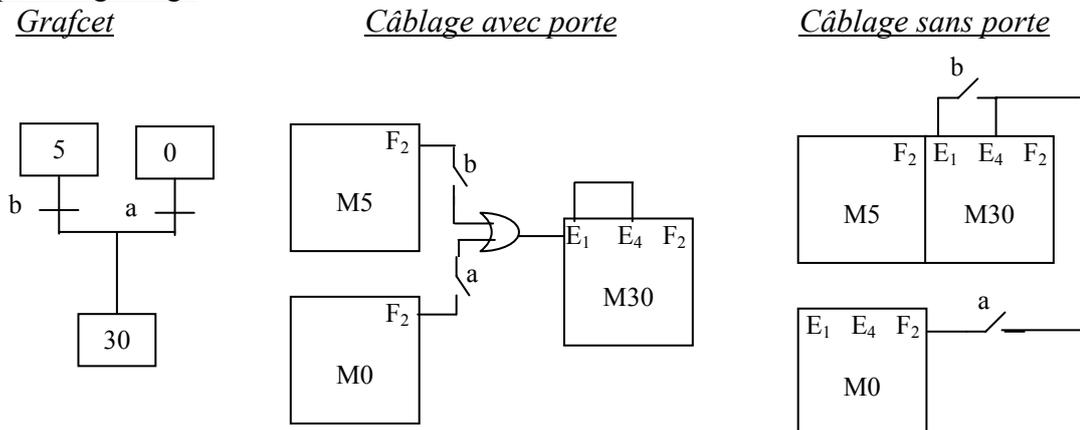
Pour réaliser des fonctions logiques, en technologie pneumatique on peut utiliser des cellules, en technologie électronique on utilise des portes logiques. En technologie électrique, qui est à base de relais, il n’y a ni porte logique ni cellule. Par conséquent dans tous les schémas de câblage des séquenceurs électriques, les portes et cellules logiques doivent être supprimées des schémas, et réalisées en utilisant au mieux les contacts du relais (module d’étape) qui se ferment quand le module est actif.

-5-a-Réalisation du « OU » logique

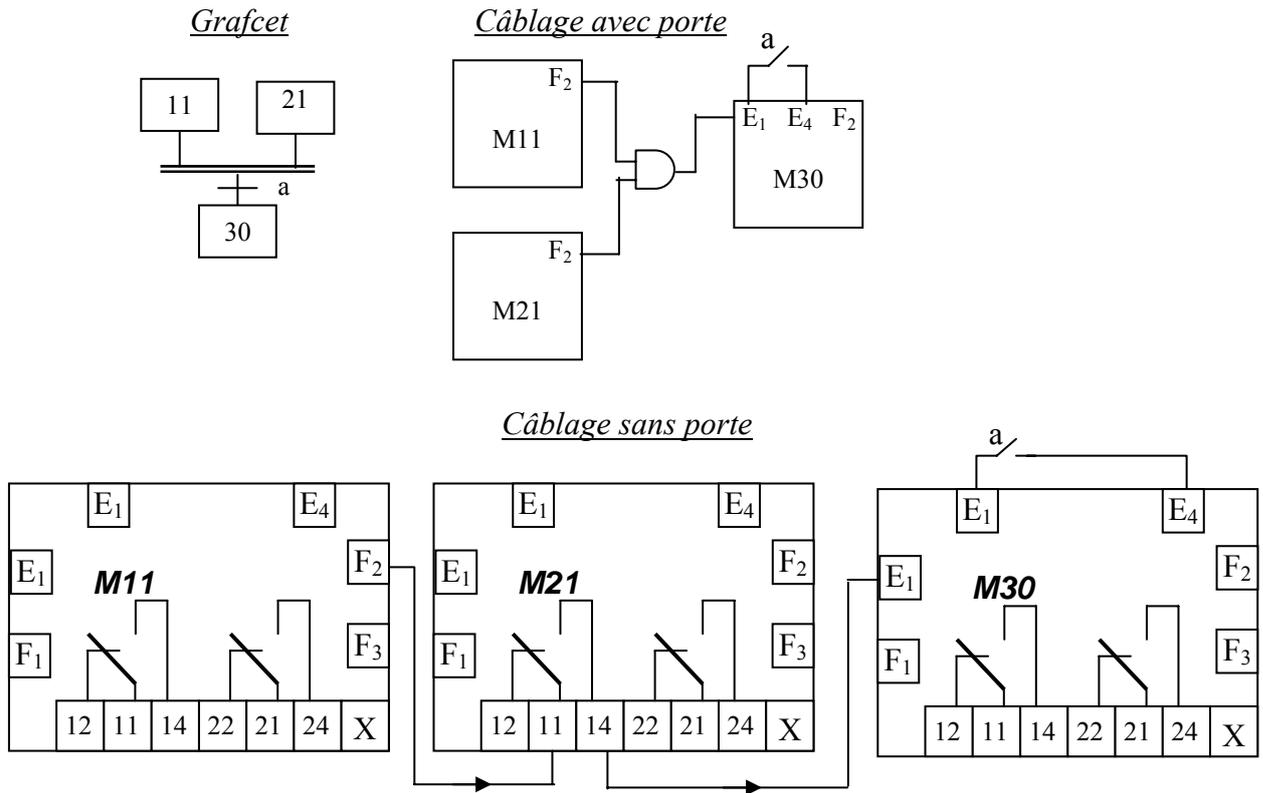


Les bornes de sortie d’un module d’étape sont des contacts de relais. Si on relie 2 sorties entre elles, cela réalise un « OU » logique. Il n’y a aucun problème si les 2 sorties « 14 » ou « F2 » sont au « 1 » logique en même temps, puisque les 2 points seront au même potentiel (en général 24 volts).

Exemple d’aiguillage

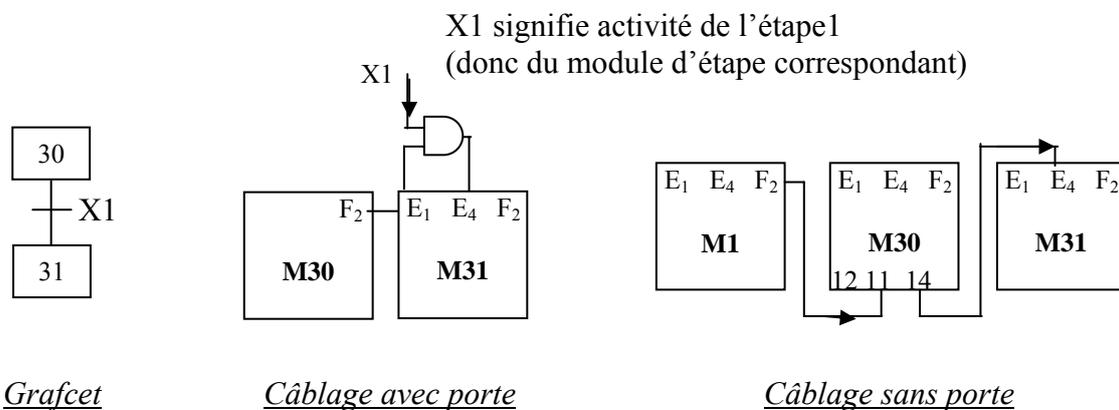


-5-b-Réalisation du « ET » logique

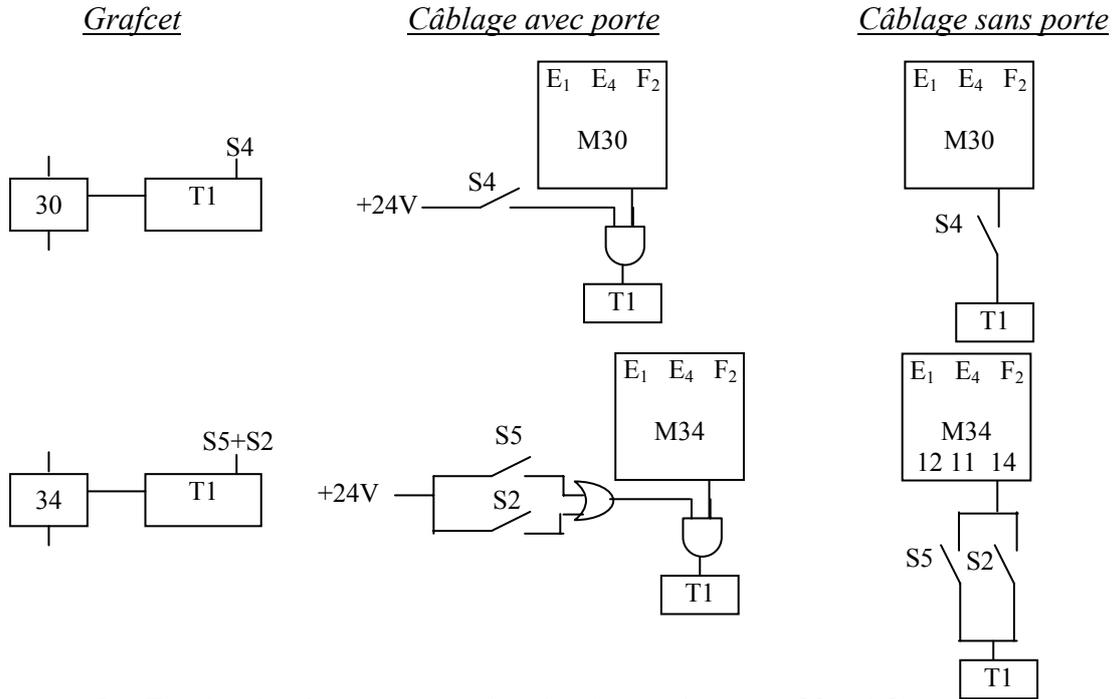


Quand la bobine du module M21 est alimentée (étape 21 active), les bornes 11 et 14 de M21 sont reliées entre elles, et le signal venant de F2 du module M11 se retrouve sur la sortie 14 de M21, ce qui a pour effet d'envoyer le signal de validation pour M30. On en conclut que M30 n'est validé que si M11 et M21 sont actifs. Pour que M30 soit actif, il faut en plus un signal sur sa borne E4, pour cela il faut que l'information « a » soit vraie.

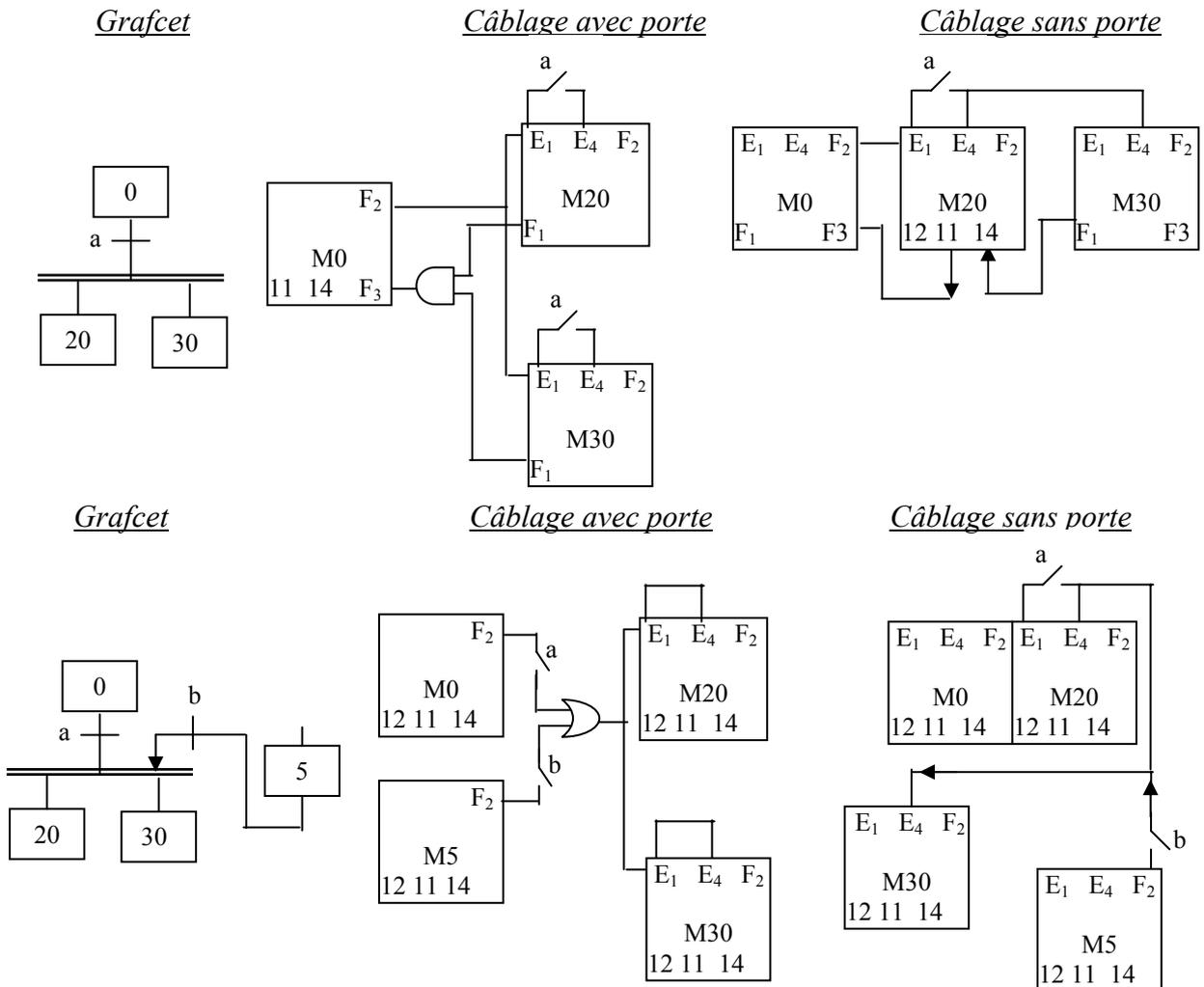
-5-c-Etape comme réceptivité (exemple page 71 vol 2)

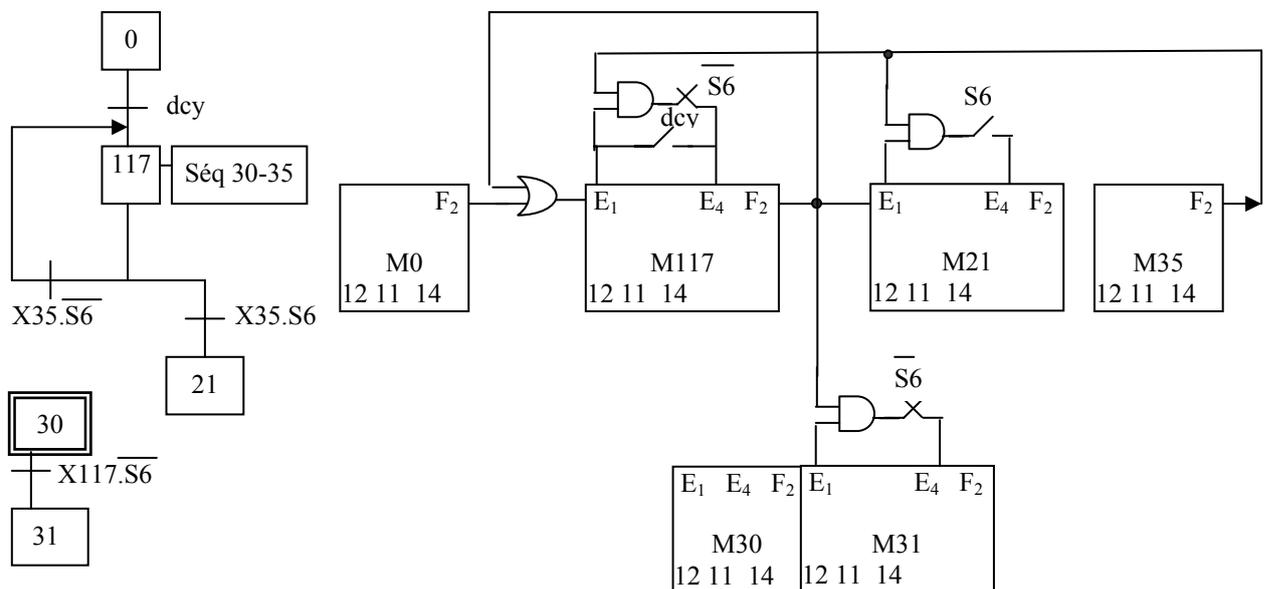


-5-d- actions conditionnelles (exemples page 74 vol 2)



-5-e-Entrées de séquences simultanées (exemples page 80 vol 2)



-5-f-Sous programmes et boucles de retour (exemples page 73 vol 2)GrafcetCâblage avec portesRemarques

En premier lieu comme la règle numéro 5 de la théorie du grafcet impose que “si une étape doit être activée et désactivée en même temps, elle doit rester active”, par conséquent le bouclage sur elle-même de l’étape 117 dans le grafcet n’a pas de raison d’être. Ce bouclage est d’autant plus inutile, que l’on a introduit dans le grafcet esclave la variable de contrôle $\overline{S6}$ qui permet le démarrage et le bouclage à l’intérieur du grafcet esclave.

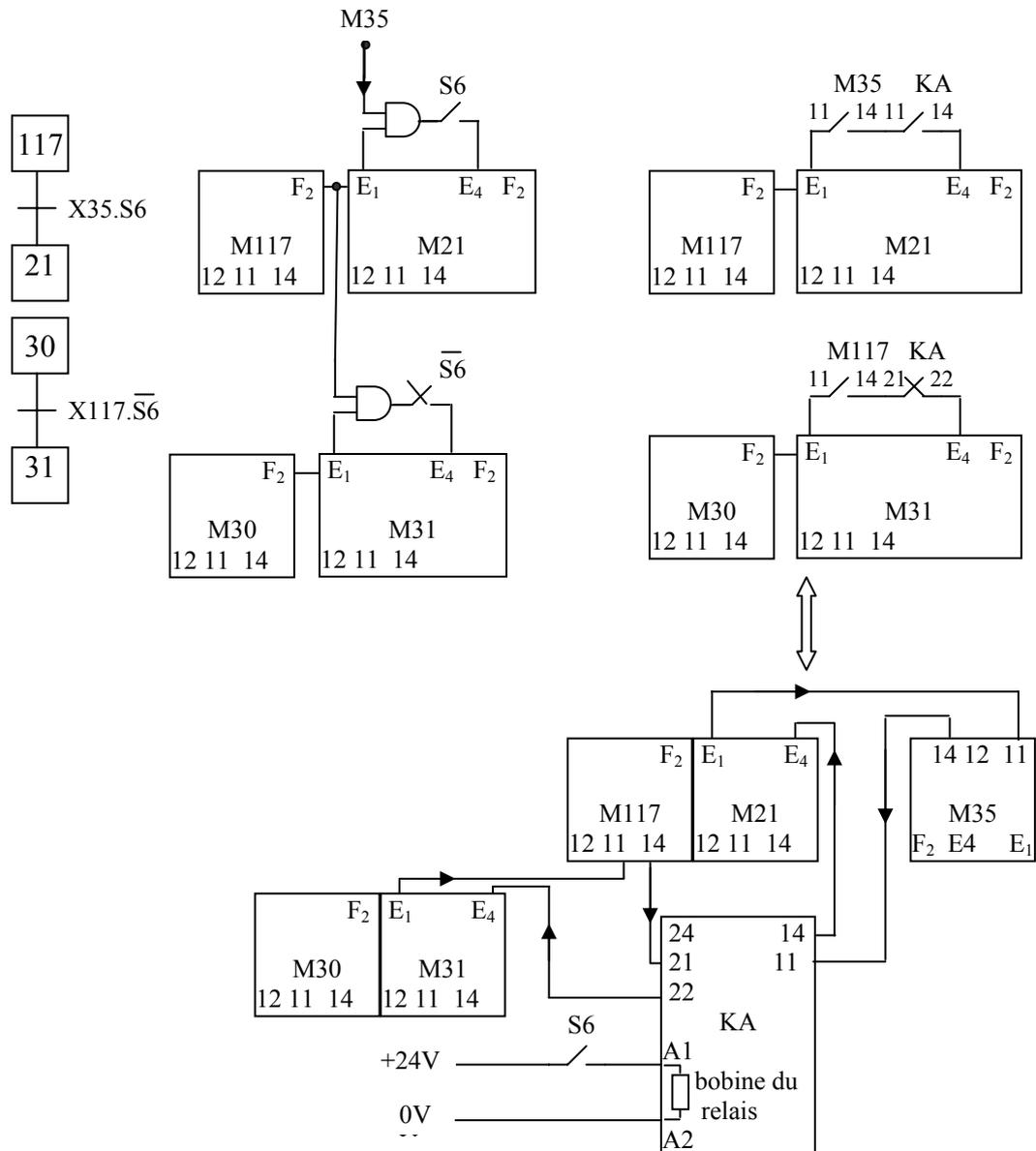
En second lieu si en technologie programmée une même étape peut être bouclée sur elle-même, en technologie câblée à base de modules d’étapes cela est formellement interdit. Par conséquent le grafcet est simplifié en supprimant la boucle de retour de l’étape 117 sur elle-même (ce qui supprime les portes logiques “OU” et “ET” du module M117).

Enfin si un capteur “C” est utilisé plusieurs fois dans le montage (comme c’est le cas ici avec une fois $\overline{S6}$ et deux fois $S6$), cette redondance de variable va être obtenue en utilisant un relais commandé par ce capteur, puis on utilise les contacts travail et repos du relais comme variables “C” et “ \overline{C} ”. Ainsi dans notre cas le capteur symbolisé par le contact $\overline{S6}$ commande la bobine d’un relais KA, puis on se sert des 3 ou 4 groupes de contacts d’E/S comme variables (11,12,14), (21,22,24), (31,32,34), (41,42,44). Dans l’exemple qui suit l’entrée se fait sur la borne 11, et la sortie se fait sur la borne 14 ou sur 12 selon que l’on a besoin de $S6$ ou de $\overline{S6}$.

Grafcet

Câblage avec portes

Câblage sans portes



Remarque :

Le schéma de câblage sans portes du haut offre l'avantage de la concision, mais son inconvénient est qu'il n'indique pas que la bobine du relais KA est alimentée via S6.

-V-LE SEQUENCEUR PNEUMATIQUE

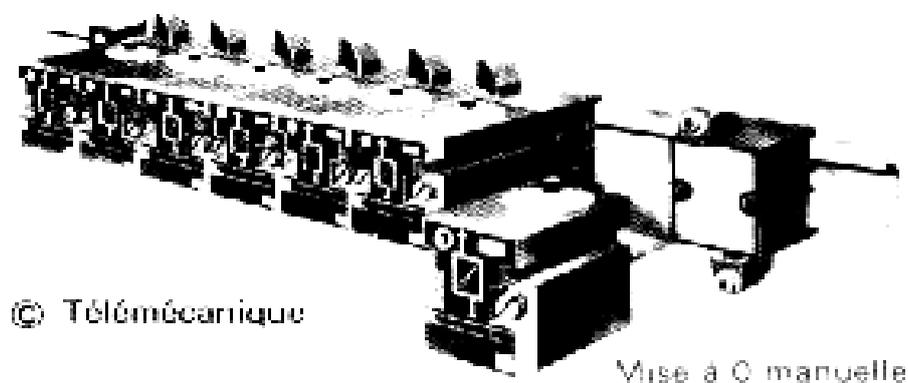


Figure 4.16 : Séquenceur pneumatique câblé

-1-Généralités

La cellule de base est une mémoire pneumatique à remise à zéro prioritaire. On associe à cette mémoire une cellule ET et une cellule OU qui sont soit installées dans l'embase (Télémécanique), soit sur la mémoire elle-même (Crouzet), ou simplement extérieures.

Télémécanique préfère la remise à zéro indépendante de chaque mémoire, ce qui impose qu'il y ait toujours une mémoire à 1, et autorise qu'il y en ait plusieurs à 1 sous réserve qu'elles soient séparées d'au moins une mémoire (inexistence de 2 mémoires consécutives à 1). Ceci permet de traiter les grafjets qui possèdent de très longues séquences où plusieurs étapes sont actives en même temps (pour rester en accord avec la théorie du grafjet où une seule étape par séquence est active à la fois, cette longue séquence est considérée comme plusieurs séquences en cascade et prises en charge chacune par un séquenceur).

Crouzet met en cascade les remises à zéro de toutes les mémoires aval, ce qui interdit d'avoir plus d'une mémoire à 1, permet de démarrer systématiquement avec toutes les mémoires à zéro, et sur la première mémoire du bloc séquenceur.

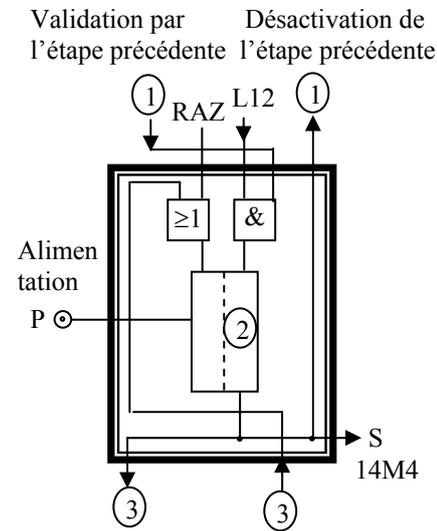
Climax utilise des mémoires à tiroir. Le passage d'une mémoire à l'autre se fait par une remise à zéro de la mémoire amont, avant l'activation de la mémoire aval.

