

POLYTECH' Marseille
Département de Mécanique Énergétique
2^e Année Option S.I.I.C

A.P.I.

Automates Programmables Industriels

L. BERGOUGNOUX

2004–2005

Table des matières

1	Généralités sur les S.A.P.	3
1.1	Description des différentes parties	3
1.1.1	La partie opérative	3
1.1.2	La partie commande	4
1.1.3	La partie relation	4
1.2	Différents types de commande	5
1.2.1	Le système automatisé combinatoire	5
1.2.2	Le système automatisé séquentiel	6
1.2.3	La logique programmée	8
1.2.4	La logique câblée	8
1.2.5	Les systèmes asservis	8
1.3	Domaines d'application des systèmes automatisés	8
1.3.1	Les avantages	9
1.3.2	Les inconvénients	9
1.3.3	Exemples de domaines d'application	9
1.4	Exemples de quelques capteurs	13
1.4.1	Les capteurs Tout Ou Rien	13
1.4.2	Les capteurs sans contact	13
1.4.3	Codes à barres	13
2	Les A.P.I.	15
2.1	La structure	15
2.1.1	Description des éléments d'un API	16
2.1.2	Langages de programmation pour API	17
2.2	Conception d'un automatisme séquentiel	19
3	Le GRAFCET	20
3.1	But du GRAFCET	21
3.2	Description du GRAFCET	23
3.2.1	Les étapes	23
3.2.2	Les transitions	23

3.2.3	Les liaisons orientées	24
3.2.4	Notation des entrées/sorties	24
3.2.5	Les actions	25
3.2.6	Les réceptivités	27
3.2.7	Les temporisations	27
3.3	Les règles d'évolution du GRAFCET	28
3.4	Les structures de base	29
3.4.1	Saut d'étape et reprise de séquence	29
3.4.2	Aiguillage entre deux ou plusieurs séquences	30
3.4.3	Parallélisme	30
3.4.4	Sous-grafcet ou séquence répétée	31
3.4.5	Gestion de conflits	33
3.5	GRAFCET de niveau 1 et 2	34
3.5.1	GRAFCET de niveau 1	34
3.5.2	GRAFCET de niveau 2	34
3.6	Modes de marches et d'arrêts	36
3.6.1	Cycle unique ou cycle par cycle	36
3.6.2	Cycle par cycle ou continu	37
3.6.3	Intervention réglages et dégagement	38
3.6.4	Arrêts	40

Chapitre 1

Généralités sur les Systèmes Automatisés de Production

Ce chapitre permet de comprendre la structure d'un Système Automatisé de Production et de définir les différentes parties de ce système. Un système de production est dit automatisé lorsqu'il peut gérer de manière autonome un cycle de travail préétabli qui se décompose en séquences et/ou en étapes.

Les systèmes automatisés, utilisés dans le secteur industriel, possèdent une structure de base identique. Ils sont constitués de plusieurs parties plus ou moins complexes reliées entre elles :

- la partie opérative (PO) ;
- la partie commande (PC) ou système de contrôle/commande (SCC) ;
- la partie relation (PR) de plus en plus intégrée dans la partie commande.