L'avantage de tous ces composants « intégrés » est la rapidité de mise en œuvre et la possibilité de transposition directe du grafjet.

-2- Comparaison des techniques développées

Figures 4.17

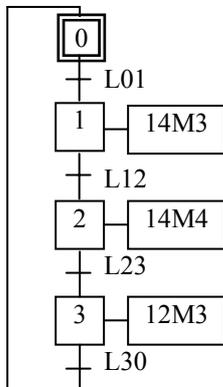
-a-Module théorique



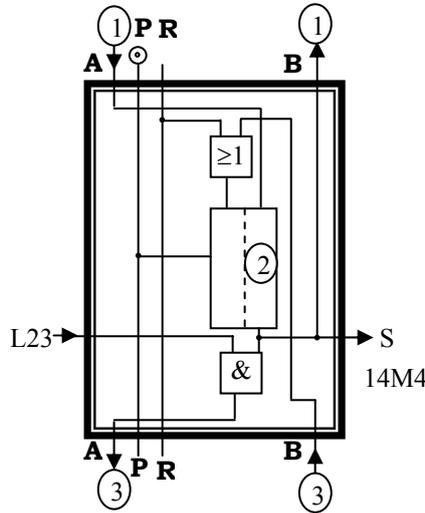
Désactivation par l'étape suivante Validation de l'étape suivante

La cellule ET et la cellule OU concernent l'enclenchement ou le déclenchement de la mémoire d'étape.

-d-Exemple de grafcet à câbler



-b-Module Télémécanique

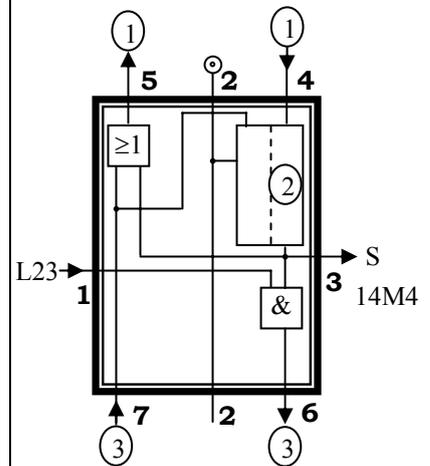


R : remise à zéro
P : alimentation
S : sortie vers les préactionneurs

- La cellule ET concerne la validation du module suivant. Par conséquent la réceptivité à câbler est la réceptivité suivante.
- La cellule OU concerne la désactivation du module courant.

- Il faut câbler la boucle de validation et la boucle de désactivation.
- Il faut valider le dernier module après un arrêt d'urgence.
- Il faut au moins trois modules.

-c-Module Crouzet



- La cellule ET concerne la validation du module suivant. Par conséquent la réceptivité à câbler est la réceptivité suivante.
- La cellule OU concerne le module précédent, et le signal remet à zéro tous les modules précédents. La mémoire est à effacement prioritaire.

- Pas de boucle à câbler.
- Pas de validation d'un module après un arrêt d'urgence.
- Deux modules suffisent pour fonctionner.

-3-Mise en cascade des modules ou câblage de grafcet

Il s'agit de câbler le séquenceur correspondant au grafcet précédent. La disposition des entrées / sorties facilite le câblage en emboîtant les modules les uns dans les autres.

-3-a- Modules Crouzet

L'association d'une mémoire à l'étape initiale est inutile. Dès l'apparition du signal L01, la mémoire du module numéro 1 se met à 1 et le cycle peut commencer.

Comme la cellule ET concerne le module suivant, et qu'il n'y a pas de module associé à l'étape zéro, alors pour la dernière étape du grafcet, on a bouclé la sortie 6 de la cellule ET du module M3 sur l'entrée 7 de la cellule OU du même module. Ainsi lorsque la réceptivité L30 est vraie, le module d'étape M3 et tous les modules précédents sont désactivés et le cycle s'arrête. Dès que L01 réapparaît le cycle peut recommencer.

Figure 4.17-b : Schéma de câblage des modules *Figure 4.17-a : Grafcet à câbler*

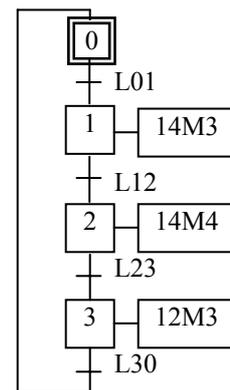
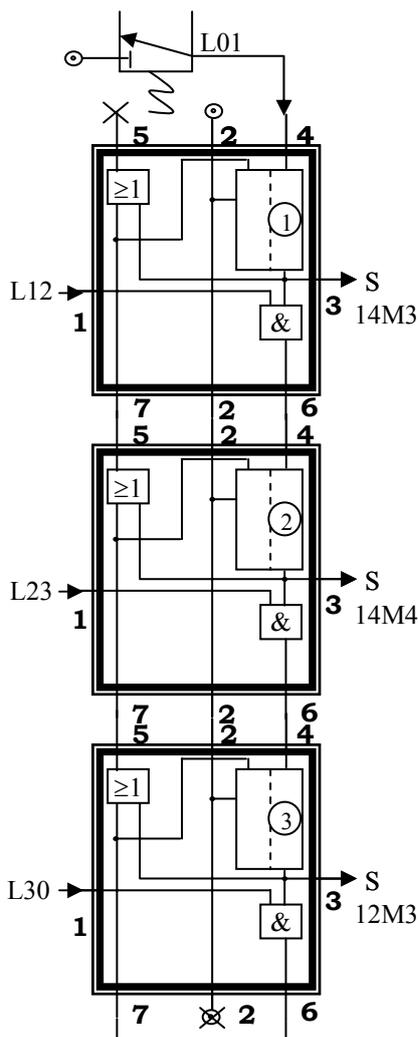
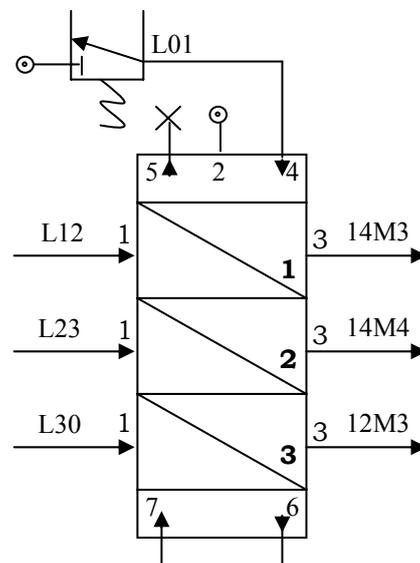


Figure 4.17-c : Représentation schématique simplifiée



-3-b- Modules Télémécanique

Le câblage est facilité par le principe de l'association d'une mémoire à chaque étape. Ainsi contrairement aux modules Crouzet, l'association d'une mémoire M0 à l'étape initiale est obligatoire. On remarque également les liaisons entre les modules M0 et M3 pour câbler la boucle de validation et la boucle de désactivation.

Figure 4.18-b : Schéma de câblage des modules *Figure 4.18-a : Grafset à câbler*

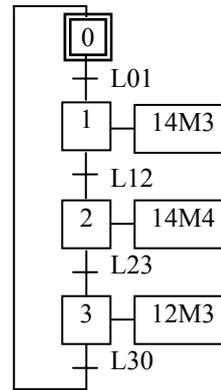
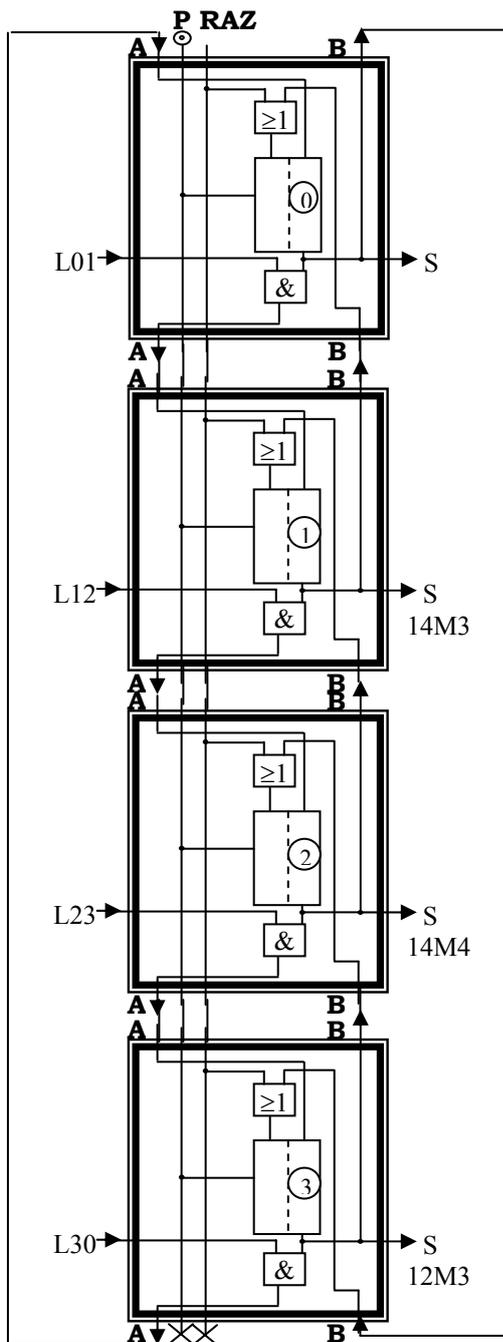
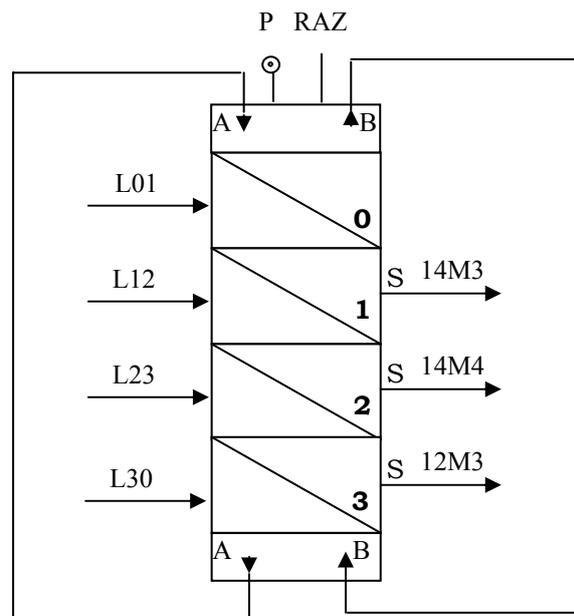


Figure 4.18-c : Représentation schématique simplifiée



-4-Câblage de grafquets à séquences multiples

GRAF CET

SEQUENCEURS

modules d'étapes Cruzet

modules d'étapes Télémécanique

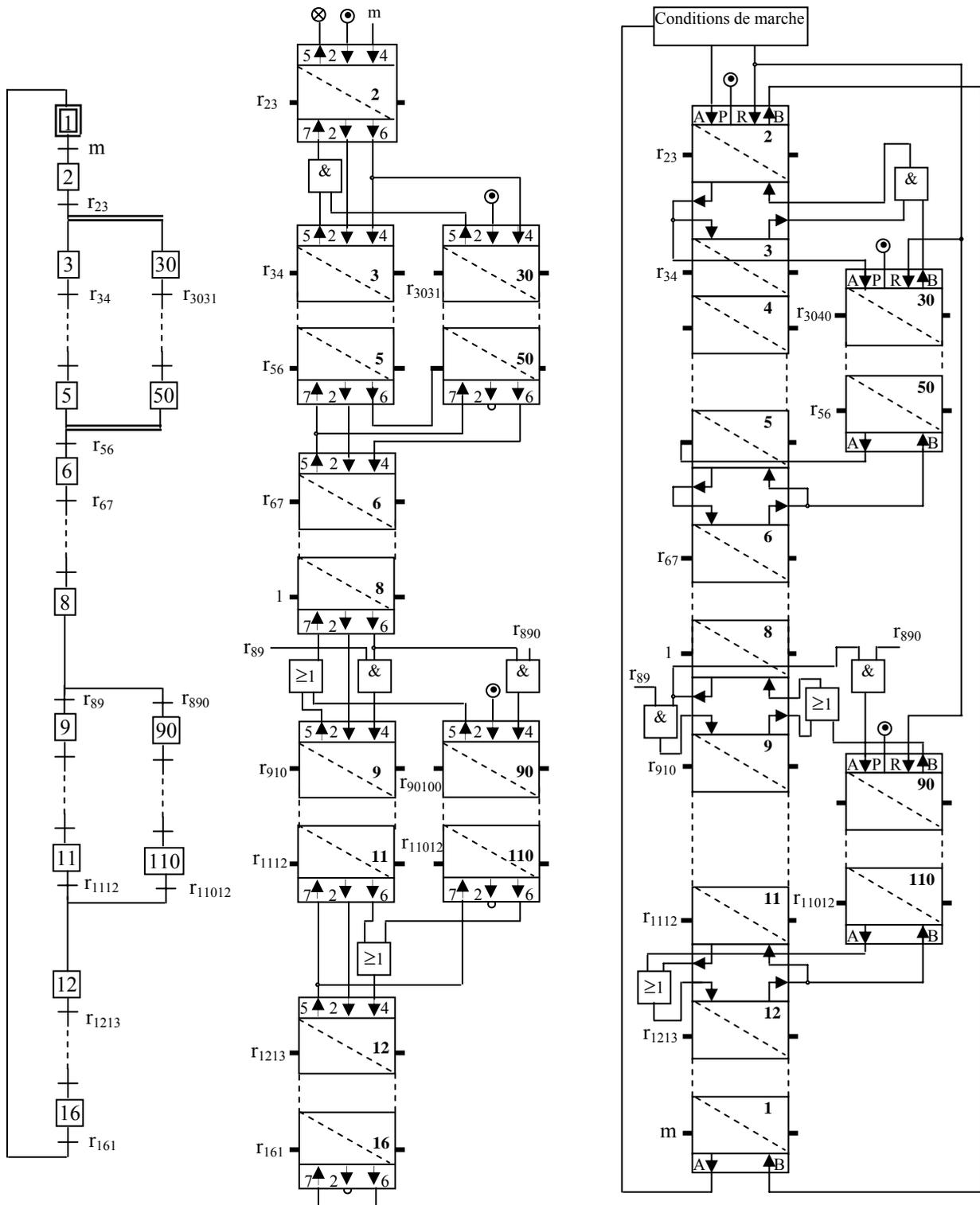


Figure 4.19 : Grafcet à séquences multiples et séquenceurs correspondants

Remarques

- On notera l'existence d'un module d'étape initiale chez Télémécanique et son absence chez Crouzet.
- Pour la séquence simultanée, la divergence en ET est réalisée par l'introduction d'une cellule ET dans le circuit de désactivation du module d'entrée numéro 2. La convergence en ET est réalisée en affectant la réceptivité commune de sortie r_{56} à un des modules, puis on considère sa sortie comme entrée de réceptivité du second module, et enfin ce dernier valide le module correspondant à l'étape de sortie de la séquence simultanée. On aurait pu obtenir le même résultat en introduisant une cellule ET pour valider le module de sortie, laquelle est pilotée par les sorties des deux modules.
- Pour l'aiguillage, la divergence en OU est réalisée par l'introduction de deux cellules ET pour la validation des modules, et une cellule OU pour la désactivation du module d'entrée. Il est utile de noter que la réceptivité pour ce module d'entrée est mise à un, et donc obligatoirement reliée à l'arrivée de pression. La convergence en OU est réalisée par l'introduction d'une cellule OU dans le circuit de validation du module de sortie.
- Les étapes d'entrée/sortie des séquences multiples sont symétriques et jouent le même rôle. Pour illustrer cela, sur les schémas de câblage de la séquence simultanée, la réceptivité r_{56} est volontairement disposée sur le module 5 dans le câblage des modules crouzet, alors qu'elle est disposée sur le module 50 pour Télémécanique.
- Par conséquent le câblage des modules d'entrée/sortie des aiguillages et séquences simultanées peut être modifié, notamment suivant la structure des réceptivités r_n .
- On notera chez télémécanique l'utilisation de modules de dérivation. Par ailleurs les conditions de marche nous donnent un ordre relié à l'entrée de remise à zéro de tous les modules. Enfin pour rester conforme au grafset, et si les conditions de sécurité le permettent, on peut directement valider le module 2 par le module 1 sous le contrôle de la réceptivité « m » correspondant au bouton de mise en marche.

CHAPITRE 5 : COMMANDE EN LOGIQUE PROGRAMMEE : L'API
API : AUTOMATE PROGRAMMABLE INDUSTRIEL

	PAGE
-I-INTRODUCTION	
-1-Généralités -----	111
-2-Liaisons de l'API avec son environnement	
-II-ARCHITECTURE	
-1-Généralités -----	112
-2-Le processeur	
-3-La zone mémoire	
-4-Les interfaces -----	114
-5-Alimentations -----	117
-III-PROGRAMMATION	
-1-Généralités -----	118
-2-Structure des instructions	
-3-Exécution d'un programme -----	119
-4-Langage à contacts (TSX 27 & TSX 17-20) -----	121
-5-Langage booléen (tsx21) -----	125
-6-Programmation des fonctions séquentielles -----	127
-a- Fonction mémoire -b- Fonction temporisation -c- Fonction comptage	
-7-Langage Grafcet : programmation de séquence unique -----	132
-a- Généralités et exemple	
-b- Programmation sur TSX-21 (Télémechanique)	
-c- Programmation sur CMP 31-34 (Crouzet) -----	135
-d- Programmation sur PB 100 (Merlin Gerin) -----	136
-e- Programmation en PL7-1 sur TSX17-10 -----	139
-f- Programmation en PL7-2 sur TSX 17-20 -----	142
-8-Introduction des temporisations -----	144
-a- Exemple	
-b- Programmation sur TSX-21	
-c- Programmation sur CMP 31-34	
-d- Programmation sur TSX 17-10	
-e- Programmation sur TSX 17-20	
-IV-PROGRAMMATION DE GRAFCETS A SEQUENCES MULTIPLES	
-1-Aiguillage sur TSX21 -----	147
-2-Séquences simultanées sur TSX21 -----	148
-3-Aiguillage sur TSX17-10 -----	152
-4-Aiguillage sur TSX17-20 -----	153
-5-Séquences simultanées sur TSX17-20 -----	155
-V-CABLAGE DES API TSX17	
-1- Cas général -----	156
-2- Fonctionnement en mode protégé -----	157

REFERENCES BIBLIOGRAPHIQUES

- 1- M.Pinot & al « Du grafcet aux automates programmables », collection L.P., édit Foucher, Paris 1986.
- 2- A.Bianciotto & P.Boye « L'informatique en automatisation industrielle », tome 1, collection Techniques et Normalisation, édit Delagrave.
- 3- Documents techniques des automates TSX 17,Télémechanique.
- 4- P.Denis & D.Murail & al « Les activités de la maintenance », MSMA, collection espaces technologiques, édit Delagrave 1994.

-I-INTRODUCTION

-1-GENERALITES

Un API est un dispositif électronique capable de piloter (assurer la commande) un processus industriel, dans des environnements très diversifiés : mécanique, fonderie, agro-alimentaire, électronique etc...

Il est programmable, donc possède une mémoire pour enregistrer des programmes, qui seront écrits dans un langage propre à chaque API. Sa programmation ne nécessite pas d'être informaticien ou la connaissance d'un langage « informatique ».

L'API représente la partie commande du schéma classique d'un processus automatisé (cf. Figure 5.1).

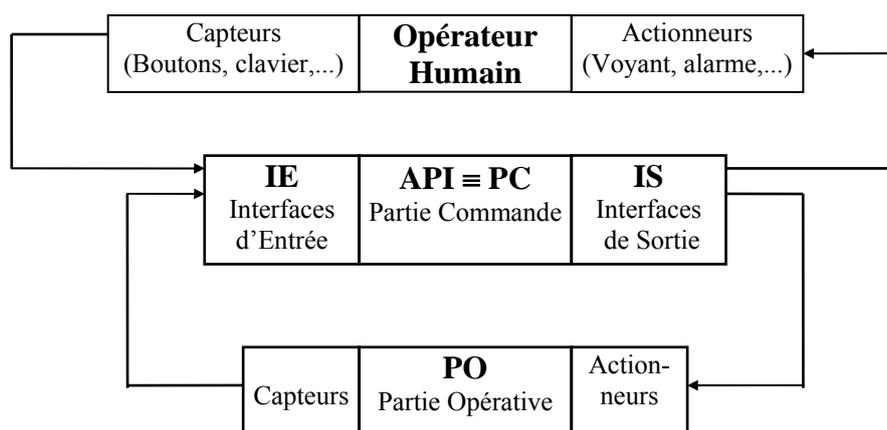


Figure 5.1 : schéma simplifié de la structure d'un automatisme

-2-LIAISONS DE L'API AVEC SON ENVIRONNEMENT

Chaque capteur de la partie opérative est relié à une borne de l'interface d'entrée de l'automate, et chaque borne de sortie de l'automate est reliée à une interface de puissance d'un actionneur de la partie opérative (relais, électrodistributeur, voyant etc...).

L'opérateur humain communique avec l'automate par le biais d'une console de programmation.

Enfin un ensemble de périphériques nécessaires aussi bien à l'interface homme-machine (écran, imprimante, programmeur d'Eeprom) qu'à la bonne conduite du processus automatisé (mémoire de masse, Eeprom) sont reliés à l'API.

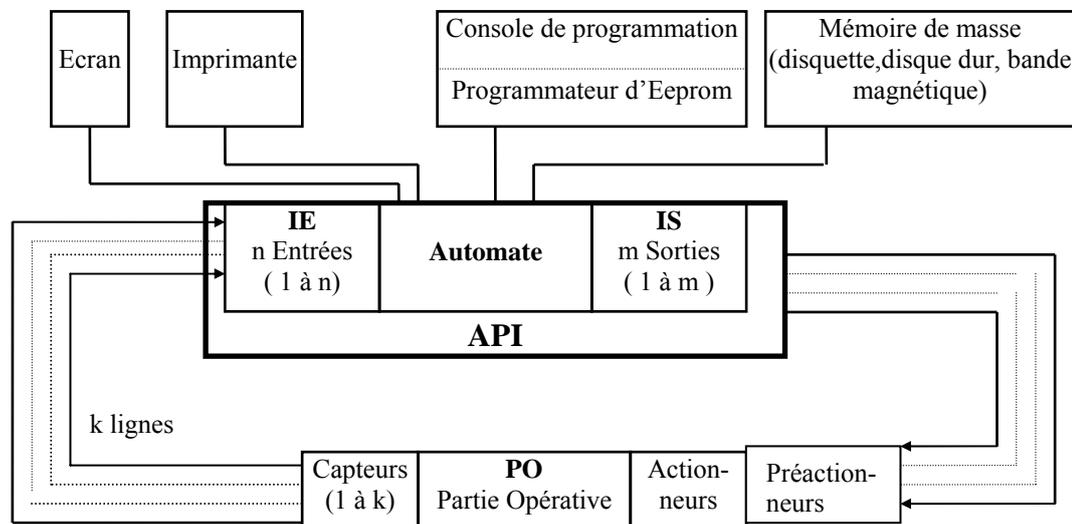


Figure 5.2 : Liaisons de l'API avec son environnement

-II-ARCHITECTURE GENERALE

-1-GENERALITES

Un API se compose de trois parties principales : le processeur, la zone mémoire et les interfaces. Elles sont reliées entre elles par l'intermédiaire d'un ensemble de fils, qui permettent d'acheminer les informations, qu'on appelle des bus internes (bus 8 bits par exemple dans le cas de regroupements des fils par 8).

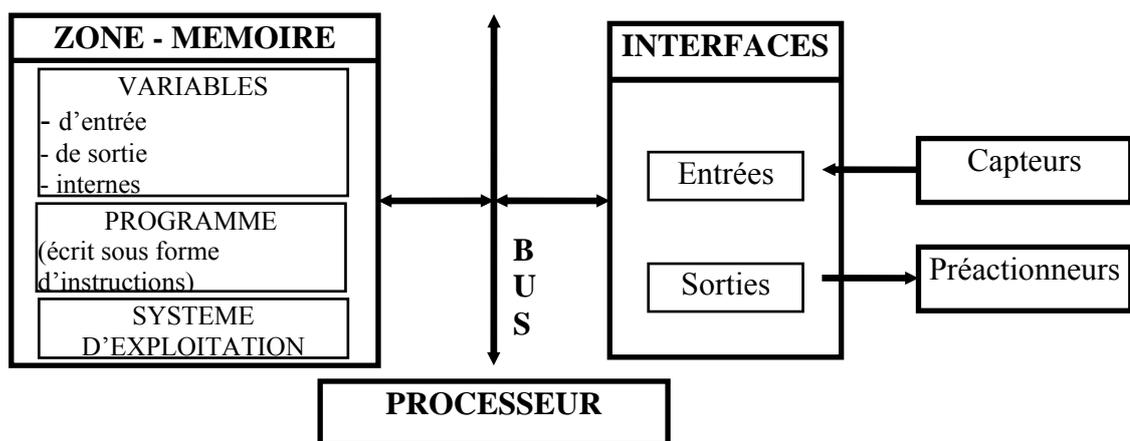


Figure 5.3 : Schéma simplifié de l'architecture d'un API

-2- LE PROCESSEUR

C'est le cerveau de l'automate : en plus de son rôle de calculateur, il gère et coordonne les relations entre les zones mémoires et les interfaces. Il procède à :

- la lecture des informations d'entrée,
- l'interprétation de ces informations et l'exécution des instructions du programme contenu en mémoire,
- la commande des sorties.