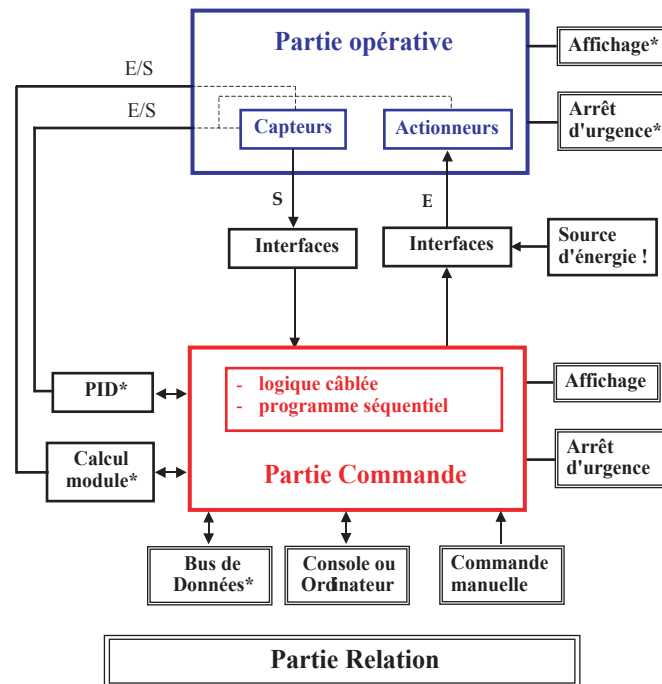
1.1 Description des différentes parties

1.1.1 La partie opérative

C'est la partie visible du système. Elle comporte les éléments du procédé, c'est à dire :

- des pré-actionneurs (distributeurs, contacteurs) qui reçoivent des ordres de la partie commande ;
- des actionneurs (vérins, moteurs, vannes) qui ont pour rôle d'exécuter ces ordres. Ils transforment l'énergie pneumatique (air comprimé), hydraulique (huile sous pression) ou électrique en énergie mécanique ;
- des capteurs qui informent la partie commande de l'exécution du travail. Par exemple, on va trouver des capteurs mécaniques, pneuma-

tiques, électriques ou magnétiques montés sur les vérins. Le rôle des capteurs (ou détecteurs) est donc de contrôler, mesurer, surveiller et informer la PC sur l'évolution du système.



* en option

FIG. 1.1 – Procédé automatisé

1.1.2 La partie commande

Ce secteur de l'automatisme gère selon une suite logique le déroulement ordonné des opérations à réaliser. Il reçoit des informations en provenance des capteurs de la Partie Opérative, et les restitue vers cette même Partie Opérative en direction des pré-actionneurs et actionneurs. L'outil de description de la partie commande s'appelle le GRAphe Fonctionnel de Commande Étape / Transition (GRAFCET).

1.1.3 La partie relation

Sa complexité dépend de l'importance du système. Elle regroupe les différentes commandes nécessaires au bon fonctionnement du procédé, c'est à dire marche/arrêt,

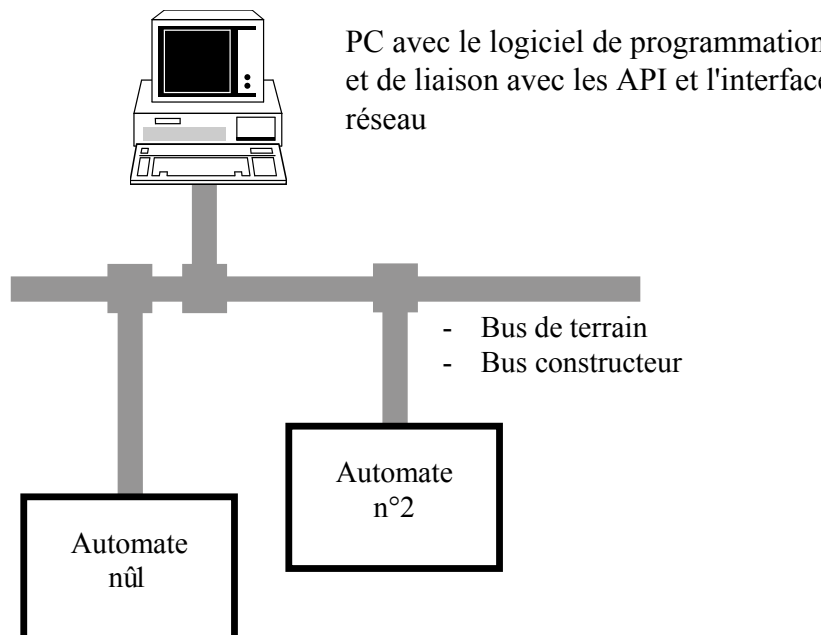


FIG. 1.2 – Forme commerciale des composants de la partie commande

arrêt d'urgence, marche automatique, etc... L'outil de description s'appelle le Guide d' Études des Modes de Marches et d' Arrêts (GEMMA).

Les outils graphiques, que sont le GRAFCET et le GEMMA, sont utilisés par les automaticiens et les techniciens de maintenance.

1.2 Différents types de commande

1.2.1 Le système automatisé combinatoire

Ces systèmes n'utilisent aucun mécanisme de mémorisation : à une combinaison des entrées ne correspond qu'une seule combinaison des sorties. La logique associée est la logique combinatoire. Les outils utilisés pour les concevoir sont l'algèbre de Boole, les tables de vérité, les tableaux de Karnaugh.

Exemple sur la figure 1.4 : si la présence de l'objet à transférer est détectée par le capteur de présence "p", alors le vérin-poussoir P entrera en fonction si l'opérateur du poste 1 ou celui du poste 2, appuie sur le bouton poussoir correspondant (BP1 ou BP2).

Les systèmes automatisés utilisant la technique "combinatoire" sont aujourd'hui très peu utilisés. Ils peuvent encore se concevoir sur des mécanismes simples où le nombre d'actions à effectuer est limité (ex : pilotage de 2 vérins). Ils présentent en outre l'avantage de n'utiliser que très peu de composants

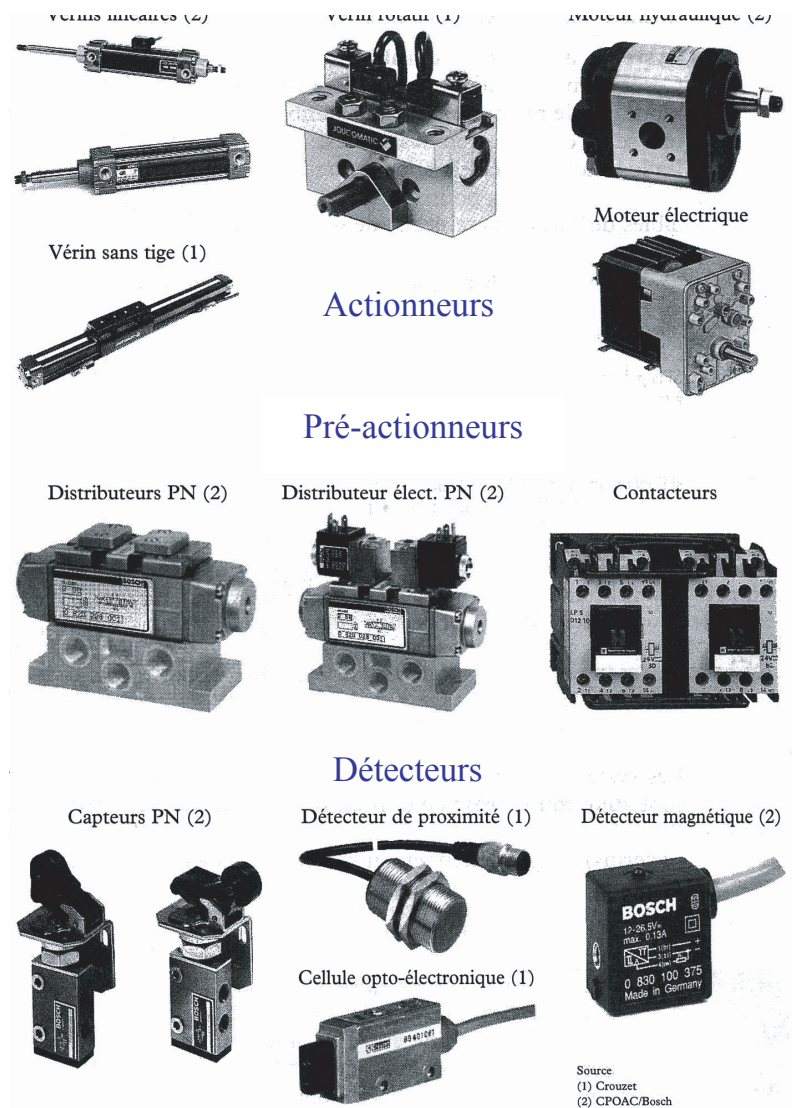


FIG. 1.3 – Forme commerciale des composants de la partie opérative