Au point de vue technologique le processeur peut être :

- soit câblé c'est à dire réalisé à l'aide de composants logiques tels que portes, bascules, registres etc... C'est le cas du TSX 21 de Télémécanique. Vu le développement actuel de la technologie des microprocesseurs et leur prix, cette solution est pratiquement abandonnée ;
- soit microprogrammé, c'est à dire qu'on utilise un microprocesseur où les fonctions sont exécutées par des programmes élémentaires internes au microprocesseur. C'est le cas du PB 100 de Merlin Gerin (μ P 6809), TSX 17 et TSX 80 de Télémécanique, PLC-2 d'Allen Bradley (μ P Z80).

-3- LA ZONE MEMOIRE

La zone mémoire se caractérise par sa capacité utile. Plutôt que de l'exprimer en kilooctets, on préfère l'exprimer en kilomots, la taille des mots variant généralement de 1 à 64 bits.

-3-a-Technologie des mémoires

Dans un API on rencontre différents types de mémoires, ayant des rôles différents en fonction de leur technologie.

-RAM (Random Access Memory) : c'est une mémoire vive dite volatile car son contenu « s'efface » dès qu'elle n'est plus alimentée. C'est la zone de travail du processeur (et de l'utilisateur éventuellement).

-ROM (Read Only Memory) : appelée mémoire morte, seule la lecture y est possible. On y stocke généralement les programmes de gestion de l'automate appelés « système d'exploitation ». Son contenu ne s'efface pas en cas de coupure de l'alimentation.

-EPROM ou REPROM (Eraseable Programmable ROM) : Elle peut être reprogrammée après avoir été exposée à un rayonnement ultraviolet. Elle est utilisée pour sauvegarder les programmes pendant la phase d'exploitation. Ce type de mémoire est de moins en moins utilisé.

-EEPROM (Electrically EPROM) : C'est une EPROM dont le contenu peut être effacé électriquement. Elle est généralement utilisée pendant la phase de développement des programmes. A cause de sa souplesse d'utilisation par rapport à l'EPROM, elle tend de plus en plus à la remplacer pour stocker les programmes de l'utilisateur pendant la phase d'exploitation. Elle se présente sous forme de cartouche à enficher directement sur le support qui lui est réservé dans l'automate.

-3-b-Fonctions de la zone mémoire

Fonctionnellement la mémoire est organisée pour :

- recevoir les informations issues des capteurs d'entrée et les mémoriser,
- recevoir les informations générées par le processeur, qu'elles soient de type variables internes (résultats de calculs intermédiaires temporaires logiques ou numériques), ou de type variables de sortie destinées à la commande des préactionneurs,
- conserver les programmes à exécuter écrits dans le langage propre à l'automate,
- conserver le système d'exploitation et les données associées.

-4- LES INTERFACES

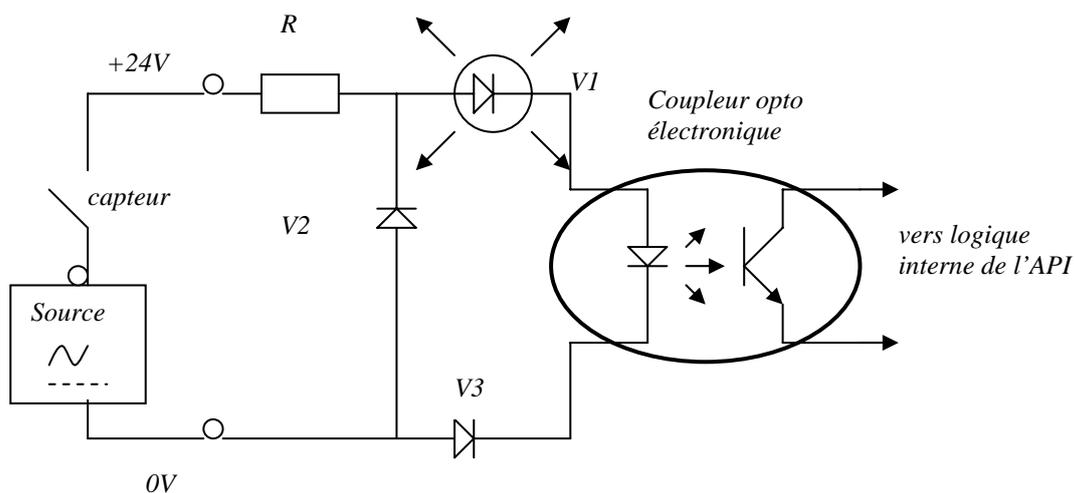
On désigne généralement les API par le nombre d'entrées sorties dont ils disposent (32 E/S : 24 entrées et 8 sorties par exemple), donc le nombre de signaux d'E/S qu'ils peuvent traiter simultanément. Bien que ces signaux soient de nature analogique ou logique, nous ne nous intéressons qu'au cas le plus fréquent qui est celui des signaux logiques (de type « tout ou rien »), certains API pouvant traiter des signaux analogiques en nombre limité.

L'API dispose d'un module d'interface pour chaque entrée (capteur) et chaque sortie (préactionneur), dont l'état est indiqué par une LED. A chaque module est associée une adresse. L'ensemble des modules d'entrée constitue l'interface (ou les interfaces) d'entrée, et ceux de sortie l'interface de sortie.

-4-a-Interfaces d'entrée

Ils assurent principalement trois fonctions : la mise en forme du signal d'entrée pour produire un signal logique bas niveau (écrêtage par diode Zéner ou seuillage par trigger de Schmidt), l'élimination des parasites industriels et l'effet de rebondissement des contacts par filtrage passe bas retardateur, et le découplage ou isolation galvanique en isolant électriquement les signaux logiques de la partie commande des signaux analogiques de la partie opérative.

Associée à un circuit retardateur écréteur (diode Zéner), une résistance de limitation de courant et une diode de protection contre les inversions de courant (cf Figure 5-a), l'interface d'entrée la plus utilisée est le découpleur optoélectronique (cf chapitre 3) car il réalise une bonne isolation galvanique. Il permet une protection contre : les perturbations électromagnétiques, les défaillances des capteurs (effet de rebondissement des lames), et les défauts de câblage (courants de fuite et mauvaise mise à la terre).



- Légende :**
- R : adaptation au seuil de tension d'entrée*
 - V1 : signalisation de l'état de l'entrée*
 - V2 : protection en cas d'inversion de tension aux entrées*
 - V3 : retardateur, diode Zéner en écréteur*

Figure 5.4-a : Schéma de principe d'interface d'entrée logique en 24V continu

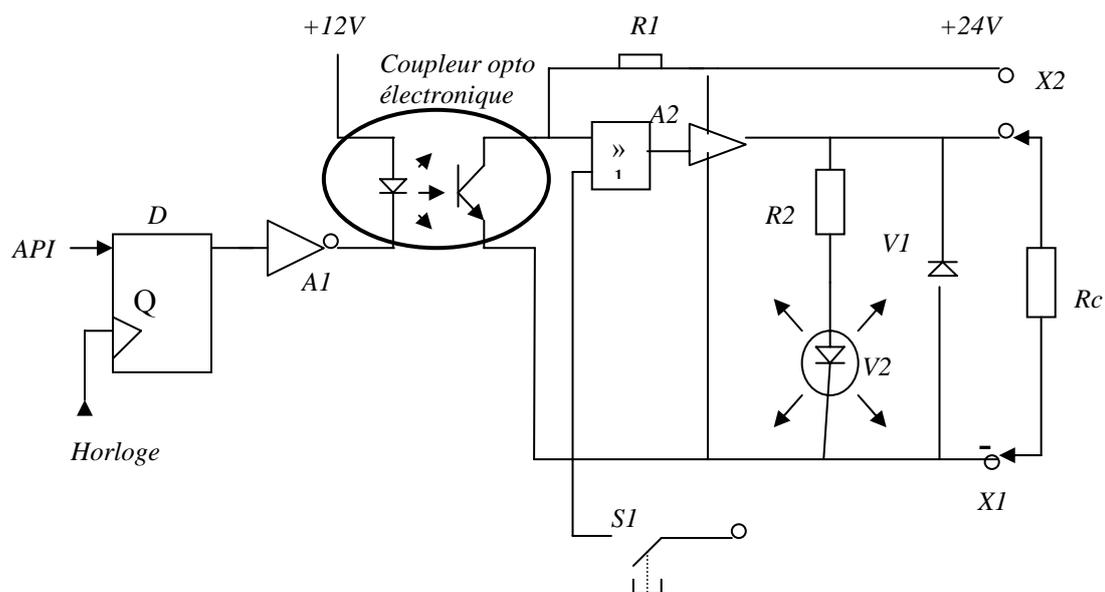
-4-b-Interfaces de sortie

En plus des fonctions d'isolation (par coupleur optoélectronique ou relais Reed) et de protection contre les courts-circuits et les surcharges (par un dispositif thermique), contre l'inversion de polarité de l'alimentation extérieure (par diode), ils assurent une fonction de préamplification car ils sont destinés à convertir les signaux bas niveau de la partie commande en signaux ayant une puissance suffisante pour piloter les préactionneurs (contacteurs, électrovannes, distributeurs...).

La commutation et l'amplification de puissance adaptées à la charge sont réalisées :

- *en alternatif* : le plus souvent par un relais statique ou électromagnétique (dans ce dernier cas le coupleur optoélectronique n'est pas indispensable). On peut également rencontrer un amplificateur intégré ou un triac.
- *en continu* : on utilise souvent un transistor de puissance ou un amplificateur intégré.

Le choix de la solution de commutation dépend en fait de la nature de la charge : puissance, nature des courants, inductif ou résistif, etc...



- Légende :**
- D* : mémorisation du résultat du traitement
 - A1, A2* : amplification
 - Rc* : charge
 - X1, X2* : bornes de la source d'alimentation extérieure
 - S1* : interrupteur de test de la sortie et de la LED

Figure 5.4-b : Schéma de principe d'une interface de sortie en 24V continu

-5- ALIMENTATIONS

Il existe à l'intérieur de l'automate un bloc d'alimentation fournissant les différentes tensions nécessaires aussi bien à la logique de commande interne (5 V à ± 15 V) qu'aux entrées (24 V à 220 V). La tension normalisée la plus courante est de 24 V (alternatif ou continu).

Généralement l'alimentation des capteurs est fournie par l'API (en 24 V) alors que celle des préactionneurs est fournie par une alimentation externe.

En courant continu, les tensions des signaux de sortie de l'API sont de l'ordre de 24, 48 ou 110V, alors que l'intensité du courant varie entre 0.5 et 2 A. En alternatif 50 HZ, elles peuvent aller jusqu'à 220 V et 5 A respectivement.

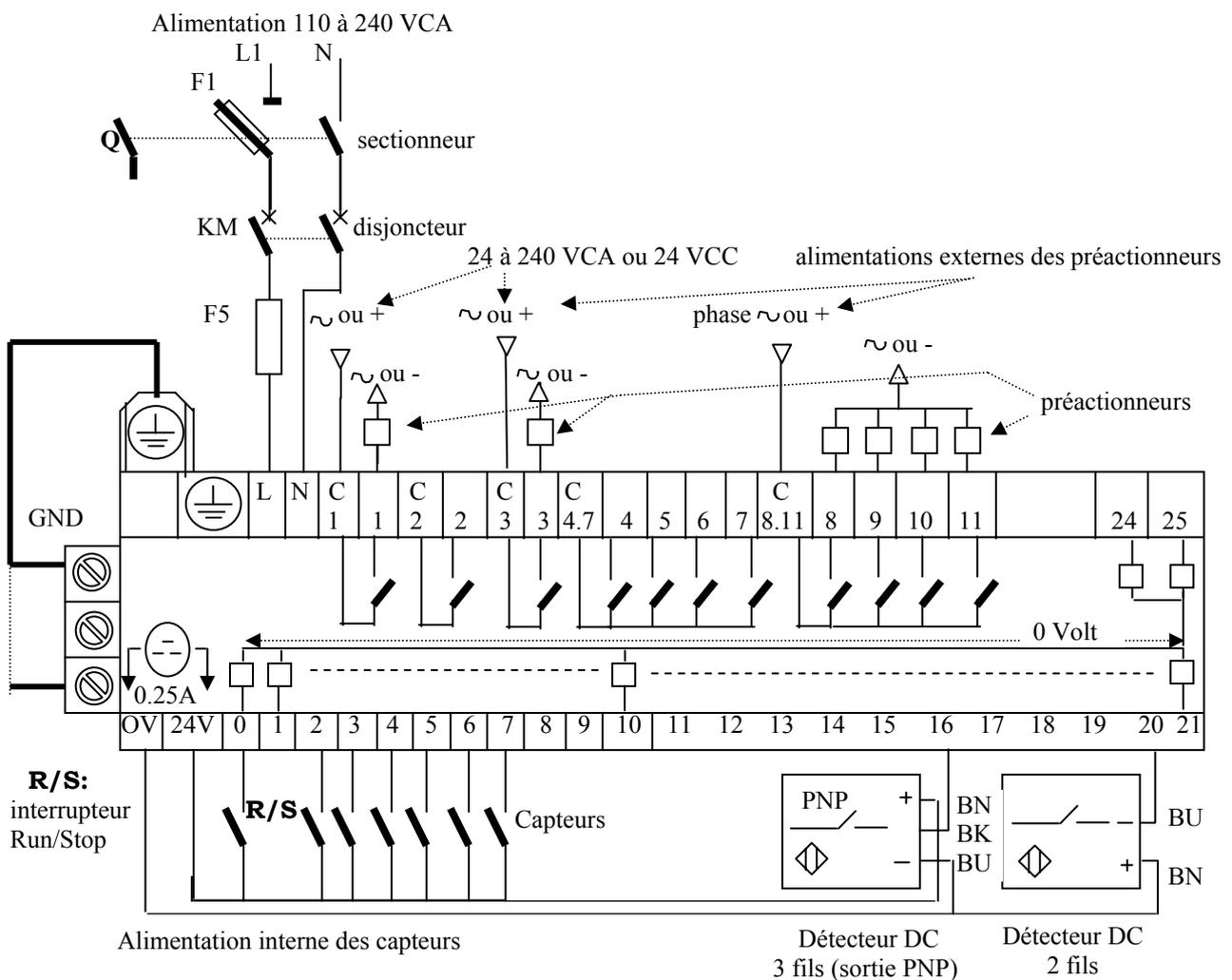


Figure 5.5 : Exemple de câblage de l'automate programmable TSX17

-III-PROGRAMMATION DES API

-1-GENERALITES

Un programme est une suite d'instructions écrites dans le langage de programmation particulier de l'automate. Les automates modernes acceptent plusieurs langages de programmation dont certains sont adaptés aux problèmes d'automatisation.

Ils permettent tous de traiter les opérations de base que l'on rencontre en logique câblée : les opérations arithmétiques élémentaires, les opérations logiques (ET, OU, NON, ...), les opérations séquentielles (comptage, décomptage, temporisation, mémoire bistable, saut,...).

Les langages que l'on rencontre sur les API peuvent être regroupés en 3 familles : le langage à contacts, les langages booléens, le langage grafcet.

-2-STRUCTURE DES INSTRUCTIONS

Une instruction décrit une opération à effectuer. Elle comprend deux parties :

- la première correspond à la nature de l'opération à effectuer (faire un OU, complémenter, initialiser un compteur,...), elle répond donc à la question « *Quoi faire ?* »,
- la seconde correspond à l'adresse de l'opérande, c'est à dire l'adresse dont le contenu fait l'objet de l'opération, elle répond donc à la question « *Sur quoi ?* ».

La nature des opérations et les adresses d'opérandes sont codifiées par les constructeurs d'API. Mais contrairement aux langages informatiques classiques où l'on saisit un programme comme si l'on « tapait » un texte, dans les API chaque instruction est matérialisée par une touche du clavier ou de la console de programmation, elle-même repérée par une abréviation de l'instruction (mnémonique) ou un symbole graphique. Certaines touches possèdent une double fonction.

Quel que soit le langage utilisé, la nature même de l'API et sa place dans la structure du système automatisé font que, vis à vis du reste du système automatisé, le rôle du programme revient à traiter des variables de type booléen repérées par des adresses : saisie d'informations d'entrées issues des capteurs et envoi d'ordres aux actionneurs par le biais des sorties.

-3-EXECUTION DU PROGRAMME

-3-a-Principe

Indépendamment de la famille du langage utilisé (donc de sa nature), l'exécution d'un programme a lieu de manière cyclique : l'ensemble du programme est évalué à chaque cycle de l'automate, d'où l'appellation d'*automate à déroulement cyclique de mémoire ou de phase*. Durant un cycle on effectue systématiquement la scrutation de toutes les entrées et la mise à jour des sorties. Cela est rendu possible par la lenteur de l'évolution du processus (changement des informations d'entrée) par rapport aux temps de réponse de l'électronique de l'automate.

On peut distinguer 4 phases ordonnées dans un cycle d'automate:

- *Gestion système* : elle englobe les opérations principales de surveillance automate (gestion du temps, bits de défaut, bits système, mise à jour des voyants,...) et de traitement des requêtes du terminal de programmation.

Cette phase de gestion système a une durée d'exécution qui est relativement négligeable par rapport aux trois autres phases.

- *Acquisition des entrées* : cette opération consiste à lire (dans l'ordre) la valeur des variables d'entrée reflétant l'état logique des capteurs, et à affecter ces valeurs aux bits correspondants (un bit par adresse d'entrée).
- *Traitement du programme* : il s'agit « d'évaluer » (donc d'exécuter) séquentiellement toutes les instructions du programme écrit par l'utilisateur. En fonction des résultats on attribue une valeur à chaque variable de sortie. A ce niveau il n'y a pas encore de commande.

Sur certains automates un « watchdog » (chien de garde) surveille l'exécution du programme. Si le temps d'exécution dépasse une durée prédéterminée (150 ms pour le TSX17-20), il y a déclenchement de l'évènement « défaut d'exécution ».

- *Mise à jour des sorties* : on met à jour les bits correspondant aux variables de sortie (un bit par adresse de sortie), ce qui revient à envoyer les commandes nécessaires sur les sorties.

-3-b-Temps de réponse automate

La Figure 5.6, décrivant la succession des cycles automate, illustre les 3 phases d'un cycle de durée TC : saisie des entrées, traitement du programme, et mise à jour des sorties (le temps de gestion système étant négligeable relativement aux autres).

Quand un évènement survient à l'entrée de l'automate, le temps nécessaire pour que cet évènement soit pris en charge est appelé temps de réponse. Il est égal à :

$$TR = TE + (1 \text{ à } 2) \cdot TC + TS$$

où : TE / TS : temps de propagation de l'information à travers l'interface d'entrée/sortie $\approx 10\text{ms}$.

TR : temps de réponse de l'automate.

TC : temps de cycle \approx entrées + traitement + sorties.

En effet si une information (événement) arrive sur une entrée après le la scrutation de l'adresse qui lui correspond dans la phase de traitement des entrées, cette information doit attendre le prochain cycle de l'automate pour être traitée et influencer éventuellement sur la mise à jour des sorties.

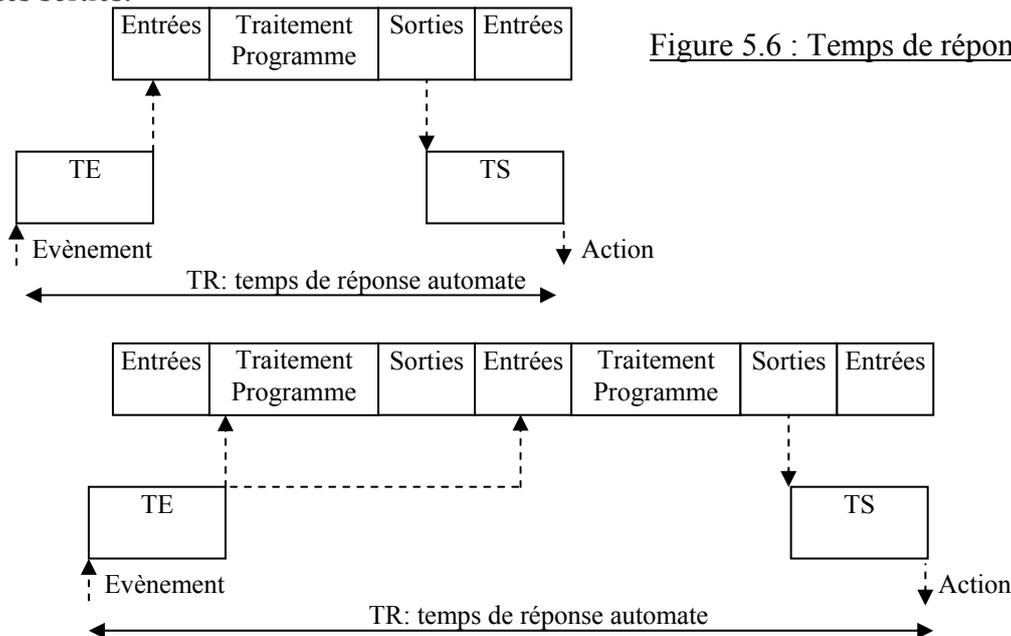


Figure 5.6 : Temps de réponse automate

Remarque

Compte tenu du temps de réponse relativement élevé de l'automate, l'API ne peut pas piloter convenablement un procédé qui nécessite des temps de réponse de l'ordre de la milliseconde. C'est pourquoi sur certains automates comme le TSX 17, on peut rencontrer deux types de tâches : une tâche principale très lente appelée tâche « maître » (dont le cycle automate a été décrit précédemment), et une tâche « esclave » rapide (temps de cycle inférieur à 2 ms). Cette dernière est destinée à des traitements de courte durée et à fréquences élevées.

Pour traiter les événements de la tâche rapide, on utilise sur le TSX17 deux entrées spécifiques d'adresses I0,24 et I0,25 appelées entrées événementielles, qui ont un TE inférieur à une milliseconde (ces entrées peuvent également être configurées et utilisées comme des entrées standard). Quel que soit leur mode de configuration, le raccordement de ces entrées nécessite l'utilisation d'un câble blindé (en raison de leur faible TE).

-4-PROGRAMMATION EN LANGAGE A CONTACTS

-4-a- Exemple du TSX 27 (Télemécanique)

INSTRUCTIONS		TOUCHES
Contact ouvert au repos		
Contact fermé au repos		
Sortie (bobine de relais)		
Liaisons horizontales		
Liaisons verticales	Montante	
	Descendante	

Tableau 1 : Extrait des mnémoniques du langage du TSX 27

Exercice d'application

Dans un processus automatisé un moteur M1 est mis en marche par action simultanée sur 2 capteurs a et b. Si le moteur M1 fonctionne, la mise en marche du moteur M2 peut être autorisée par action sur un capteur c. L'arrêt des deux moteurs est provoqué par un capteur d. Appelons KM1 et KM2 les bobines des contacteurs (préactionneurs) des deux moteurs.

Donner le programme en langage à contacts du schéma de commande.

Solution

VARIABLES E/S	ADRESSES
Entrée :	
d	01
a	02
b	03
c	04
Sortie :	
KM1	100
KM2	101

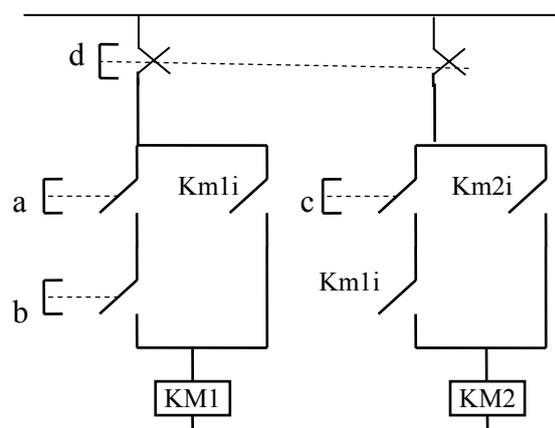
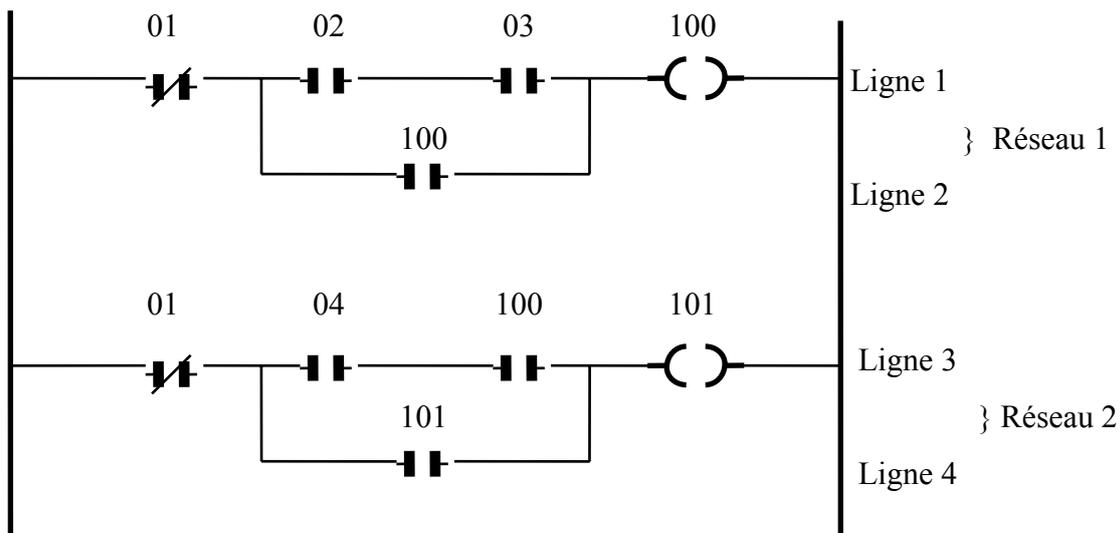


Tableau des variables d'E/S

Figure 5.7-a : Schéma électrique à contacts

Figure 5.7-b : Programme pour automate TSX 27

-4-b-Exemple du TSX17-20 (Télemécanique)

Le langage PL7-2 de programmation de l'automate TSX 17-20 est un langage de type graphique permettant de programmer en langage à contacts et en langage grafcet.

-b-1-Description du langage à contacts PL7-2

Ce langage appelé langage LADDER est adapté à la programmation des traitements logiques. Les programmes écrits en langage LADDER se composent d'une succession de réseaux de contacts, véhiculant des informations logiques.

Les principaux éléments de ce langage sont :

- les contacts qui permettent de « lire » (tester) des niveaux logiques d'entrée,
- les bobines qui permettent d'« écrire » (de définir) des niveaux logiques de sortie,
- les blocs fonctions : qui permettent de réaliser des fonctions d'automatismes préprogrammées (temporisateur, compteur, programmateur cyclique...),
- les blocs opérations : qui permettent d'effectuer des opérations arithmétiques, logiques, de transfert ...

-b-2-Eléments graphiques de base

Désignation	Elément	Graphe	Fonction
Eléments de test (contacts)	- Contact NO à fermeture ou direct ou ouvert au repos		Ce contact se ferme quand l'objet bit qui le pilote (càd la variable d'entrée correspondante) est à l'état 1.
	- Contact NF à ouverture ou indirect ou fermé au repos		Ce contact s'ouvre quand l'objet bit qui le pilote (càd la variable d'entrée correspondante) est à l'état 1.
Eléments d'action (bobines)	- Bobine directe		L'objet bit associé prend la valeur 1 (càd la variable de sortie correspondant à l'adresse est à 1) quand le résultat logique de la zone test est à 1. Sinon c'est zéro.
	- Bobine inverse		L'objet bit associé prend la valeur inverse du résultat logique de la zone test.
	- Bobine d'enclenchement (Set) mémorisée		L'objet bit associé prend la valeur 1 quand le résultat logique de la zone test est à 1. Il garde cet état 1 même si le résultat de la zone test repasse à zéro. Il n'est remis à zéro que par la bobine de déclenchement.
	- Bobine de déclenchement (Reset)		L'objet bit associé est remis à zéro quand le résultat logique de la zone test est à 1. Il est remis à 1 par la bobine d'enclenchement.
	- Bobine saut à un autre réseau (Jump)		Permet un branchement à un réseau étiqueté amont ou aval.
Eléments de liaison	- Connexion horizontale		Permet de relier en série les éléments graphiques de test et d'action entre les 2 barres de potentiel.
	- Connexion verticale		Permet de relier en parallèle les éléments graphiques de test et d'action.

Tableau 2 : Extrait des mnémoniques du langage à contacts PL7-2

-b-3-Caractéristiques

* Structure d'un réseau PL7-2

Un réseau est constitué de :

- une *étiquette* numérotée de 1 à 9999 (obligatoire),
- un *commentaire* de 15 caractères maximum (facultatif),
- un *réseau de contacts*.

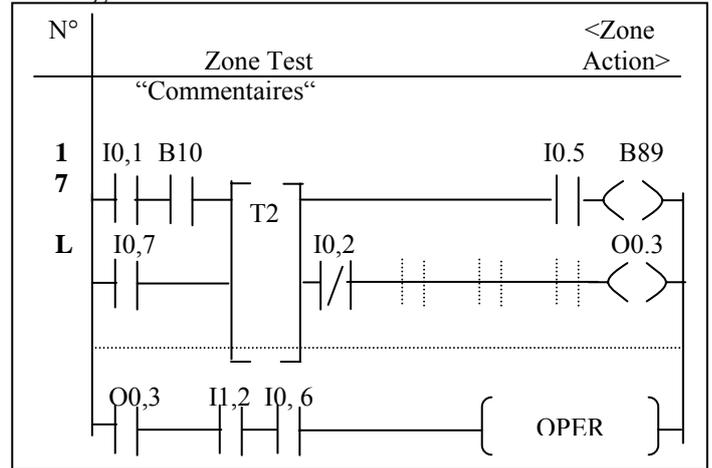
Un réseau de contacts comprend 4 lignes et dix colonnes partagées en deux zones :

- une *zone test* où on rencontre des contacts, des blocs fonctions, des blocs opérations logiques ;
- une *zone action* où on rencontre des bobines, des blocs opérations de calcul et transfert de données.

Un réseau de contacts maximum comprend :

- 4 lignes de 9 contacts (zone test),
- 4 bobines (zone action).

Figure 5.8-a :
Exemple de réseau de contacts en PL7-2
(le numéro de réseau ou étiquette est L17)



* Constitution d'un programme

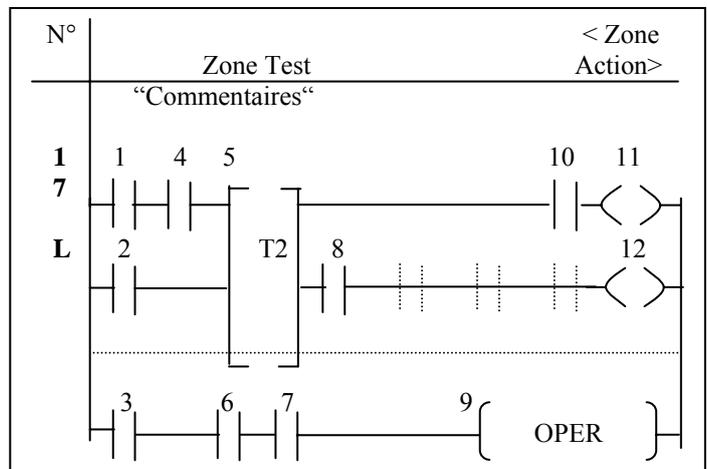
Un programme en langage à contacts est une suite de réseaux de contacts saisis dans un ordre aléatoire. Cependant la scrutation ou exécution d'un réseau se fait dans l'ordre de saisie des réseaux, et non pas selon la numérotation de leur étiquette.

* Scrutation d'un réseau

L'exécution d'un programme en langage à contacts PL7-2 a lieu de manière cyclique : durant un cycle de l'automate, on effectue la scrutation un à un de l'ensemble des réseaux. La scrutation d'un réseau se fait :

- colonne par colonne de gauche à droite,
- l'exécution des blocs (fonctions et opérations) s'effectue dès la rencontre de leur extrémité gauche.

Figure 5.8-b :
Scrutation des contacts
d'un réseau en PL7-2
(Le numéro indique l'ordre de scrutation)



-5-PROGRAMMATION EN LANGAGE BOOLEEN**-5-a-Généralités**

Pratiquement tous les API admettent ce langage qui est plus pratique et plus universel que les autres, mais moins « parlant » car à la place d'un graphe on a sous les yeux une équation booléenne. La dénomination usitée sur les automates est le mode LIST, le mode LS (langage Littéral Structuré), le mode BLOCS ou LOGIQUE qui est un mode graphique (les mnémoniques sont remplacés par des blocs à plusieurs E/S réalisant les opérations logiques de base).

Quelle que soit la nature des données de départ (schéma électrique à contacts, logigramme), la programmation nécessite en premier lieu une phase de préparation: d'abord affecter des adresses aux variables d'entrée/sortie, puis se ramener à une équation logique, ensuite identifier les opérateurs logiques nécessaires en utilisant les mnémoniques du constructeur. Ce n'est qu'après qu'on effectue la programmation proprement dite.

Le format respecte la syntaxe classique des instructions du langage assembleur, il comprend deux parties: le mnémonique (code opération) et l'adresse des entrées/sorties (opérande).

-5-b-Exemple de programmation sur TSX 21 (Télémechanique)**-5-b-1-Codes opérations de base**

MNEMONIQUE	OPERATION EFFECTUEE	TOUCHE CORRESPONDANTE
L	Lire l'état de la variable indiquée (Load)	<input type="button" value="L"/>
A	Réaliser l'opération logique ET (And)	<input type="button" value="A"/>
O	Réaliser l'opération logique OU (Or)	<input type="button" value="O"/>
=	Réaliser l'activation de la sortie indiquée par l'adresse	<input "="" type="button" value="="/>
N	Négation. Cette opération est combinée avec l'une des 4 opérations précédentes : LN:Load Not (Charger le contraire de ...) AN:And Not(Faire un ET avec le contraire de...) ON:Or Not(Faire un OU avec le contraire de ...) =N :Equal Not(Activer ou Ranger le contraire...)	<input type="button" value="LN"/> <input type="button" value="AN"/> <input type="button" value="ON"/> <input type="button" value="=N"/>

Tableau 3 : Extrait des mnémoniques du TSX 21

Remarque : l'exécution d'une instruction de lecture dans un programme (L ou LN) a pour effet de stocker automatiquement le résultat de l'instruction précédente dans un registre spécial appelé registre intermédiaire, ayant pour adresse 770. Le contenu de ce registre ou retenue peut être utilisé par le biais des instructions A770, O770, AN770, ON770. Ces instructions permettent l'économie d'utilisation de variables internes, et l'introduction d'un niveau de parenthèse.

-5-b-2-Adressage des variables d'E/S

La codification des adresses est exprimée en code octal. On dispose au maximum de 128 adresses d'entrées sorties (32 par carte), et de 64 adresses pour des variables internes (@ de 100 à 177).

Remarque : toute carte d'entrée/sortie non présente offre la possibilité d'utiliser ses 8 adresses de sortie en variables internes.

Configuration	64 Entrées	32 sorties	32 entrées/sorties programmables
Adresses de la première carte	200 à 217	230 à 237	220 à 227
Adresses de la deuxième carte	240 à 257	270 à 277	260 à 267
Adresses de la troisième carte	300 à 317	330 à 337	320 à 327
Adresses de la quatrième carte	340 à 357	370 à 377	360 à 367

-5-b-3-Exercice d'application

Ecrire les programmes correspondant aux 2 équations logiques suivantes :

$$X = a \cdot d \cdot (b + \bar{c})$$

$$Y = [\bar{a} (\bar{b} + c)] \cdot d (f + e)$$

Tableau des variables d'E/S et adresses associées

Variabes	Adresses	Variabes	Adresses
a	200	e	204
b	201	f	205
c	202	X	230
d	203	Y	231

Programmes correspondants

<u>Sortie X</u>				<u>Sortie Y</u>		
Ligne	Code Op	Adresse		Ligne	Code Op	Adresse
000	L	200		000	LN	200
001	A	203		001	L	202
002	L	201		002	ON	201
003	ON	202		003	A	770
004	A	770		004	A	203
005	=	230	005	L	205	
			006	O	204	
			007	A	770	
			008	=	231	

-6-PROGRAMMATION DES FONCTIONS SEQUENTIELLES

-6-a-Fonction mémoire

Dans le cas d'un processeur réalisé en logique câblée, cette fonction est remplie par une bascule bistable (type RS).

En logique microprogrammée, cette fonction revient à affecter l'état 0 ou 1 à une variable par une instruction de mise à un ou de mise à zéro (une adresse mémoire est bien sûr affectée à la variable). Cette fonction peut être complétée par le choix de la priorité de l'une des 2 instructions (lorsqu'elles sont présentes en même temps) : mise à un ou mise à zéro prioritaire (cas du PB 100 de Merlin Gerin ou bascules SR ou RS sur les automates Moeller).

-6-b-Fonction temporisation

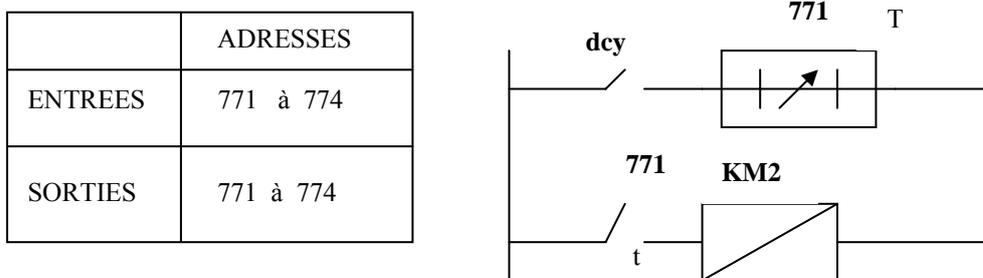
On trouve sur les automates deux solutions de temporisations : analogique et par comptage.

5-b-1-Temporisation analogique

Elle est constituée d'un circuit RC (résistance-condensateur) dont la constante de temps correspond à la base de temps de la temporisation. Sa durée est directement réglable par un bouton moleté fixé sur l'automate. Elle est enclenchée par programme.

Ainsi dans un programme son ordre d'enclenchement est considéré comme une variable de sortie ordinaire, et son déclenchement est testé comme une variable d'entrée. On peut la schématiser sur un schéma à contacts par un circuit retardateur.

L'automate TSX 21 possède ce type de temporisation. On peut utiliser quatre temporisations simultanément, dont les adresses varient de 771 à 774.



6-b-2-Temporisation par comptage

Suivant les automates, deux à quatre bases de temps sont disponibles (1/10 de seconde, seconde, 1/10 de mn ...), qui font appel aux horloges de l'automate (en réalité chaque unité de la base de temps correspond à un microprogramme préenregistré). Leur enclenchement ou armement s'effectue par programme, où la base de temps choisie ainsi que la durée doivent être spécifiées.

Les automates Crouzet CMP 31 et 34 par exemple possèdent deux bases de temps : T1 en 1/10 de seconde et T2 en 1/10 de minute. Une temporisation de deux secondes s'écrit donc dans un programme sous la forme : TEMP 1 20. La notion d'activation (enclenchement) de la temporisation n'existe pas. C'est une variable que l'on teste uniquement, et c'est sa lecture qui provoque son enclenchement.

6-b-3-Temporisation mixte

Ce troisième type de temporisation associe les avantages des deux précédentes. C'est une temporisation par comptage (réalisée par boucle de programmation qui est parcourue plusieurs fois, suivant la valeur sélectionnée dans un compteur), dont l'armement (activation) est réalisé comme pour une action classique, et dont le déclenchement est testé comme une variable d'entrée ordinaire. C'est le cas des automates TSX17 de Télémécanique, le TSX17-10 par exemple possède deux temporisations T0 et T1 de cette nature. Le seul inconvénient est qu'il faut écrire la valeur choisie pour la temporisation (base de temps et durée) à une adresse spécifique, dans une phase d'initialisation en dehors du programme proprement dit.

6-b-4- Cas de l'automate TSX 17-20 (Telemecanique)

Comme pour les automates Siemens (série Simatic) ou les automates Moeller (série PS4), la temporisation est considérée comme un bloc fonction permettant de commander avec retard des fonctions spécifiques.

La temporisation se programme de manière graphique dans le langage à contact. La saisie graphique du bloc fonction temporisation se fait à l'écran selon 2 niveaux. Le bloc est d'abord représenté dans un réseau de contacts sous la forme d'un rectangle (2 colonnes sur 3 lignes) associé à un numéro, par appui sur la **touche dynamique [T]**. A partir de cette représentation *on accède au deuxième niveau par la touche <ZM>*, ce qui visualise les paramètres du bloc fonction. A chaque paramètre modifiable est associée une touche dynamique.

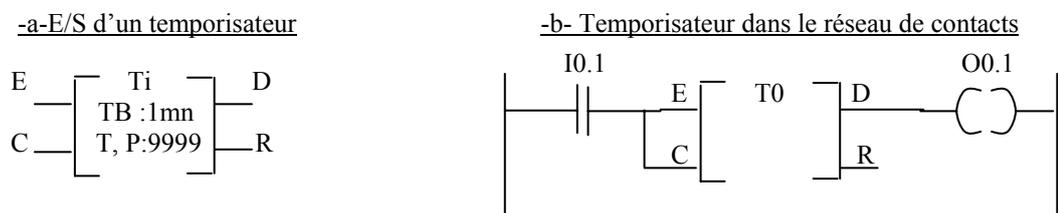


Figure 5.9: Retard à l'enclenchement d'une bobine

Chaque sortie D ou R peut piloter une / plusieurs bobine (s) ou un autre bloc fonction. Au moins une sortie d'un bloc fonction est à relier à une bobine de sortie.

Signification	Désignation	Fonctionnement
Numéro de la temporisation	Ti	Varie de T0 à T31
Base de temps	TB	1mn, 1s, 100 ms, 10 ms (1mn par défaut)
Valeur de présélection (Preset)	Ti,P (Preset)	0<Ti,P<9999 (9999 par défaut). C'est un mot qui peut être lu, testé et écrit par programme.
Valeur courante	Ti,V	C'est un mot qui décroît de Ti,P à 0 sur écoulement du temporisateur. Peut être lu, testé, mais non écrit.
Entrée « armement »	E (Enable)	Initialise le temporisateur sur front montant. Donc sur état 1 fait que Ti,V=Ti,P.
Entrée « contrôle »	C(Control)	Sur état 1 valide l'écoulement du temps. Doit donc être égal à 1 durant l'écoulement.
Sortie « temporisateur écoulé »	D (Done)	Le bit associé est égal à 1 si la temporisation est écoulée, donc si Ti,V=0.
Sortie « temporisateur en cours »	R (Running)	Le bit associé est égal à 1 si la temporisation est en cours, donc si Ti,P>Ti,V>0 et si C est à 1.

Tableau 4 : Paramètres du bloc fonction temporisateur

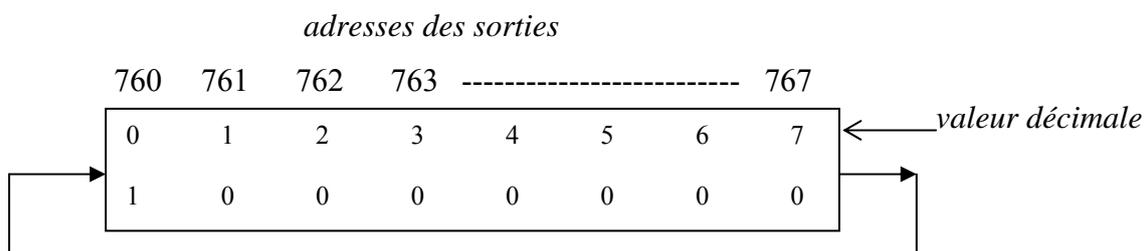
-6-c-Fonction comptage

La programmation de cette fonction peut se faire de trois manières :

- soit lire la valeur numérique d'un mot écrit (dans un code binaire, décimal ou hexadécimal) à une adresse particulière, et qui peut être incrémentée ou décrétementée. C'est le cas des structures de programmation classiques en informatique, et qui est un cas peu fréquent sur les automates ;
- soit lire l'état 1 d'une parmi n variables internes, représentant les n sorties d'un compteur, sachant qu'à chaque sortie est affectée une valeur numérique décimale comprise entre zéro et n-1. Dans ce cas le compteur se comporte comme un registre à décalage qui peut être bouclé en anneau. C'est le cas du TSX 21 où ce principe s'applique à des compteurs 16 bits ;
- soit programmer de manière graphique les différentes entrées et sorties du compteur dans l'environnement du langage à contacts. Le compteur est un bloc fonction au même titre que le temporisateur ou le programmeur cyclique (horodateur). C'est le cas des automates Siemens, Moeller et de l'automate TSX17-20 de Telemecanique.

6-c-1-Exemple de compteur 8 bits sur TSX 21

Les trois adresses 760, 761 et 763 ont une double fonction : elles sont également utilisées comme entrées de remise à zéro, et d'horloge pour le comptage et le décomptage respectivement.



6-c-2-Exemple de compteur sur TSX17-20

Le bloc fonction compteur permet d'effectuer le comptage ou le décomptage d'événements. Ces deux opérations pouvant être simultanées ou non.

Le principe de fonctionnement est le même que celui du temporisateur : le compteur se programme de manière graphique dans le langage à contacts, il est représenté dans un réseau de contacts sous la forme d'un rectangle (2 colonnes sur 4 lignes).

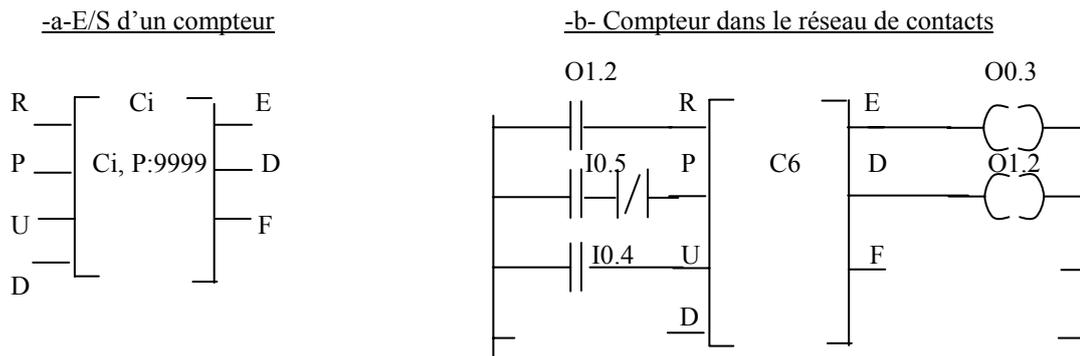


Figure 5.10 : Compteur en PL7-2

Chaque sortie E ou D ou F peut piloter une ou plusieurs bobines ou un autre bloc fonction. Au moins une sortie d'un bloc compteur est à relier directement ou indirectement à une bobine de sortie.

Signification	Désignation	Fonctionnement
Numéro du compteur	Ci	Varie de C0 à C30
Valeur de présélection (Preset)	Ci,P (Preset)	$0 < C_i,P < 9999$ (9999 par défaut). C'est un mot qui peut être lu, testé et écrit par programme.
Valeur courante	Ci,V	C'est un mot qui peut être incrémenté ou décrétementé en fonction des entrées U et D. Peut être lu, testé, mais non écrit.
Entrée de remise à zéro	R (Reset)	Sur état 1, $C_i,V = 0$.
Entrée de présélection	P(Preset)	Sur état 1 initialise le compteur : $C_i,V = C_i,P$
Entrée de comptage	U (Up)	Incrémente C_i,V sur front montant
Entrée de décomptage	D (Down)	Décrémente C_i,V sur front montant
Sortie débordement décomptage	E (Empty)	Le bit associé $C_i,E = 1$ lorsque décomptage C_i,V passe de 0 à C_i,P .
Sortie présélection atteinte	D (Done)	Le bit associé $C_i,D = 1$ lorsque $C_i,V = C_i,P$ en comptage, ou $C_i,V = 0$ en décomptage.
Sortie de débordement comptage	F (Full)	Le bit associé $C_i,F = 1$ lorsque comptage C_i,V passe de C_i,P à 0.

Tableau 5 : paramètres du bloc fonction compteur en pl7-2

-7-PROGRAMMATION EN LANGAGE GRAFCET DE SEQUENCE UNIQUE

-7-a-Généralités

L'automate se comporte comme un pas à pas électronique, dans lequel une partie de la mémoire des données est affectée à chacune des étapes du grafcet. Par conséquent à chaque étape du grafcet est associée une mémoire d'étape ou module d'étape du séquenceur, à laquelle est affectée une adresse.

L'activation et la désactivation des étapes (et par conséquent des mémoires d'étapes correspondantes) se fait selon les règles de fonctionnement normal du grafcet : l'activation de la mémoire i désactive la mémoire précédente $i - 1$.

L'activation sans condition de l'étape initiale au début du cycle se fait par une mise à un forcée de la mémoire d'étape qui lui correspond. Ensuite cette mémoire est activée et désactivée de la même manière que pour une étape ordinaire. La technique de forçage à 1 de l'étape initiale est différente d'un constructeur à un autre.

Bien que les API permettent tous de traiter les séquences uniques et les séquences multiples, les règles d'écriture varient beaucoup d'un constructeur à l'autre.

-7-b-Exécution d'un programme

L'évolution du processus est très lente par rapport au temps de réponse de l'électronique de l'automate, c'est à dire que le changement des réceptivités se fait à une fréquence très faible par rapport aux à celle des horloges de l'automate. Il y a donc plusieurs milliers de tops d'horloges avant qu'une réceptivité ne change de valeur et n'entraîne l'évolution de l'étape en aval. L'automate à déroulement cyclique de mémoire ou de phase (cf § 3.b page 120) utilise ces temps morts pour traiter tout un programme en langage grafcet.

La mémoire programme est divisée en plusieurs régions, chacune d'entre elles matérialisant une phase de calcul de l'évolution du grafcet. L'automate parcourt de façon cyclique et ordonnée l'ensemble de ces phases. Le cycle d'exécution du programme est le suivant:

Phase d'initialisation

On commence par mettre un zéro dans les adresses associées aux étapes inactives et aux sorties. Puis on met un dans les adresses associées aux étapes actives (étapes initiales).

Phase d'acquisition des entrées

Phase de calcul des transitions franchissables

La variable logique t_i associée à une transition est le produit de sa réceptivité r_i par les variables associées aux étapes amont. Elle est égale à 1 si la transition est franchissable, et à zéro sinon.

$$t_i = r_i \times \sum X_{i-1}$$

On évalue chaque variable t_i et on remplit la table des transitions franchissables.

Phase de mise à jour de la table des phases actives

L'actualisation de la table des étapes actives se fait en deux temps : d'abord la mise à zéro des variables X_i associées aux étapes devant être désactivées, puis la mise à un des X_i des variables devant être activées.

Dans tous les cas on teste la valeur des variables t_i associées aux transitions (en amont ou en aval de l'étape considérée). Lorsque ces variables sont à zéro, on saute les instructions du programme jusqu'à l'instruction correspondant à l'étude de la transition suivante.

Phase d'évolution des sorties

On affecte la valeur des sorties en fonction des nouvelles valeurs obtenues dans la table des étapes actives.

Enfin on retourne à la phase deux d'acquisition des entrées.

-7-b-Exemple de programmation sur TSX 21

Dans un processus automatisé une table est animée d'un mouvement rectiligne alternatif par un moteur M, commandé par 2 relais KM1 et KM2 (translation respectivement à droite et à gauche). Un capteur dcy assure le démarrage du cycle, et 2 capteurs d et g détectent les positions droite et gauche.

Grafset

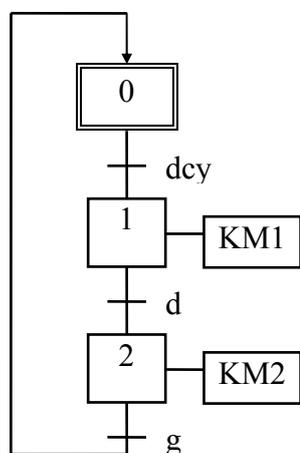


Tableau des variables d'E/S

Variable	Adresse
dcy	200
d	201
g	202
KM1	230
KM2	231

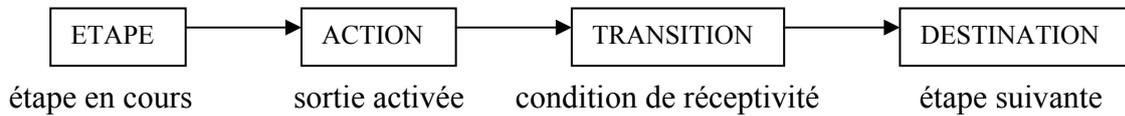
Programme

Il comprend trois phases : initialisation, séquençement, et une phase de commande des sorties. Le nombre d'étapes maximal pour cet automate est de 64, numérotées de 000 à 077.

Code Op	Adresse	Observations
L	002	Initia lisa tion
A	202	
=	000	
L	000	S é q u e n c e m e n t
A	200	
=	001	
L	001	
A	201	
=	002	
L	002	
A	202	
=	000	S o r t i e s
L	001	
=	230	
L	002	
=	231	

-7-c-Programmation sur CMP 31 et CMP 34 (Crouzet)

Ces automates n’acceptent que le langage grafcet. La conception de la programmation est par conséquent différente de celle utilisée sur les autres automates : elle s’appuie directement sur les règles du grafcet. Le tableau de programmation se présente sous la forme :



MNEMONIQUE	SIGNIFICATION	TOUCHE CORRESPONDANTE		
SORT	Sortie accompagnée de son numéro	<table border="1" style="text-align: center;"><tr><td><u>SORT</u></td></tr><tr><td>4</td></tr></table>	<u>SORT</u>	4
<u>SORT</u>				
4				
DEST	Destination, prochaine étape	<table border="1" style="text-align: center;"><tr><td><u>DEST</u></td></tr><tr><td>()</td></tr></table>	<u>DEST</u>	()
<u>DEST</u>				
()				
ENT	Entrée accompagnée de son numéro	<table border="1" style="text-align: center;"><tr><td><u>ENT</u></td></tr><tr><td>1</td></tr></table>	<u>ENT</u>	1
<u>ENT</u>				
1				
ETAP	Etape accompagnée de son numéro	<table border="1" style="text-align: center;"><tr><td><u>ETAP</u></td></tr><tr><td>3</td></tr></table>	<u>ETAP</u>	3
<u>ETAP</u>				
3				
ET	Réalise l’opération logique ET	<table border="1" style="text-align: center;"><tr><td>▼</td></tr><tr><td>ET</td></tr></table>	▼	ET
▼				
ET				
OU	Réalise l’opération logique OU	<table border="1" style="text-align: center;"><tr><td>▲</td></tr><tr><td>OU</td></tr></table>	▲	OU
▲				
OU				
()	Réalise une intersection par mise entre parenthèses	<table border="1" style="text-align: center;"><tr><td><u>DEST</u></td></tr><tr><td>()</td></tr></table>	<u>DEST</u>	()
<u>DEST</u>				
()				
/	Réalise le complément de la variable indiquée	<table border="1" style="text-align: center;"><tr><td><u>TEMP</u></td></tr><tr><td>/</td></tr></table>	<u>TEMP</u>	/
<u>TEMP</u>				
/				
TRNS	Transition accompagnée de son numéro	<table border="1" style="text-align: center;"><tr><td><u>TRNS</u></td></tr><tr><td>7</td></tr></table>	<u>TRNS</u>	7
<u>TRNS</u>				
7				
TEMP	Temporisation accompagnée de son numéro et de la durée	<table border="1" style="text-align: center;"><tr><td><u>TEMP</u></td></tr><tr><td>/</td></tr></table>	<u>TEMP</u>	/
<u>TEMP</u>				
/				

Tableau 6 : Extrait des mnémoniques du CMP 31-34

Remarque : l'étape 0 ne doit pas être utilisée par le programmeur. Elle sert à l'initialisation automatique de l'automate.

Programmation du grafcet de l'exemple du 6-b

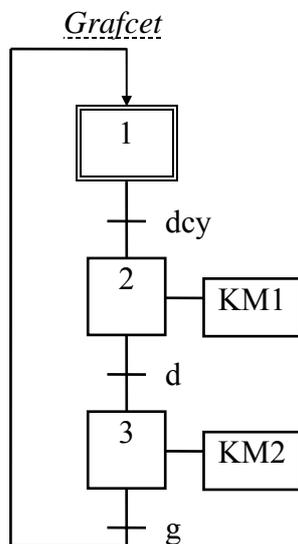


Tableau des variables d'E/S

Variable	Adresse
Dcy	1
d	2
g	3
KM1	5
KM2	6

Tableau de programmation et programme

ETAPE	ACTION	TRANSITION	DESTINATION
1		ENT 1	2
2	SORT 5	ENT 2	3
3	SORT 6	ENT 3	1

-7-d-Programmation sur PB 100 (Merlin Gerin)

-7-d-1-Généralités

Cet automate se distingue aussi bien par sa structure que son langage.

En effet il n'y a pas de zone mémoire réservée au pas à pas comme pour le TSX 21. Pour adresser les étapes d'un grafcet, on utilise des adresses de variables internes. Pour activer ou désactiver les étapes, il faut mettre à un ou à zéro les adresses qui leur ont été associées. Cela s'effectue par des instructions explicites : MU (mise à un) et MZ (mise à zéro).

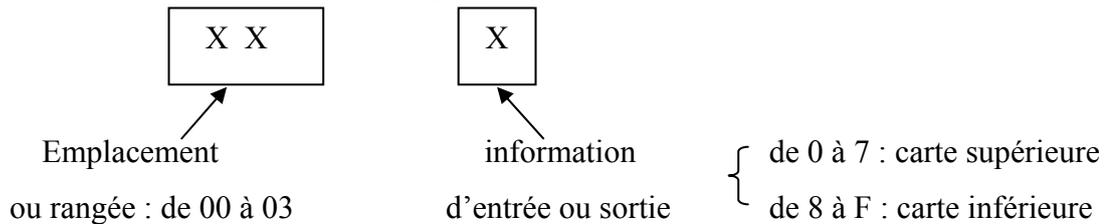
Par ailleurs toutes les opérations logiques sont décomposées et les résultats intermédiaires stockés dans des variables internes. De plus cet automate a une manière particulière d'utiliser les fonctions logiques de base ET et OU : après avoir déclaré les variables d'entrée de l'opérateur logique les unes après les autres, on appelle cet opérateur (par écriture du mnémonique ET ou bien OU) en lui passant comme opérande l'adresse d'une variable où sera stocké le résultat.

-7-d-2-Adressage des variables

On dispose au maximum de 64 adresses d'entrée/sortie réparties sur 8 cartes, chacune disposant de 8 entrées ou 8 sorties. Les cartes sont groupées par rangées de deux : une carte supérieure (généralement pour les entrées) et une carte inférieure (pour les sorties). On a donc au maximum quatre doubles rangées de cartes numérotées de 00 à 03.

Quant aux variables internes, elles sont au nombre de 512, aux adresses A00 à BFF.

La structure d'une adresse est la suivante :



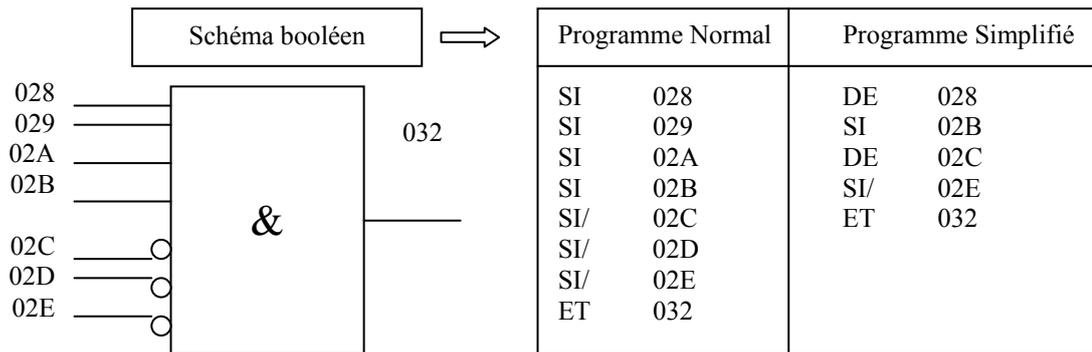
Par exemple l'adresse 03D signifie la sortie D de la carte inférieure d'emplacement n° 3.

-7-d-3-Extrait du langage de programmation

Mnémonique	Signification
SI	Lecture ou test de la variable indiquée
SI/	Test du complément de la variable indiquée
ET	Réalise le ET logique des variables précédemment lues
OU	Réalise le OU logique des variables précédemment lues
MZ	Mise à zéro de la variable dont l'adresse est indiquée
MU	Mise à un de la variable dont l'adresse est indiquée
SAUT	Instruction de saut (utilisée aussi en fin de programme)
DE	Réitération de l'instruction suivante (voir remarque)
AV	Activation de l'étape indiquée et désactivation de l'étape précédente, dont les numéros doivent se suivre obligatoirement.

Tableau 7 : Extrait des mnémoniques du PB100

Remarque: Si plusieurs informations traitées par le même code opération ont des adresses hexadécimales successives, on utilise le mnémonique DE pour alléger la programmation en évitant les répétitions, comme dans l'exemple booléen ci-dessous.



-7-d-4-Programmation de l'exemple du 6-b

Variable	Adresse	Etape	Adresse
G	010	0	A00
D	011	1	A01
Dcy	012	2	A02
KM1	028		
KM2	029		

Tableau des adresses des variables :

Entrées, Sorties, Etapes

Programme

N°ligne	Code.OP	Adresse	Observations
0DB1	DE	A01	Initialisation des étapes: mise à zéro de 1 et 2,
0DB2	MZ	A02	
0DB3	MU	A00	
0DB4	SI	012	Activation étape 1
0DB5	AV	A01	
0DB6	SI	011	
0DB7	AV	A02	Activation étape 2
0DB8	SI	010	Désactivation de la dernière étape et Réactivation de l'étape initiale
0DB9	MZ	A02	
0DBA	MU	A00	
0DBB	SI	A01	
0DBC	ET	028	Commande de KM1
0DBD	SI	A02	Commande de KM2
0DBE	ET	029	
0DBF	SAUT	0DB4	Retour à l'étape initiale

-7-e-Programmation en PL7-1 sur TSX 17-10

Le langage PL7-1 de programmation du TSX17-10 est un langage de structure booléenne destiné à la programmation des grafçets.

-7-e-1-Extrait du langage de programmation

MNEMONIQUE	TOUCHE	SIGNIFICATION
LAB	P	Label : étiquette pour un ensemble de lignes d'instructions
.*	P	Lecture de l'étape de grafçet indiquée par le numéro qui suit
=	=	Activer la sortie indiquée par l'adresse
L	L	Lecture de l'information d'entrée indiquée par l'adresse qui suit
=*=	L	Début de la zone de traitement séquentiel ou postérieur
POST	L	Argument de l'instruction « =*= » pour marquer le début de la zone de traitement postérieur (initialisation des sorties)
A	A	Réaliser l'opération logique « ET »
O	O	Réaliser l'opération logique « OU »
N	% N	Complément de la variable indiquée par l'adresse
XOR	XOR	Fonction logique ou exclusif
#	XOR	Activation de l'étape indiquée par le numéro qui suit
JMP	XOR	Saut à la ligne de programme indiquée
NOP	NOP	Instruction nulle
EP	NOP	Instruction de fin de programme
I	7	Préfixe qui précède une adresse d'entrée
O	8	Préfixe qui précède une adresse de sortie
X	6	Préfixe qui précède un numéro d'étape
TMR	4	Temporisation (suivie de l'opérande T0 ou T1)
T0		Temporisation numéro zéro

Tableau 8 : Extrait des mnémoniques du langage PL7-1

7-e-2-Adressage des variables

On peut avoir au maximum 4 cartes d'entrée/sortie (numérotées de 0 à 3) et comportant chacune 12 entrées et 8 sorties. Les entrées sont numérotées de 00 à 11 et les sorties de 00 à 07. Le format d'une adresse comporte un indicateur d'entrée (I) ou de sortie (O) suivi du numéro de la carte (0 à 3), et séparé par un point le numéro de l'entrée ou de la sortie sur la carte.

Ainsi par exemple IO.03 signifie adresse de l'entrée numéro 3 sur la carte numéro zéro, et O2.05 signifie adresse de la sortie numéro 5 de la carte numéro 2.

-7-e-3-Prise en charge de la parenthèse

Un seul niveau de parenthèse est accepté. Dans l'évaluation d'une expression, le ET logique mnémotechnique « A ») est interprété comme une parenthèse fermante.

Exemple

Soient a,b,c,d des variables d'entrée et @ le symbole de l'adresse.

-a- L @c O @d A @b

Ce programme est interprété comme étant l'expression : (c+d)b

-b- L @c A @a O @d A @b

Ce programme est interprété comme étant l'expression : (ca + d)b

-c- L @b A @a A @c O @d

Ce programme est interprété comme étant l'expression : (bac + d)

-d- L @c O @d A @b A @a

Ce programme est interprété comme étant l'expression : (c+d)ba

-e- L @a A @b L @c A @d

Ce programme est interprété comme étant l'expression : cd

-f- Comme résultat de tout ce qui précède, si l'on désire programmer l'expression (ab + cd), il faut la séparer en 2 expressions distinctes. Supposons que cette expression permet de passer d'une étape 2 à une étape 3 d'un grafctet. Le programme correspondant sera alors :

-*- 02 L @a A @b #03 L @c A @d #03

On remarquera au passage qu'il est inutile de mettre deux fois de suite l'instruction « -*- 2 » .

-7-e-4-Programmation de l'exemple du paragraphe 6-bTableau des variables d'E/S

Variables	dcy	d	g	KM1	KM2
Adresses	I0.01	I0.02	I0.03	O0.01	O0.02

Les lignes d'un programme sont numérotées à partir de zéro. Un programme est constitué de deux zones de traitement:

- **séquentiel** : où on déclare le séquençement du grafctet, c'est à dire les étapes et leurs conditions d'évolution. On déclarera donc chaque étape suivie de la condition qui fait passer de cette étape à l'étape suivante ;
- **postérieur** : où pour chaque étape, on déclare les actions (sorties) associées.

Programme

MNEMO-NIQUE	ADRESSE	COMMENTAIRES
=*=	001	Début de zone de traitement séquentiel et initialisation étape initiale (numéro1)
L	I0.01	Si dcy (d'adresse I0.01) fermé alors
#	002	Activation étape 2
-*-	002	Lire l'étape 2
L	T0	Et si information déclenchement de temporisation T0 présente
#	003	Alors activer étape 3
-*-	003	
L	I0.02	
#	004	
-*-	004	
L	T0	
#	005	
-*-	005	
L	I0.03	
#	001	Bouclage sur l'étape initiale
=*=	POST	Début de la zone de traitement postérieur (activation des sorties)
L	X02	Si étape 2 active
=	T0	Alors enclencher temporisation zéro
L	X03	
=	O0.01	Commande de KM1
L	X04	
=	T0	
L	X05	
=	O0.02	
EP		Fin de programme

-7-f-Programmation en PL7-2 sur TSX17-20

-7-f-1- Structure d'un programme

Pour programmer un grafcet en langage PL7-2 on utilise en même temps le langage à contacts et le langage grafcet. Le grafcet est "dessiné" graphiquement en se servant des primitives de base du grafcet disponibles sur l'écran, puis les réceptivités et les actions sont saisies en faisant appel au langage à contacts.

Un programme écrit en langage PL7-2 comprend plusieurs parties qui doivent être saisies dans l'ordre suivant (c'est la même structure pour le langage PL7 pro):

- Le **grafcet** proprement dit, c'est à dire un graphe qui donne le séquençement des étapes, dessiné sur l'écran à l'aide de primitives graphiques.

Dans le mode séquentiel (**SEQ**) du sous menu programme, on saisira l'ordre des étapes avec leurs numéros sans donner d'informations sur les réceptivités des transitions. La programmation se fait comme sur le TSX 21: d'abord l'initialisation (indiquer comment on arrive à l'étape initiale à partir de la dernière étape du graphe), puis le séquençement des étapes, et enfin le retour à l'étape initiale.

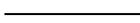
						
Etape initiale	Etape	Transition	Renvoi Indique une étape de destination ou d'origine	Activation ou désactivation simultanée d'étapes	Début et fin d'aiguillage	Bobine de passage d'une étape à une autre (cette bobine n'est utilisée que pour la saisie d'une réceptivité)

Tableau 4.9 : Eléments graphiques du grafcet utilisés en PL7-2 & PL7 pro

- Le **programme séquentiel en langage à contacts** : pour cela se positionner en face de chaque transition du grafcet précédent, puis appuyer sur la **touche ZM** (pour entrer dans "l'environnement langage à contacts"). Un réseau de contacts apparaît (cf figure 5-11) étiqueté par le numéro de l'étape précédente, dans lequel on verra comme action l'étape X_{i+1} . On saisira les réceptivités correspondant à cette transition comme conditions nécessaires pour passer de l'étape X_i à l'étape X_{i+1} .

- Le **programme postérieur en langage à contacts** : après entrée dans le mode postérieur à partir du sous menu programme, un réseau de contacts apparaît. Pour chaque réseau on doit donner obligatoirement un numéro d'étiquette programme. Pour chaque étape Xi (entrée comme condition, ie contact ouvert au repos), on donnera les actions à effectuer (bobines à activer).

- Le **programme préliminaire écrit en langage à contacts** : on entrera dans ce mode à partir du sous menu programme par la touche **PRE** et un réseau de contacts apparaît. Pour chaque réseau donner obligatoirement un numéro d'étiquette programme.

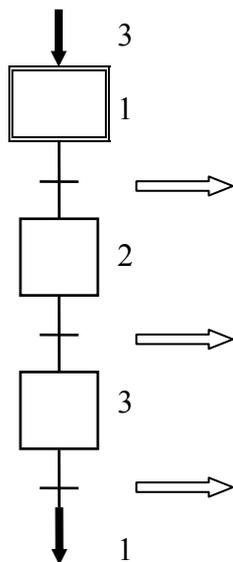
Le traitement préliminaire qui n'existe pas sur tous les grafjets, sert à prendre en compte les initialisations sur reprise secteur (*bits système SYi*) ainsi que le prépositionnement du graphe, la programmation des bits internes (**Bi**) comme des mémoires pour mémoriser des variables auxiliaires, les modes de marches de l'application, la logique d'entrée, les sécurités etc...

7-f-2-Programmation de l'exemple du paragraphe 6-b en PL7-2

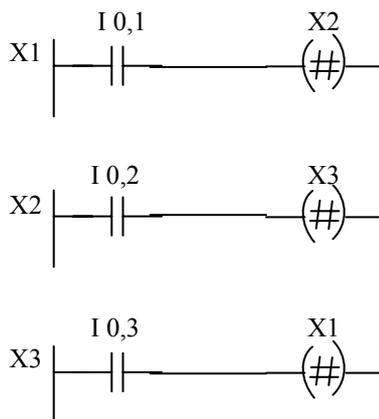
Tableau des variables d'E/S

Variables	Dcy	D	g	KM1	KM2
Adresses	I0.01	I0.02	I0.03	O0.01	O0.02

Grafjet
des étapes et transitions



Programme séquentiel
définition des réceptivités



Programme postérieur
définition des actions

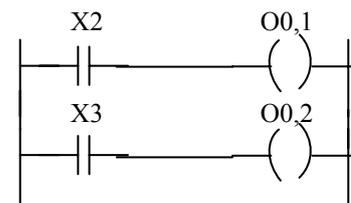
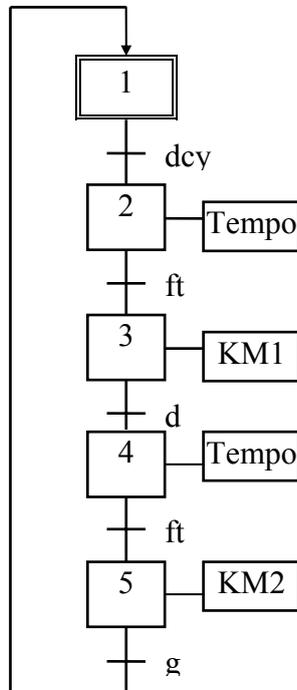


Figure 5.11 : Programme en langage PL7-2

-8- INTRODUCTION DES TEMPORISATIONS**-8-a-Exemple**

Reprenons l'exemple traité en 6-b (TSX21) et 6-c (CMP31) en conservant les mêmes adresses, et introduisons une temporisation de deux secondes, qui retarde la commande des moteurs. Le nouveau grafcet est le suivant :

**8-b-Programmation sur TSX 21**

N°lign	Code OP.	Adresse	N°lign	Code OP.	Adresse	N°lign	Code OP.	Adresse
000	L	005	011	L	003	022	L	002
001	A	202	012	A	201	023	=	771
002	=	001	013	=	004	024	L	003
			014	L	004	025	=	230
003	L	001	015	A	771	026	L	004
004	A	200	016	=	005	027	=	771
005	=	002	017	L	005	030	L	005
006	L	002	020	A	202	031	=	231
007	A	771	021	=	001			
010	=	003						

-8-c-Programmation sur CMP 31/34

ETAPE	ACTION	TRANSITION	DESTINATION
1		ENT 1	2
2		TEMP 1 20	3
3	SORT 5	ENT 2	4
4		TEMP 1 20	5
5	SORT 6	ENT 3	1

-8-d-Programmation en PL7-1 sur TSX 17-10

Tableau des variables d'E/S

Variables	dcy	d	g	ft	KM1	KM2	Tempo
Adresses	I0.01	I0.02	I0.03	T0	O0.01	O0.02	T0

Programme

MNEMONIQUE	ADRESSE	COMMENTAIRES
=*=	001	Début de zone de traitement séquentiel et initialisation étape initiale (numéro1)
L	I0.01	Si dcy (d'adresse I0.01) fermé alors
#	002	Activation étape 2
-*-	002	Lire l'étape 2
L	T0	Et si information déclenchement de temporisation T0 présente
#	003	Alors activer étape 3
-*-	003	
L	I0.02	
#	004	
-*-	004	
L	T0	
#	005	
-*-	005	
L	I0.03	
#	001	Bouclage sur l'étape initiale

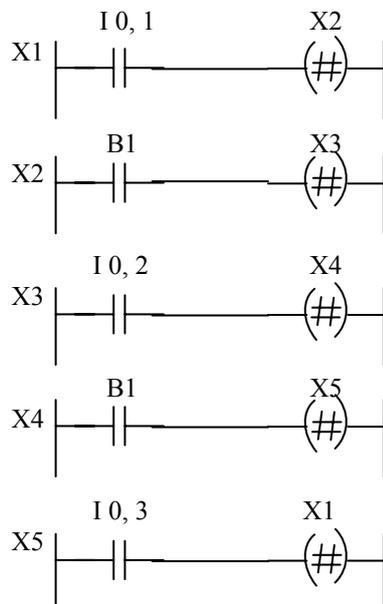
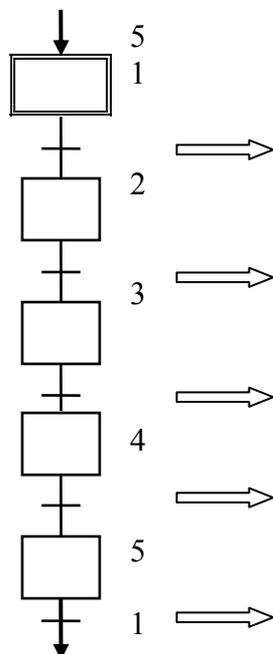
=*=	POST	Début de la zone de traitement postérieur (activation des sorties)
L	X02	Si étape 2 active
=	T0	Alors enclencher temporisation zéro
L	X03	
=	O0.01	Commande de KM1
L	X04	
=	T0	
L	X05	
=	O0.02	
EP		Fin de programme

-8-e-Programmation en PL7-2 sur TSX 17-20

Tableau des variables d'E/S

Variables	dcy	d	g	ft	KM1	KM2	Tempo
Adresses	I0.01	I0.02	I0.03	B1	O0.01	O0.02	T0

Programme séquentiel
Grafset étapes / transitions Définition des réceptivités)



Programme postérieur
définition des actions

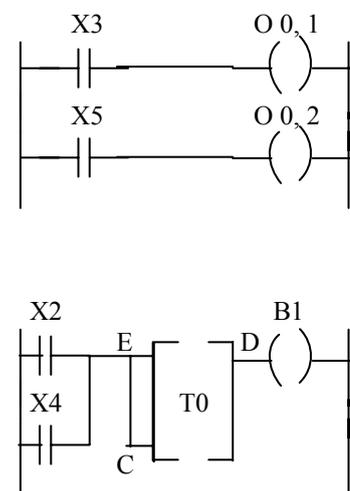


Figure 5.12 : Programme en langage PL7-2 avec temporisation

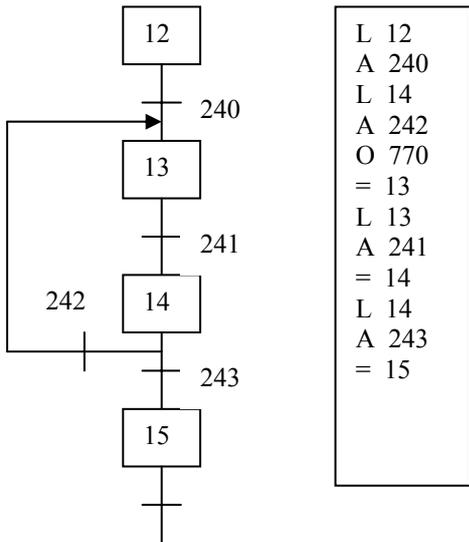
-IV-PROGRAMMATION DE GRAFCETS A SEQUENCES MULTIPLES

-1-AIGUILLAGES SUR TSX21

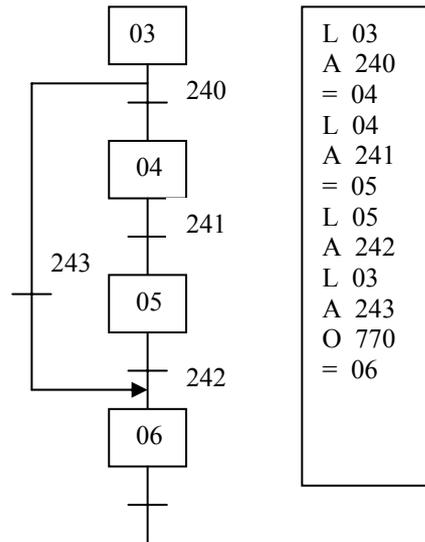
-1-a-Saut d'étapes et reprise en séquence

La programmation du saut d'étape et de la reprise en séquence doivent être prises en compte dans le séquenceur au moment de l'activation de l'étape à laquelle elles aboutissent. Cela représente une convergence en OU qui se programme par l'instruction O770.

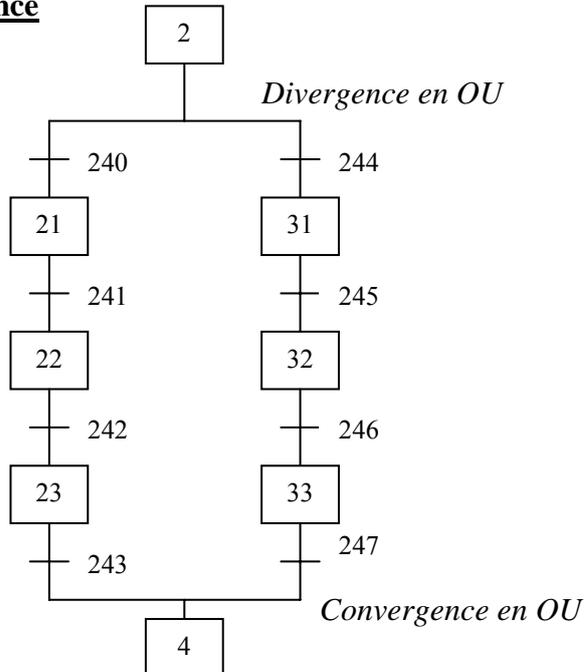
Reprise en séquence



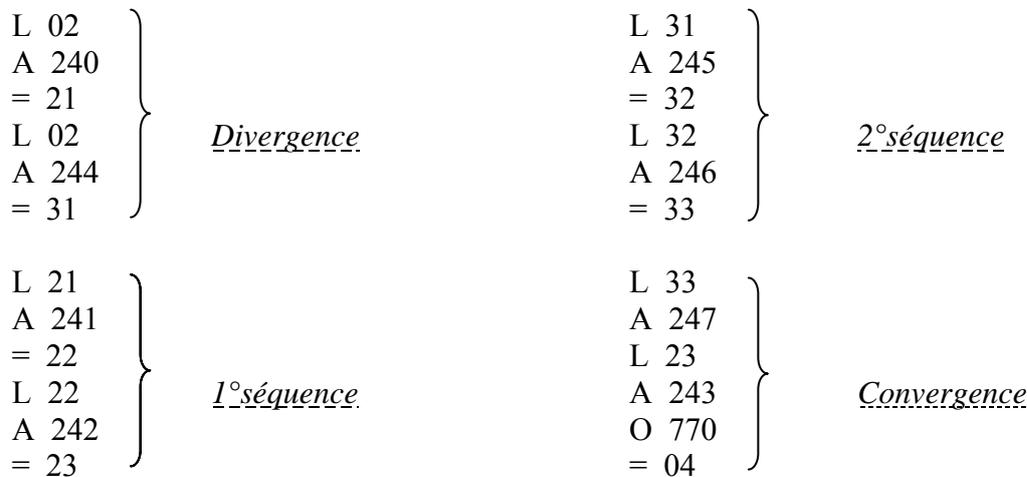
Saut d'étapes



-1-b-Sélection de séquence



On commence par programmer la divergence en OU, puis chaque séquence séparément, et enfin la convergence en OU.



-2-SEQUENCES SIMULTANÉES SUR TSX21

-2-a-Généralités et utilisation des compteurs

Pour respecter les règles du grafset, la programmation doit permettre, après l'activation simultanée des étapes d'entrée de ces séquences, une évolution des étapes actives dans chaque séquence de manière indépendante. Cela n'est possible qu'en écrivant pour chaque séquence un programme particulier qui sera pris en charge par un pas à pas électronique, indépendant du séquenceur principal (de l'API). Le séquenceur le plus utilisé pour cela est le compteur.

La structure d'un programme sera donc la suivante :

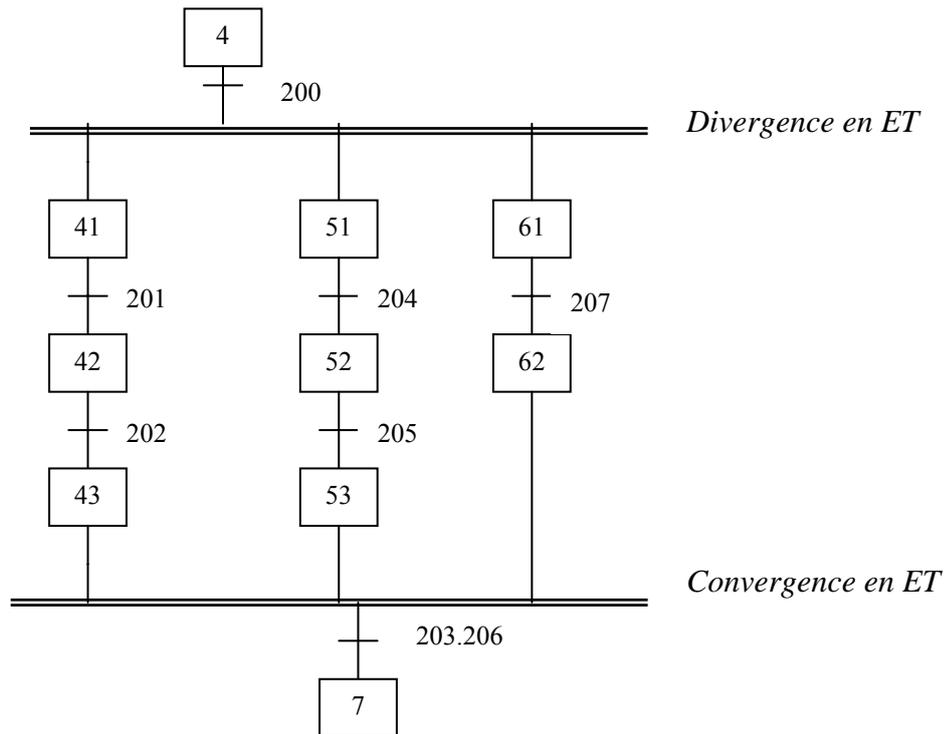
- programmer d'abord la divergence en ET (entrée dans les séquences simultanées),
- programmer chaque séquence séparément,
- programmer la convergence en ET (sortie des séquences simultanées).

Le TSX 21 possède 15 compteurs 16 bits. On peut donc avoir, en plus de la séquence principale, 15 séquences simultanées comportant chacune au maximum 16 étapes. Le principe de ces compteurs a été exposé au paragraphe III-6-c. Chaque compteur possède 16 sorties numérotées en octal correspondant aux valeurs décimales 0 à 15. Les adresses des sorties 0, 1 et 3 sont également utilisées comme entrées de remise à zéro (Raz), comptage et décomptage respectivement.

Adresses de base	Nombre de compteurs	Limites des adresses	Entrées particulières (Raz, comptage, décomptage)
400	4 : compteur 1 compteur 2 compteur 3 compteur 4	400 à 417 420 à 437 440 à 457 460 à 477	400,401, 403 420,421,423 440, 441, 443 460, 461, 467
500	4	500 à 577	
600	4	600 à 677	
700	3	700 à 757	

-2-b-Exemple de programmation

Exemple de grafcet



Remarques préliminaires à la programmation

- Au lieu d'utiliser trois compteurs (un pour chaque séquence), on peut faire prendre en charge la première séquence par le séquenceur principal. A la seconde correspondra le compteur d'adresse 400, et à la troisième le compteur d'adresse 500.
- A chaque adresse de sortie du compteur (correspondant à une valeur décimale) on affecte une étape de la séquence du grafcet.
- La remise à zéro des compteurs des séquences doit avoir lieu de préférence immédiatement à la sortie des séquences simultanées. De plus il est recommandé de prévoir cette remise à zéro comme action au niveau de l'étape initiale (ce qui est normal car à ce niveau doivent avoir lieu toutes les initialisations).
- Les adresses 401 et 501 sont utilisées comme sorties par l'API pour commander les compteurs, car ce sont les entrées d'horloges des 2 compteurs (pour le comptage). Elles sont également considérées comme des entrées de l'API pour lire la valeur décimale 1 sur les sorties des compteurs.

Le tableau de correspondance étapes-adresses sera donc le suivant :

ETAPE	ADRESSE	ETAPE	ADRESSE
51	401	61	501
52	402	62	502
53	403		

On supposera dans ce programme que les compteurs ont été remis à zéro avant l'arrivée à l'étape 4 du grafcet. Par ailleurs nous n'allons pas programmer de façon rigoureuse le grafcet (pas d'initialisation ni actions), car nous avons voulu uniquement illustrer la programmation des séquences simultanées.

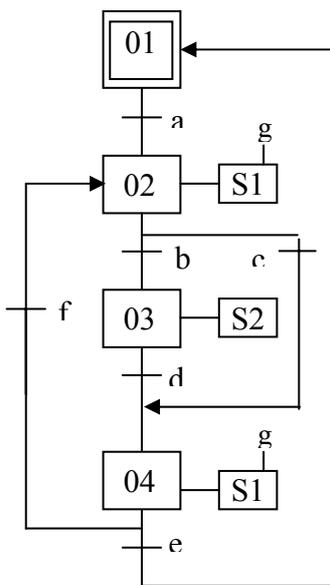
Programme correspondant

INSTRUCTIONS	COMMENTAIRES	INSTRUC-TIONS	COMMENTAIRES
L 004 } A 200 } Diver- = 041 } gence = 401 } en ET = 501 }	Si l'étape 04 est active ; Et si entrée 200 à 1 ; Alors Mise à 1 de l'étape 41, Et Mise à un des entrées de comptage des compteurs, ⇒ Les sorties des compteurs 401 et 501 seront à 1 (càd les étapes d'entrée des séquences 51 et 61 seront actives)	L 501 } A 207 } = 501 }	Si sortie 501 du compteur active (étape 61) ; & Si entrée 207 active ; Alors Activation de l'entrée de comptage ⇒ 502 active ⇒ étape 62 active
L 041 } A 201 } = 042 } Séquenc L 042 } n° 1 A 202 } = 043 }		L 043 } A 403 } A 502 } A 203 } A 206 } = 007 }	Convergence en ET
L 401 } A 204 } = 401 } Séquenc L 402 } n° 2 A 205 } = 401 }	Si sortie compteur 401 active (càd étape 51 active) ; Et Si entrée 204 active ; Alors activation entrée de comptage (désactivation de sortie 401 et activation de sortie 402) Si 402 active (càd étape 52 active) Et si entrée 205 à l'état 1, alors Activation entrée de comptage, donc désactivation de la sortie compteur 402 et activation de la sortie compteur 403 (donc passage à l'étape 53)	L 007 = 400 = 500	Raz des compteurs

-3-AIGUILLAGE SUR TSX17-10

Contrairement au TSX21, ici les sauts d'étapes et les reprises en séquence d'un grafcet se programment au moment où elles ont lieu.

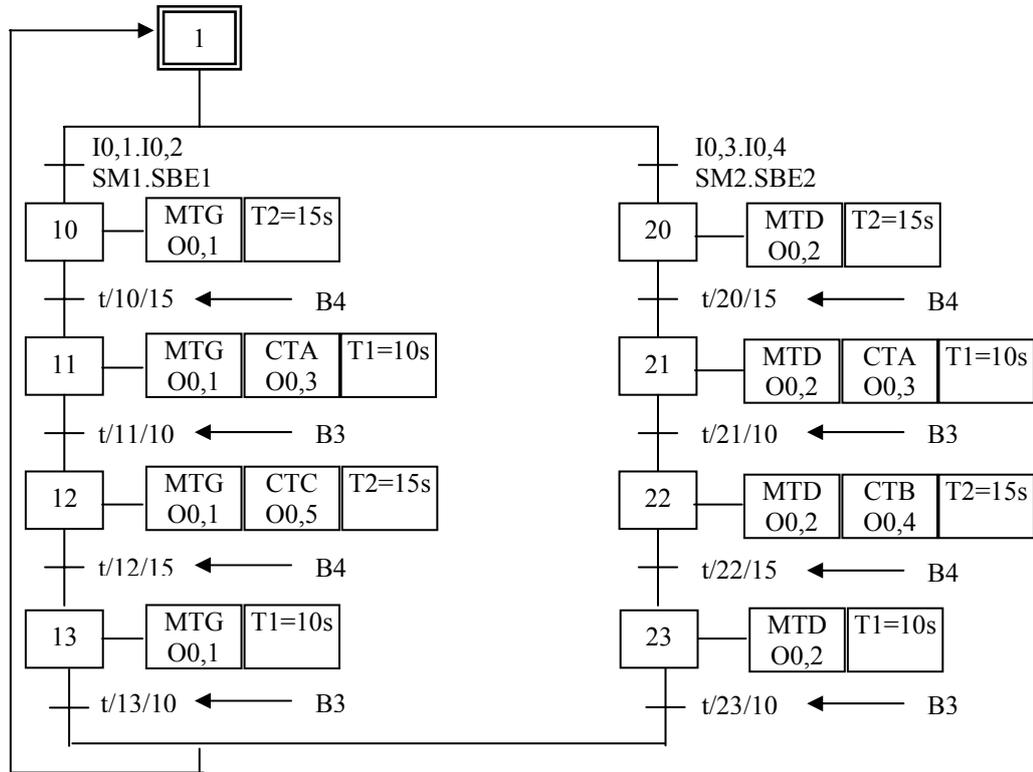
Comme conséquence, pour exprimer un choix entre deux possibilités (aiguillage), on ne doit pas mettre deux fois de suite le mnémonique «-*-» avec le même opérande (deux instructions successives «-*- N°» avec un même numéro d'étape).

Exemple de grafcetProgramme correspondant

<i>==</i> 01	<i>==</i> <i>POST</i>
L a	L X02
# 02	O X04
<i>-*-</i> 02	A g
L b	= S1
# 03	L X03
L c	= S2
# 04	EP
<i>-*-</i> 03	
L d	
# 04	
<i>-*-</i> 04	
L e	
# 01	

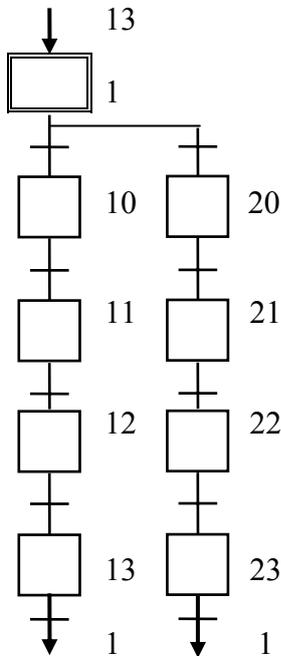
-4-AIGUILLAGE SUR TSX 17-20

-a- Grafcet avec adresses des variables

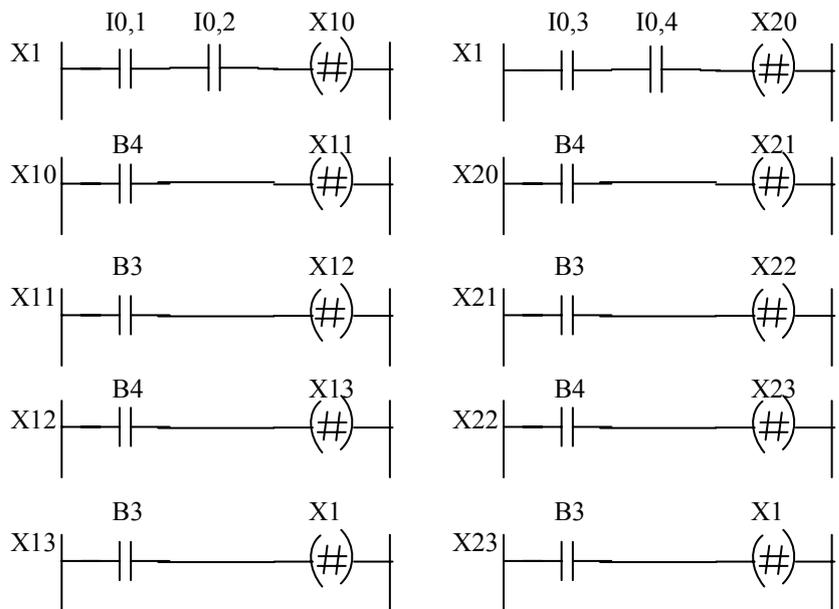


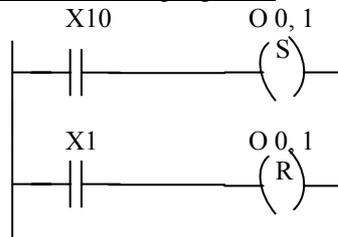
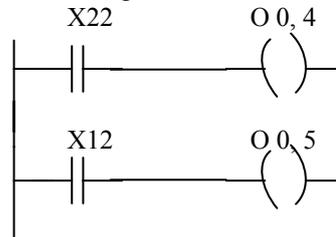
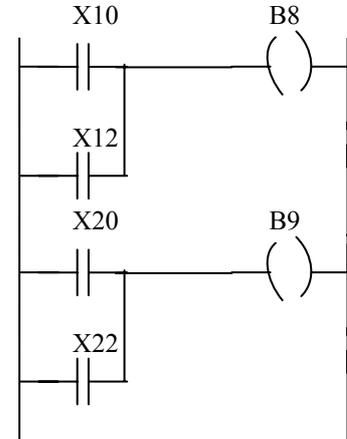
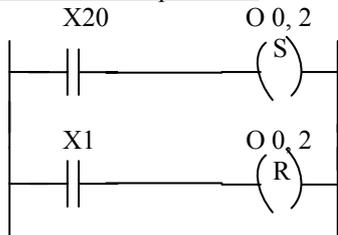
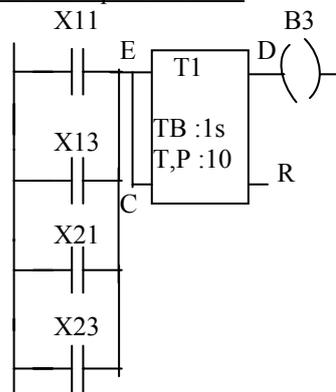
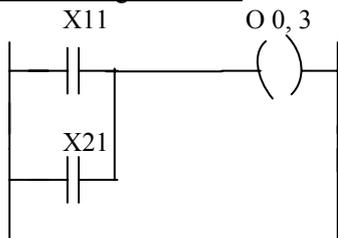
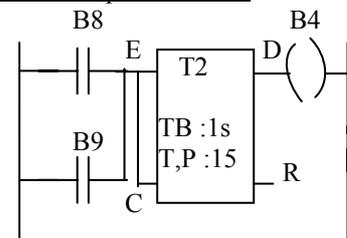
-b- Programme séquentiel

-b1- Grafcet des étapes et transitions



-b2- Définition des réceptivités

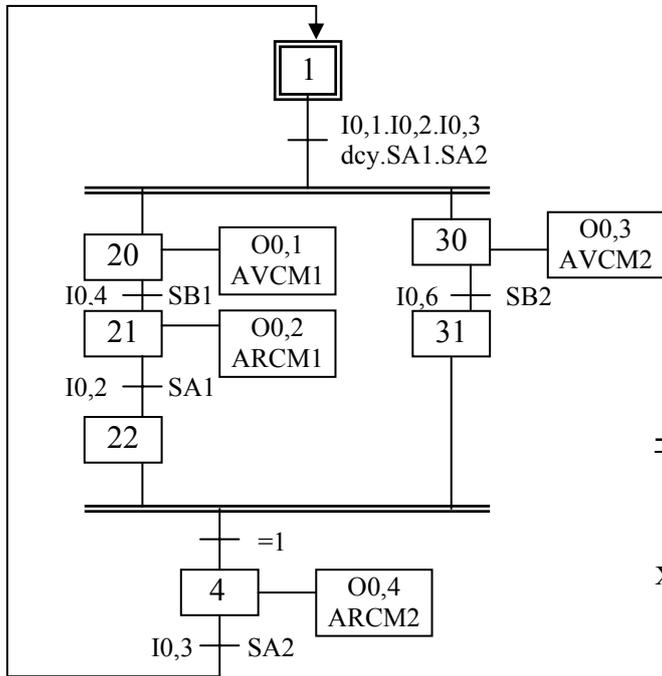


-d- Programme postérieur*définition des actions*Label 1 : moteur tapis gaucheLabel 4 : vidange trémies B et CLabel 6 : préparation tempo T2Label 2 : moteur tapis droiteLabel 5 : temporisation T1Label 3 : vidange trémie ALabel 7: temporisation T2Remarques

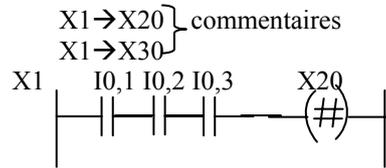
- **Attention ! Il est formellement interdit d'écrire la même sortie (action) plusieurs fois.** Elle risque d'être activée et désactivée dans la même séquence. Si une même sortie doit être activée et/ou désactivée plusieurs fois dans le grafset, utiliser pour cela des équations booléennes, mais en n'écrivant la sortie qu'une seule fois (cas de MTD et MTG).
- Pour économiser l'utilisation de plusieurs réseaux de contacts, les bobines ordinaires des actions répétitives MTG et MTD sont remplacées par des bobines bistables (mémorisation) avec mise à 1 et remise à zéro.
- Les bits internes B3 et B4 sont utilisés comme variables de fins de temporisations T1 et T2. On observera également les deux manières utilisées pour lancer les temporisations.

-5-SEQUENCES SIMULTANÉES SUR TSX 17-20

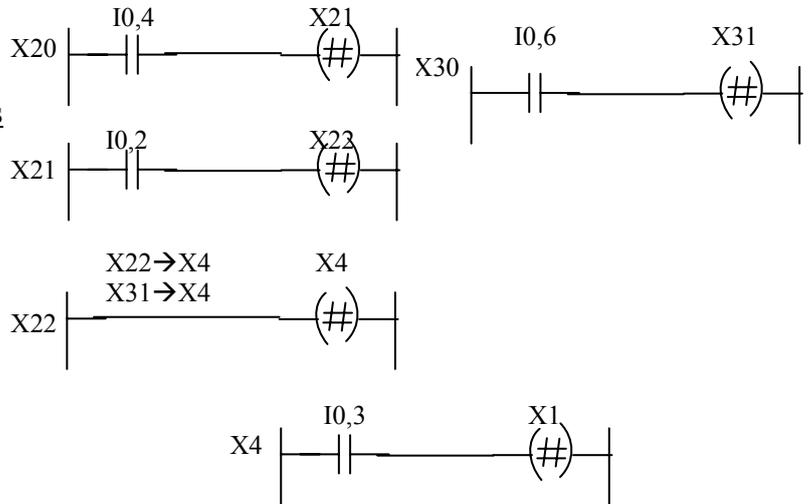
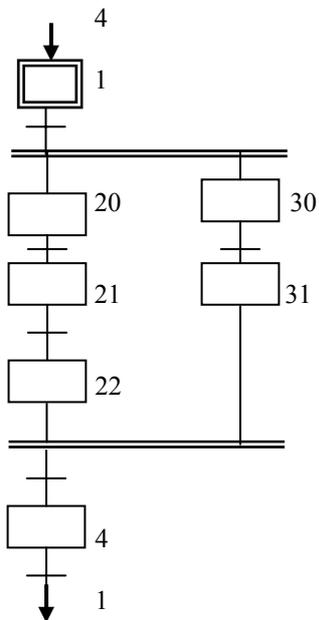
-a- Grafcet avec adresses des variables (cf. exercice 2-10 volume 2)



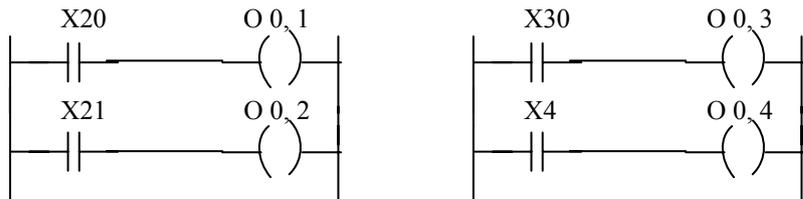
-c- Définitions des réceptivités



-b- Grafcet des étapes et transitions



-d- Actions associées



-V-CABLAGE DES AUTOMATES TSX 17

-1-CAS GENERAL

L'automate fournit l'alimentation en 24 V= (0.25A) pour les capteurs, qui seront ensuite reliés aux différentes entrées. En sortie il ne peut pas alimenter les préactionneurs: **leur alimentation est obligatoirement externe** mais elle passe par l'automate, ce dernier se contentant d'ouvrir ou fermer les contacts de ses relais de sortie en fonction des ordres d'activation ou de désactivation des sorties contenus dans le programme (dans le cas d'autres automates des transistors de sortie remplacent les relais et on y autorise ou non le passage du courant).

Quant à l'alimentation de l'automate en 220 V, elle ne se fait pas directement sur le secteur, mais on rajoute un disjoncteur KM (ou contacteur de ligne) suivi d'un relais magnétothermique (ou fusible) F5.

Pour une commande automatique par API d'un système automatisé, tous les capteurs et préactionneurs sont pris en charge directement par le programme de l'API. Cela permet de simplifier considérablement le schéma de commande et d'éliminer beaucoup de dispositifs électromécaniques, notamment ceux avec des temporisations.

Les communs des sorties de commande de l'automate TSX 17 sont soit regroupés, soit indépendants, permettant ainsi de commander des préactionneurs avec des tensions différentes. Prenons l'exemple de deux actionneurs KM1 et KM2 commandés en 220 volts, et un troisième KM3 commandé en 24 volts. Le 220V sera relié à la borne d'alimentation commune C8.11, et les sorties 8 et 9 commandent KM1 et KM2. Quant au 24V il sera relié à la borne C3 et la sortie 3 commande KM3.

Remarque

L'entrée I0,0 et la sortie O0,0 de l'automate TSX 17 ont chacune deux modes de fonctionnement : le mode normal, et les modes « Run/Stop » et « Secu » respectivement.

Si l'entrée I0,0 de l'automate est configurée en mode Run/Stop, on relie l'interrupteur 2 positions run/stop du système automatisé à l'entrée I0.0 de l'automate. Alors la position Run (marche) met en route l'API et par conséquent exécute le programme en mémoire. La position Stop (arrêt) arrête l'exécution du programme.

Si la sortie O0,0 est configurée en mode Secu , alors elle se met à l'état logique 1 si l'automate est en mode Run (interrupteur en position Run) et sans défaut, et à l'état logique 0 si l'automate est en mode Stop ou en mode Run avec présence de défaut.

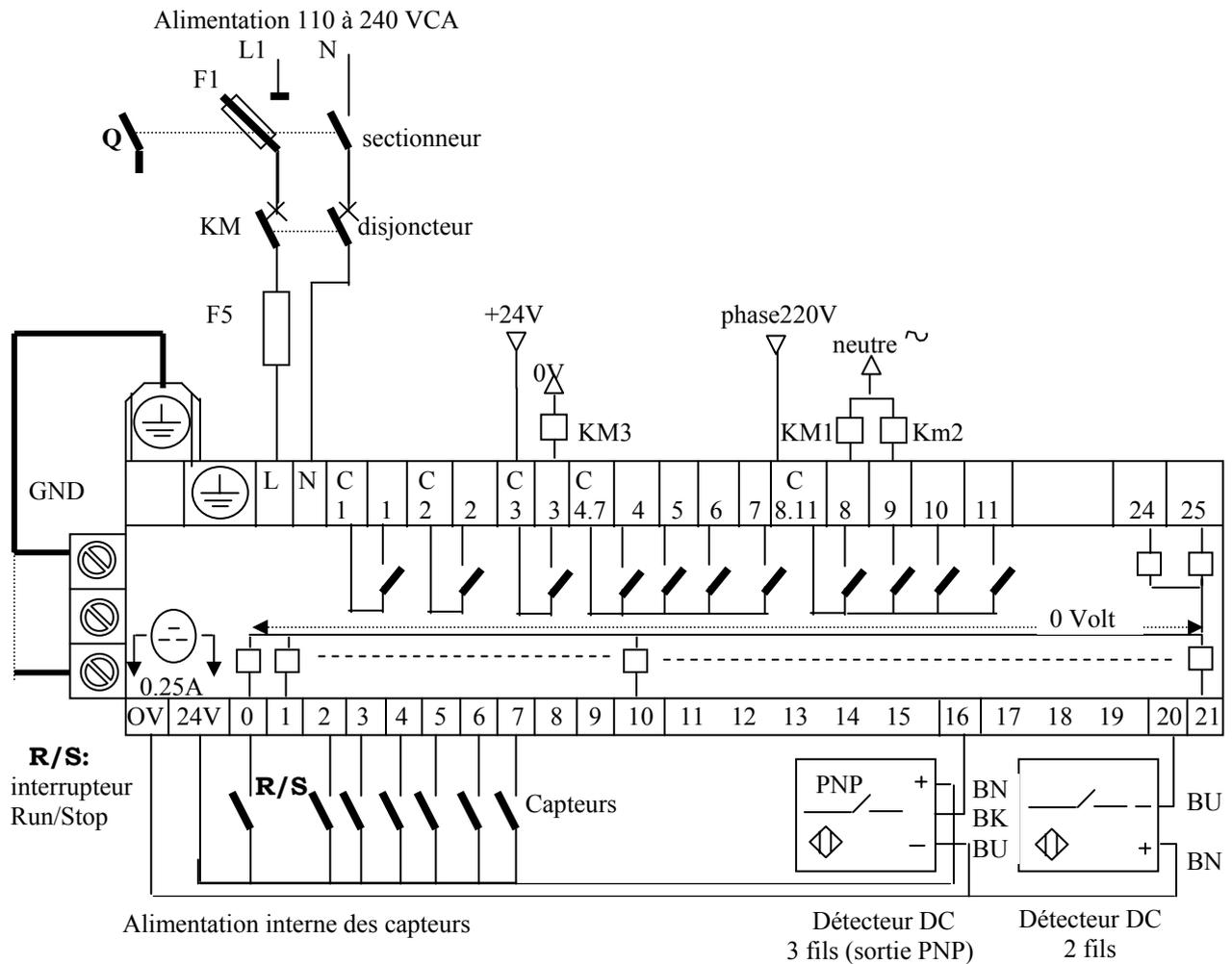


Figure 5.13 : Exemple de câblage de l'automate programmable TSX17

2-FONCTIONNEMENT EN MODE PROTEGE

En marche entièrement automatique, il est naturel de chercher à sécuriser au maximum l'installation électrique. On se sert pour cela de la sortie "SECU" de l'automate (d'adresse O0.0) pour piloter l'alimentation de l'automatisme. En mode manuel cette sortie est inutilisée.

On va donc utiliser un contacteur supplémentaire d'asservissement KA. Il sera commandé directement en mode manuel, et piloté par la sortie "Secu" en mode automatique. Ainsi dans le nouveau schéma de l'exemple précédent (cf figure 14), le transformateur et la commande des bobines des contacteurs ne seront alimentés que si le contacteur KA est alimenté. Les commandes des préactionneurs se trouvent ainsi conditionnées par le contacteur KA, qui est lui-même commandé par l'état logique 1 de la sortie Secu (bornes C0 et 0) en mode automatique.

Par ailleurs dans toute installation industrielle, avant redémarrage de l'installation consécutif à un arrêt (provoqué par une coupure secteur ou l'action sur le bouton poussoir d'arrêt d'urgence), les normes de sécurité imposent une autorisation préalable par le personnel d'exploitation : d'où l'introduction du bouton poussoir REP sur le schéma de la figure (pour reprise et acquittement ou validation de l'arrêt d'urgence).

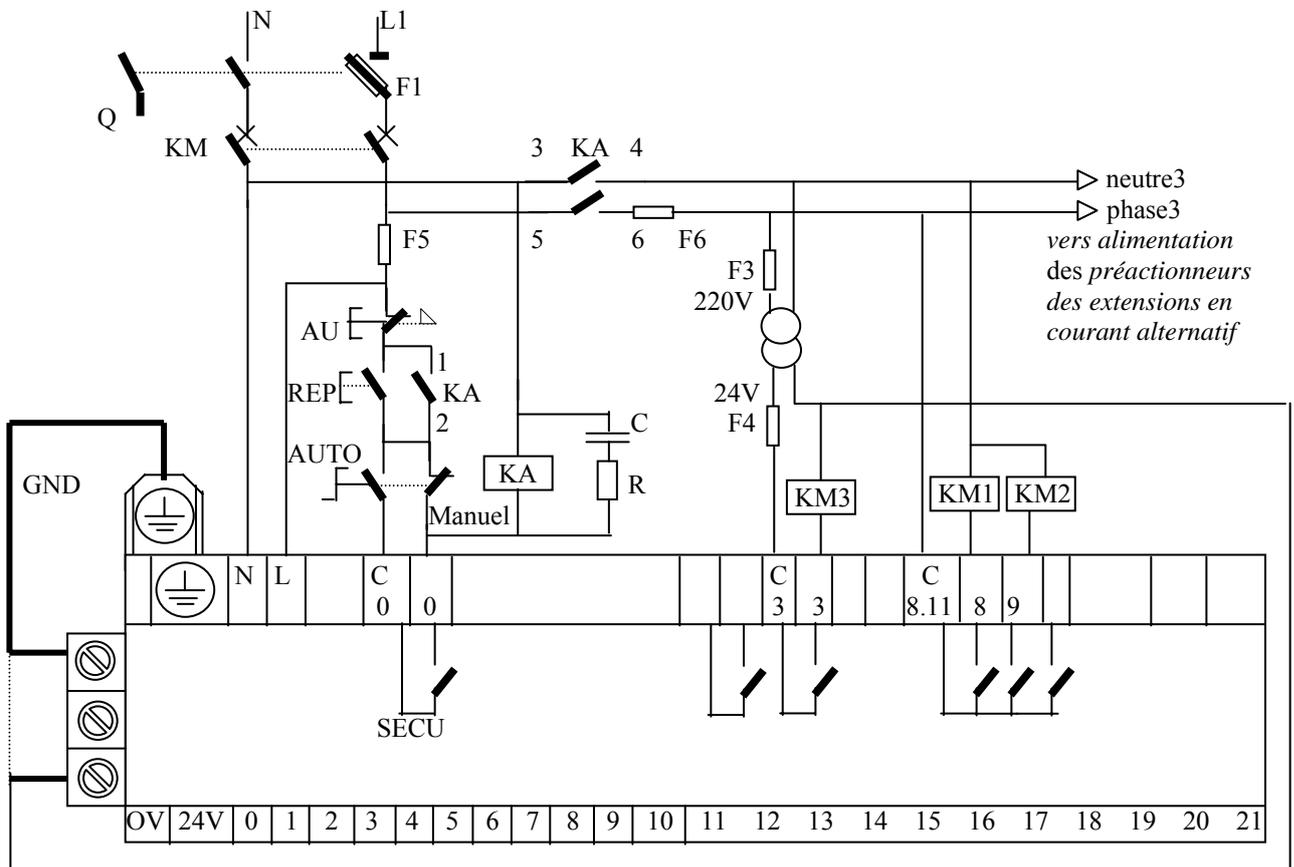


Figure 5-14 - Câblage de l'automate en mode automatique protégé

Remarque

Les bornes neutre 3 et phase 3 servent à l'alimentation des extensions qui nécessitent du courant alternatif. Ainsi par exemple le contacteur KM1 précédent (cf. figure 15) qui doit être commandé en 220 V, les bornes de sa bobine doivent être branchées entre les bornes neutre 3 et phase 3. C'est à dire phase 3 envoie le 220 V sur la borne C8.11 de l'automate, quand le relais de la sortie 8 de l'automate se ferme, le 220 V entre dans la bobine KM1. Le circuit se referme en reliant la sortie de la bobine à la borne neutre 3. La commande du contacteur KM1 est ainsi conditionnée par le contacteur d'asservissement KA.

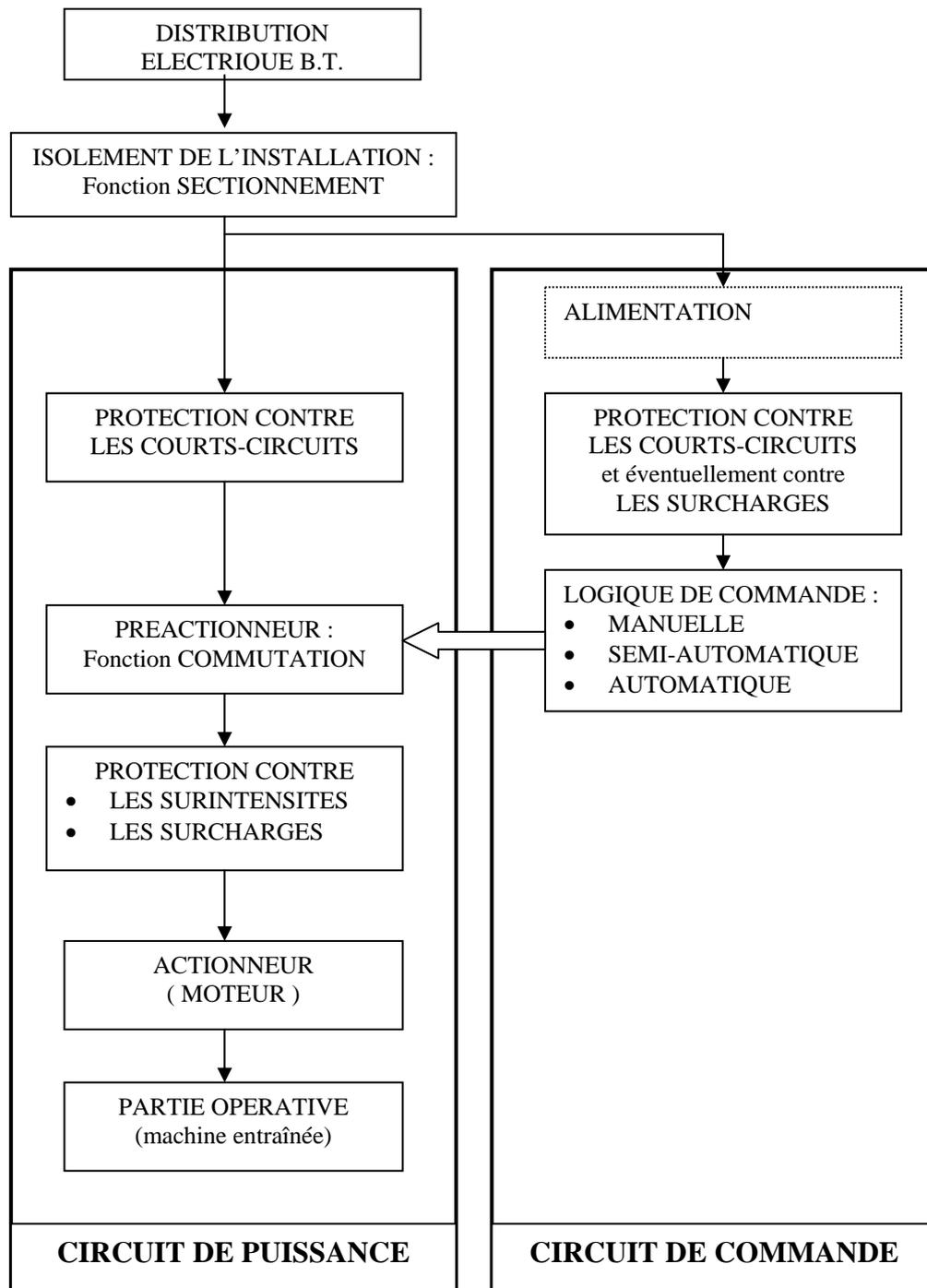
REFERENCES BIBLIOGRAPHIQUES

- A.Bianciotto & P.Boye** « L'informatique en automatisation industrielle », tome 1, collection Techniques et Normalisation, édit Delagrave, France 1995.
- G.Boujat & J.P.Pesty** « Automatismes », collection AGATI, série Bac Pro, édit Dunod, Paris 1993.
- F.Degoulange & al** « Automatismes », classes de première et terminale E,F,BT- formation permanente, édit Dunod, Paris 1983.
- P.Denis & D.Murail & al** « Les activités de la maintenance », MSMA, collection espaces technologiques, édit Delagrave, France 1994.
- Y.Lecourtier & B.Saint-Jean** « Introduction aux automatismes industriels », édit Masson 1989.
- G.Michel & C.Laurgeau & B.Espiau** « Les Automates Programmables Industriels », chapitre 1, édit 1980, France.
- Norme française NF-C03-190** «Diagramme fonctionnel GRAFCET pour la description des systèmes logiques de commande», édit Afnor (Association Française de Normalisation), 1982.
- Norme internationale IEC-848** «Etablissement des diagrammes fonctionnels pour systèmes de commande », édit CEI (Commission Electrotechnique Internationale), 1988.
- Parker pneumatic & Télépneumatic** « Constituants pneumatiques », catalogue juin 1995.
- J.Perrin & al** «Automatique industrielle », édit Dunod, Paris 1982.
- M.Pinot & al** « Du grafcet aux automates programmables », collection L.P., édit Foucher, Paris 1986.
- Télemécanique** « Automatismes Industriels », méthodes-automatismes, notice technique NT-60 010-F-8N, édit La télemécanique, France, 1980
- Télemécanique** , Documents techniques du TSX 17-10, TSX 17-20.
- R.Thomas** « Les évolutions du grafcet et de sa normalisation », CETIM informations, n°113- décembre 1989, Cetim-établissement de Saint-Etienne, France

ALIMENTATION ET COMMANDE DES CIRCUITS ELECTRIQUES

-1-STRUCTURE FONCTIONNELLE D'UNE INSTALLATION ELECTRIQUE

Les règles à observer dans la conception des installations électriques basse tension (jusqu'à 1000V en alternatif et 1500V en continu) sont définies par des normes (exemple NFC 15-100). Tous les équipements de force motrice sont construits sur le modèle fonctionnel suivant :



-2-APPAREILS MIS EN ŒUVRE DANS UN EQUIPEMENT DE DEMARRAGE

Tout système de démarrage se doit de limiter l'intensité absorbée par le moteur, tout en maintenant les performances mécaniques de l'ensemble « moteur - machine entraînée » conformes au cahier de charges.

Le démarrage du moteur peut être direct, par paliers, progressif, variable suivant une loi de vitesse.

Les deux premières solutions font appel à une technologie électromécanique, la dernière à la technologie électronique. Dans tous les cas le circuit terminal moteur dit « circuit de démarrage » doit satisfaire les quatre fonctions principales :

- Sectionnement : pour isoler ce circuit du circuit amont et permettre les interventions de maintenance en toute sécurité ;
- Protection contre les courts-circuits : avec une détection et suivie d'une coupure rapide pour éviter la détérioration de l'installation ;
- Commutation : dont la commande peut être manuelle, semi-automatique ou automatique ;
- Protection contre les surcharges : avec une détection et une coupure qui doivent éviter que toute élévation intempestive de la température du moteur n'entraîne la détérioration de ses isolants.

Un équipement de démarrage est une association d'appareils qui satisfait à ces 4 fonctions.

FONCTIONS	SOLUTIONS FONCTIONNELLES	DESIGNATION DES APPAREILS					
		SECTION-NEUR	FUSIBLE	DISJONCTEUR	RELAIS THERMIQUE	CONTACTEUR	VARIATEUR ELECTRONIQUE
Sectionnement	*par pôles spécifiques	●					
	*avec les pôles de puissance			●			
Protection contre les courts-circuits	*par fusibles		●				
	*par déclencheur magnétique			●			
Commutation	*à commande manuelle			●			
	*à commande automatique					●	
	*à commande électronique						●
Protection contre les surcharges	Par déclencheur thermique				●		

**- LOIS GENERALES D'ELECTROTECHNIQUE -
(moteur asynchrone)**

-1-PRINCIPE

Le moteur asynchrone ou moteur à induction est un transformateur à champ tournant :

- le **stator** est le *primaire*, il est alimenté par le réseau d'énergie ;
- le **rotor** en *court-circuit* et libre de tourner constitue le *secondaire*.

-2-FONCTIONNEMENT

Les courants triphasés alimentant le stator donnent naissance à un **champ tournant**, qui induit des courants dans les conducteurs du rotor. L'action du champ sur les courants rotoriques génère des **forces électromagnétiques** qui font tourner le rotor.

-3-VITESSE DE ROTATION ET GLISSEMENT

- **Vitesse du champ tournant ou vitesse de synchronisme:**

$$\Omega_S = \omega = 2\pi n_S$$

$$n_S = f/p$$

avec :

Ω_S et ω : vitesse angulaire en rad/s

n_S : vitesse en tr/s

f : fréquence du réseau en Hz

p : nombre de paires de pôles du moteur

- **Limites de vitesses :**

Pour une fréquence donnée, la vitesse ne dépend que du facteur technologique de construction : le nombre de paires de pôles;

La limite de vitesse supérieure est obtenue pour le nombre de paires de pôles minimal, soit $p=1$.

A la fréquence industrielle de 50Hz, la vitesse de synchronisme maximale est :

$$n_S = f/p = 50\text{tr/s} = 3000\text{tr/mn}$$

La limite inférieure est fonction des contraintes technologiques liées aux problèmes de réalisation et de positionnement des paires de pôles dans la machine. On ne descend généralement pas en dessous de 500tr/mn, ce qui correspond à $p=6$ paires de pôles.

- **Inversion du sens de rotation :**

Pour inverser le sens de rotation du champ tournant, il suffit d'invertir deux des trois phases d'alimentation .

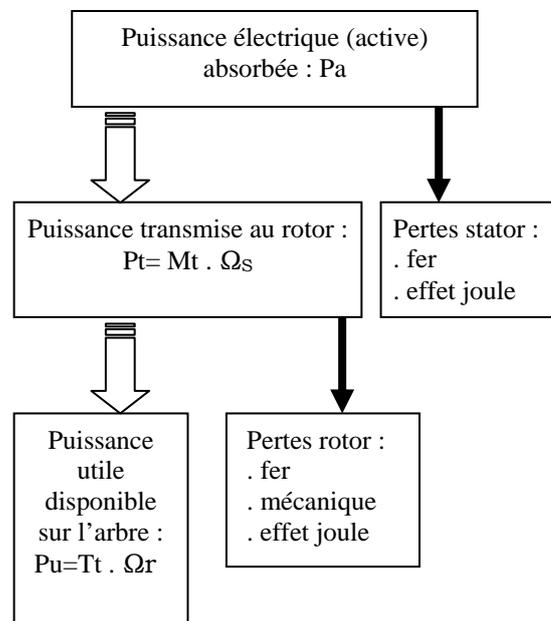
- **Vitesse rotorique, glissement**

Le rotor tourne à une vitesse Ω_r inférieure à la vitesse de synchronisme Ω_S .

La différence $\Omega_S - \Omega_r$ est dite *vitesse de glissement*, et les rapports $g = (\Omega_S - \Omega_r) / \Omega_S = 1 - \Omega_r / \Omega_S = 1 - n_r / n_S$ définissent le glissement.

-4-BILAN DES PUISSANCES, COUPLE TRANSMIS, RENDEMENTS

- **Bilan des puissances**



- **Couple transmis au rotor:** $T_t = P_u / \Omega_r$

- **Rendement industriel :** $\eta_i = P_u / P_a$

- **Rendement du rotor :** $\eta_r = 1 - g$

- **La puissance nominale P_n portée sur la plaque signalétique du moteur correspond à la puissance utile P_u**

- FORMULAIRE -

MECANIQUE

VITESSE ANGULAIRE : $\omega = 2 \pi n = 2 \pi N / 60$

ω en radians /seconde, rad/s.

n en tours par seconde, tr/s.

N en tours par minute, tr/mn.

FREQUENCE DE ROTATION SYNCHRONE

(sensiblement égale à la vitesse de rotation à vide d'un moteur asynchrone)

$$\omega = 2 \pi n = 2 \pi f / p$$

avec :

n : vitesse de rotation en tr/s

f : fréquence du réseau en Hz

p : nombre de paires de pôles du moteur

COUPLE NOMINAL : $T_n = P_n / \omega_n$

Avec :

T_n couple nominal en newtons-mètres

P_n puissance nominale en watts

ω_n vitesse angulaire nominale en rad/s

COUPLE ACCELERATEUR : $T_a = T_m - T_r$

Avec :

T_a : couple accélérateur en newtons-mètres

T_m : couple moteur en newtons-mètres

T_r : couple résistant en newtons-mètres

DUREE DE DEMARRAGE : $t = J \omega_n / T_a$

(de la vitesse 0 à la vitesse ω_n avec un couple accélérateur constant)

Avec :

t : temps de démarrage en secondes

J : moment d'inertie totale des masses en mouvement (moteur +charge) en kg.m²

ω_n : vitesse angulaire nominale en rad/s

T_a : couple accélérateur en newtons-mètres

MOMENT D'INERTIE D'UN CORPS DE

MASSE m : $J = m r^2$

Avec :

J : moment d'inertie en kg.m²

m : masse en kilogrammes

r : rayon de giration en mètres

RAYON DE GIRATION

Cylindre plein : $r = (r_1)^2$

Cylindre creux : $r = ((r_1)^2 + (r_2)^2) / 2$

Avec : r₁ = rayon extérieur

r₂ = rayon intérieur

ELECTRICITE

LOI D'OHM

• Circuit à résistance seule : $U = RI$

• Circuit à réactance seule : $U = XI$

• Circuit à réactance et résist. : $U = ZI$

Avec :

U tension aux bornes du circuit en volts

I intensité du courant en ampères

R résistance du circuit en ohms

X = X_L ou X_C : réactance du circuit en Ω

Z impédance du circuit en ohms

REACTANCE INDUCTIVE : $X_L = L \omega$

REACTANCE CAPACITIVE : $X_C = 1 / C \omega$

Avec :

L inductance en henrys

C capacité en farads

ω pulsation en rad/s

f fréquence en hertz

LOI DE JOULE (en monophasé) : $W = RI^2 t$

Avec :

W énergie dissipée en joules

R résistance du circuit en ohms

I intensité du courant en ampères

t temps de passage du courant en seconde

PUISSANCE APPARENTE : $S = \sqrt{P^2 + Q^2}$

(absorbée sur le réseau)

Avec :

S puissance apparente en voltampères

P puissance active en watts

Q puissance réactive en vars

cosφ facteur de puissance du circuit

$$\cos \varphi = P / S$$

PUISSANCE ACTIVE

• En continu : $P = UI$

• En monophasé : $P = UI \cos \varphi$

• En triphasé : $P = UI \sqrt{3} \cos \varphi$

Avec :

P puissance active en watts

U tension en volt (entre phases en triphasé)

I intensité du courant en ampères

cosφ facteur de puissance du circuit

FLUIDIQUE

DEBIT VOLUMIQUE : $D_V = S V$

DEBIT MASSIQUE : $D_m = \rho S V$

PUISSANCE MECANIQUE DE TRANSLATION : $P = F V$

$$P = F V = \rho S V^2 = \rho D_V V$$

Avec :

D_V débit volumique en m^3

D_m débit massique en kg/s

S section en m^2

V vitesse d'écoulement du fluide en m/s

masse volumique du fluide en kg

P puissance en watts (W)

F force en newtons (N)

p pression en pascals (Pa)

ELECTRICITE**PUISSANCE REACTIVE**

• En monophasé : $Q = UI \sin \varphi$

• En triphasé : $Q = UI \sqrt{3} \sin \varphi$

Avec :

Q puissance réactive en vars

U tension en volt (entre phases en triphasé)

I intensité du courant en ampères

$\cos \varphi$ facteur de puissance du circuit

COURANT ABSORBE PAR UN MOTEUR

• En continu : $I = P/U$

• En monophasé : $I = P/(U \cos \varphi)$

• En triphasé : $I = P/U \sqrt{3} \cos \varphi$

Avec :

I courant absorbé par le moteur en ampères

P puissance active en watts

U tension en volt (entre phases en triphasé)

$\cos \varphi$ facteur de puissance du moteur

PUISSANCE UTILE : $P_u = P \cdot \eta$

Avec :

P_u puissance utile disponible sur l'arbre du moteur
(= puissance nominale P_n)

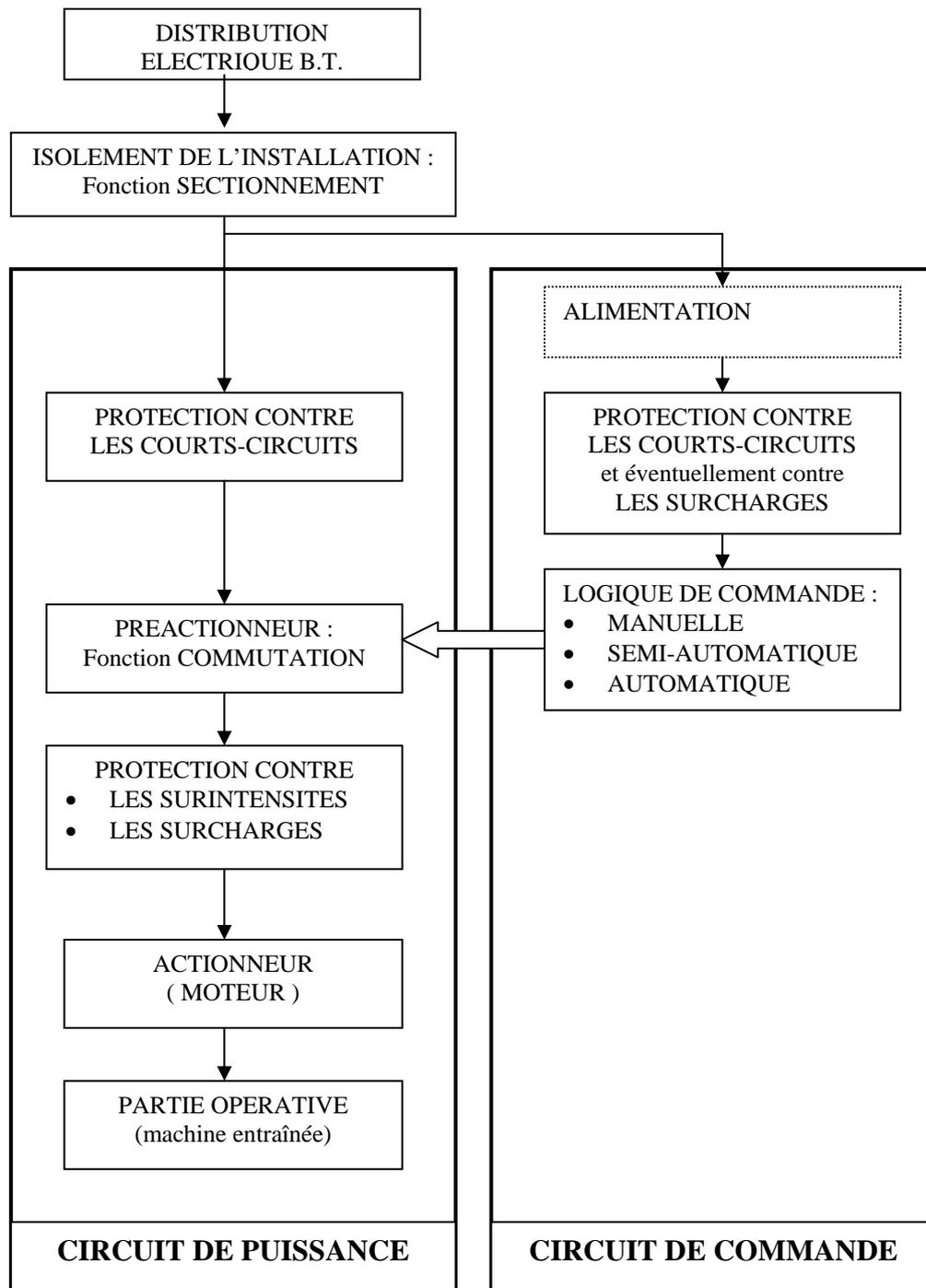
P puissance active absorbée sur le réseau

η rendement du moteur

ALIMENTATION ET COMMANDE DES CIRCUITS ELECTRIQUES

-1-STRUCTURE FONCTIONNELLE D'UNE INSTALLATION ELECTRIQUE

Les règles à observer dans la conception des installations électriques basse tension (jusqu'à 1000V en alternatif et 1500V en continu) sont définies par des normes (exemple NFC 15-100). Tous les équipements de force motrice sont construits sur le modèle fonctionnel suivant :



-2-APPAREILS MIS EN ŒUVRE DANS UN EQUIPEMENT DE DEMARRAGE

Tout système de démarrage se doit de limiter l'intensité absorbée par le moteur, tout en maintenant les performances mécaniques de l'ensemble « moteur - machine entraînée » conformes au cahier de charges.

Le démarrage du moteur peut être direct, par paliers, progressif, variable suivant une loi de vitesse.

Les deux premières solutions font appel à une technologie électromécanique, la dernière à la technologie électronique. Dans tous les cas le circuit terminal moteur dit « circuit de démarrage » doit satisfaire les quatre fonctions principales :

- Sectionnement : pour isoler ce circuit du circuit amont et permettre les interventions de maintenance en toute sécurité ;
- Protection contre les courts-circuits : avec une détection et suivie d'une coupure rapide pour éviter la détérioration de l'installation ;
- Commutation : dont la commande peut être manuelle, semi-automatique ou automatique ;
- Protection contre les surcharges : avec une détection et une coupure qui doivent éviter que toute élévation intempestive de la température du moteur n'entraîne la détérioration de ses isolants.

Un équipement de démarrage est une association d'appareils qui satisfait à ces 4 fonctions.

FONCTIONS	SOLUTIONS FONCTIONNELLES	DESIGNATION DES APPAREILS					
		SECTION-NEUR	FUSIBLE	DISJONCTEUR	RELAIS THERMIQUE	CONTACTEUR	VARIATEUR ELECTRONIQUE
Sectionnement	*par pôles spécifiques	●					
	*avec les pôles de puissance			●			
Protection contre les courts-circuits	*par fusibles		●				
	*par déclencheur magnétique			●			
Commutation	*à commande manuelle			●			
	*à commande automatique					●	
	*à commande électronique						●
Protection contre les surcharges	Par déclencheur thermique				●		

**- LOIS GENERALES D'ELECTROTECHNIQUE -
(moteur asynchrone)**

-1-PRINCIPE

Le moteur asynchrone ou moteur à induction est un transformateur à champ tournant :

- le **stator** est le *primaire*, il est alimenté par le réseau d'énergie ;
- le **rotor** en *court-circuit* et libre de tourner constitue le *secondaire*.

-2-FONCTIONNEMENT

Les courants triphasés alimentant le stator donnent naissance à un **champ tournant**, qui induit des courants dans les conducteurs du rotor. L'action du champ sur les courants rotoriques génère des *forces électromagnétiques* qui font tourner le rotor.

-3-VITESSE DE ROTATION ET GLISSEMENT

- **Vitesse du champ tournant ou vitesse de synchronisme:**

$$\Omega_S = \omega = 2\pi n_S$$

$$n_S = f/p$$

avec :

Ω_S et ω : vitesse angulaire en rad/s

n_S : vitesse en tr/s

f : fréquence du réseau en Hz

p : nombre de paires de pôles du moteur

- **Limites de vitesses :**

Pour une fréquence donnée, la vitesse ne dépend que du facteur technologique de construction : le nombre de paires de pôles;

La limite de vitesse supérieure est obtenue pour le nombre de paires de pôles minimal, soit $p=1$.

A la fréquence industrielle de 50Hz, la vitesse de synchronisme maximale est :

$$n_S = f/p = 50\text{tr/s} = 3000\text{tr/mn}$$

La limite inférieure est fonction des contraintes technologiques liées aux problèmes de réalisation et de positionnement des paires de pôles dans la machine. On ne descend généralement pas en dessous de 500tr/mn, ce qui correspond à $p=6$ paires de pôles.

- **Inversion du sens de rotation :**

Pour inverser le sens de rotation du champ tournant, il suffit d'invertir deux des trois phases d'alimentation .

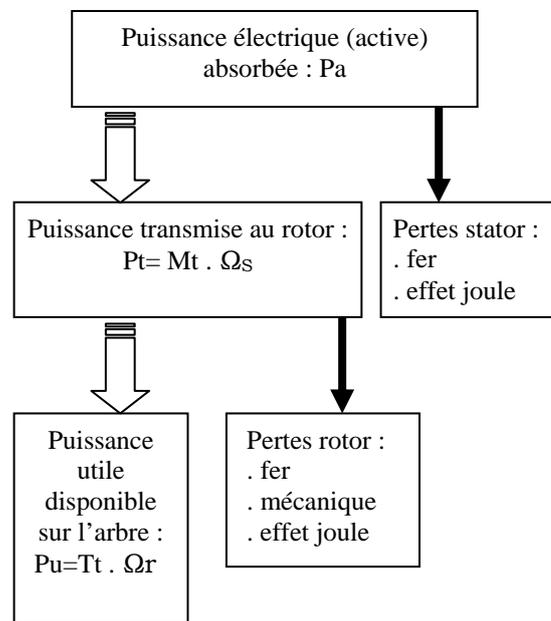
- **Vitesse rotorique, glissement**

Le rotor tourne à une vitesse Ω_r inférieure à la vitesse de synchronisme Ω_S .

La différence $\Omega_S - \Omega_r$ est dite *vitesse de glissement*, et les rapports $g = (\Omega_S - \Omega_r) / \Omega_S = 1 - \Omega_r / \Omega_S = 1 - n_r / n_S$ définissent le glissement.

-4-BILAN DES PUISSANCES, COUPLE TRANSMIS, RENDEMENTS

- **Bilan des puissances**



- **Couple transmis au rotor:** $T_t = P_u / \Omega_r$

- **Rendement industriel :** $\eta_i = P_u / P_a$

- **Rendement du rotor :** $\eta_r = 1 - g$

- **La puissance nominale P_n portée sur la plaque signalétique du moteur correspond à la puissance utile P_u**

- FORMULAIRE -

MECANIQUE

VITESSE ANGULAIRE : $\omega = 2 \pi n = 2 \pi N / 60$

ω en radians /seconde, rad/s.

n en tours par seconde, tr/s.

N en tours par minute, tr/mn.

FREQUENCE DE ROTATION SYNCHROME
(sensiblement égale à la vitesse de rotation à vide d'un moteur asynchrone)

$$\omega = 2 \pi n = 2 \pi f / p$$

avec :

n : vitesse de rotation en tr/s

f : fréquence du réseau en Hz

p : nombre de paires de pôles du moteur

COUPLE NOMINAL : $T_n = P_n / \omega_n$

Avec :

T_n couple nominal en newtons-mètres

P_n puissance nominale en watts

ω_n vitesse angulaire nominale en rad/s

COUPLE ACCELERATEUR : $T_a = T_m - T_r$

Avec :

T_a : couple accélérateur en newtons-mètres

T_m : couple moteur en newtons-mètres

T_r : couple résistant en newtons-mètres

DUREE DE DEMARRAGE : $t = J \omega_n / T_a$

(de la vitesse 0 à la vitesse ω_n avec un couple accélérateur constant)

Avec :

t : temps de démarrage en secondes

J : moment d'inertie totale des masses en mouvement (moteur +charge) en kg.m²

ω_n : vitesse angulaire nominale en rad/s

T_a : couple accélérateur en newtons-mètres

MOMENT D'INERTIE D'UN CORPS DE

MASSE m : $J = m r^2$

Avec :

J : moment d'inertie en kg.m²

m : masse en kilogrammes

r : rayon de giration en mètres

RAYON DE GIRATION

Cylindre plein : $r = (r_1)^2$

Cylindre creux : $r = ((r_1)^2 + (r_2)^2) / 2$

Avec : r₁ = rayon extérieur

r₂ = rayon intérieur

ELECTRICITE

LOI D'OHM

• Circuit à résistance seule : $U = RI$

• Circuit à réactance seule : $U = XI$

• Circuit à réactance et résist. : $U = ZI$

Avec :

U tension aux bornes du circuit en volts

I intensité du courant en ampères

R résistance du circuit en ohms

X = X_L ou X_C : réactance du circuit en Ω

Z impédance du circuit en ohms

REACTANCE INDUCTIVE : $X_L = L \omega$

REACTANCE CAPACITIVE : $X_C = 1 / C \omega$

Avec :

L inductance en henrys

C capacité en farads

ω pulsation en rad/s

f fréquence en hertz

LOI DE JOULE (en monophasé) : $W = RI^2 t$

Avec :

W énergie dissipée en joules

R résistance du circuit en ohms

I intensité du courant en ampères

t temps de passage du courant en seconde

PUISSANCE APPARENTE : $S = \sqrt{P^2 + Q^2}$
(absorbée sur le réseau)

Avec :

S puissance apparente en voltampères

P puissance active en watts

Q puissance réactive en vars

cosφ facteur de puissance du circuit

$$\cos \varphi = P / S$$

PUISSANCE ACTIVE

• En continu : $P = UI$

• En monophasé : $P = UI \cos \varphi$

• En triphasé : $P = UI \sqrt{3} \cos \varphi$

Avec :

P puissance active en watts

U tension en volt (entre phases en triphasé)

I intensité du courant en ampères

cosφ facteur de puissance du circuit

FLUIDIQUE

DEBIT VOLUMIQUE : $D_V = S V$

DEBIT MASSIQUE : $D_m = \rho S V$

PUISSANCE MECANIQUE DE TRANSLATION : $P = F V$

$$P = F V = \rho S V^2 = \rho D_V V$$

Avec :

D_V débit volumique en m^3

D_m débit massique en kg/s

S section en m^2

V vitesse d'écoulement du fluide en m/s

masse volumique du fluide en kg

P puissance en watts (W)

F force en newtons (N)

p pression en pascals (Pa)

ELECTRICITE**PUISSANCE REACTIVE**

• En monophasé : $Q = UI \sin \varphi$

• En triphasé : $Q = UI \sqrt{3} \sin \varphi$

Avec :

Q puissance réactive en vars

U tension en volt (entre phases en triphasé)

I intensité du courant en ampères

$\cos \varphi$ facteur de puissance du circuit

COURANT ABSORBE PAR UN MOTEUR

• En continu : $I = P/U$

• En monophasé : $I = P/(U \cos \varphi)$

• En triphasé : $I = P/U \sqrt{3} \cos \varphi$

Avec :

I courant absorbé par le moteur en ampères

P puissance active en watts

U tension en volt (entre phases en triphasé)

$\cos \varphi$ facteur de puissance du moteur

PUISSANCE UTILE : $P_u = P \cdot \eta$

Avec :

P_u puissance utile disponible sur l'arbre du moteur
(= puissance nominale P_n)

P puissance active absorbée sur le réseau

η rendement du moteur

- AUTOMATISMES LOGIQUES : modélisation et commande -

VOLUME 2 : EXERCICES CORRIGES

- 2° édition revue et augmentée -

<u>SOMMAIRE</u>	<u>PAGE</u>
PARTIE I: DIMENSIONNEMENT DE VERINS ELECTROPNEUMATIQUES ET UTILISATION DES DISTRIBUTEURS	1
PARTIE II: MODELISATION D'AUTOMATISMES PAR GRAFCET	19
PARTIE III: COMMANDE PAR SEQUENCEURS ELECTRIQUES Annexe: compléments sur le câblage des séquenceurs électriques	66
PARTIE IV: COMMANDE PAR API	82
PARTIE V: MODELISATION D'AUTOMATISMES PAR RDP	98
PARTIE VI: MODELISATION GEOMETRIQUE DE ROBOTS	114
PARTIE VII: COMMANDE DE MACHINES ASYNCHRONES : MODES SEMI-AUTOMATIQUE ET AUTOMATIQUE	138

- AUTOMATISMES LOGIQUES : modélisation et commande -

VOLUME 3 : DIMENSIONNEMENT ET CABLAGE DES AUTOMATISMES ELECTRIQUES

- 2° édition revue et augmentée -

- TABLE DES MATIERES -

- PAGES-

Chapitre 6 : FONCTIONS SECTIONNEMENT, PROTECTION ET COMMUTATION D'UN AUTOMATISME ELECTRIQUE	192
-I- Structure d'une installation électrique	
-1-Structure fonctionnelle d'une installation électrique	193
-2-Réseau d'alimentation et rappel des lois fondamentales d'électricité	
-3-Sectionnement ou isolement de l'installation	194
-4-Circuit de puissance	195
-a-Protection contre les courts-circuits	
-b-Protection contre les surintensités et les surcharges	
-5-Circuit de commande	198
-a-Alimentation	
-b-Protection: disjoncteur de contrôle ou cartouche fusible	
-II- Préactionneurs ou organes de commutation: principes de fonctionnement	201
-1-Organe de commutation à commande manuelle : le disjoncteur et le disjoncteur moteur	
-2-Organe de commutation à commande automatique : le contacteur	201
-3-Organe de commutation à commande électronique : le variateur électronique	213
-III- Appareils mis en œuvre dans un équipement de démarrage	213
-IV- Conception du schéma de commande d'un système automatisé	216
-V- Exemple de synthèse : commande d'un moteur de ventilateur	220-226
 Chapitre 7 : COMMANDE DES MOTEURS ELECTRIQUES	 227
-I- Les moteurs : principes généraux	
-1- Moteur à courant continu	
-2- Moteur pas à pas	
-3- Moteur asynchrone triphasé	228
-II- Démarrage des moteurs asynchrones triphasés à cage	
-1-Démarrage direct	232
-2-Réduction du courant de démarrage	
*par commutation montage étoile-triangle	233
*par élimination de résistances statoriques ou rotoriques	236
-III- Freinage des moteurs asynchrones triphasés à cage	
-1- Freinage en contre-courant	241
-2-Freinage par injection de courant continu	243
-IV- Exemple de synthèse : commande d'une porte de garage	245
-1- Commande semi-automatique	247
-2- Commande automatique par API (TSX 17-10)	255-260
 Chapitre 8 : EXERCICES SUR LA COMMANDE DES MOTEURS en modes semi-automatique et automatique	 261
-1- Moteur asynchrone à bagues en triphasé, mode semi-automatique	
-2- Moteur asynchrone à cage à deux sens de marche	
-3-Application de monte charge en triphasé	
-4- Moteur à 2 sens de marche en catégorie AC3	
-5- Table de perçage en monophasé, modes semi automatique et automatique	
-6- Commande en triphasé de deux chariots par API TSX 17-10	
-7-Station de transport de gravier en triphasé par API TSX 17-10	
 Annexes	 280
-I- Formulaire (mécanique, électricité)	
-II- Lois générales d'électrotechnique	