(vérins, distributeurs, capteurs, cellules).

1.2.2 Le système automatisé séquentiel

Ces systèmes sont les plus répandus dans le domaine industriel. Le déroulement du cycle s'effectue étape par étape. A une situation des entrées peuvent correspondre plusieurs situations de sortie. La sélection d'une étape ou d'une autre dépend de la situation antérieure du dispositif.

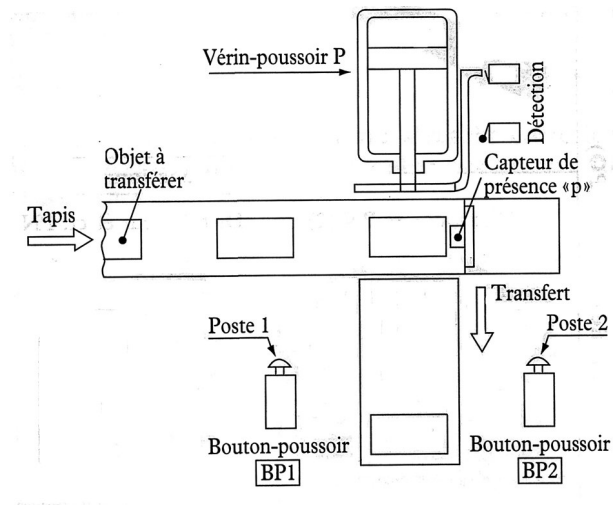


FIG. 1.4 – Exemple de système automatisé combinatoire

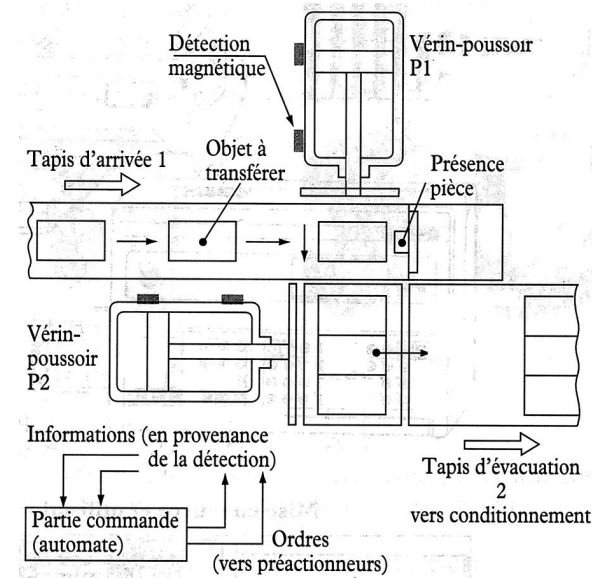


FIG. 1.5 – Exemple de système automatisé séquentiel

Exemple sur la figure 1.5 : le système place les objets côte à côte 3 par 3, sur un tapis d'évacuation 2. Ainsi placés les objets sont emmenés vers un dispositif de conditionnement sous film plastique rétractable. La logique associée est appelée logique séquentielle. Elle peut être avec commande :

- pneumatique, c'est alors de la logique câblée ;
- ou électrique, c'est de la logique programmée.

1.2.3 La logique programmée : commande électrique

L'élément principal s'appelle l'Automate Programmable Industriel ou l'API. La détection est électrique. Le pilotage des actionneurs se fait par l'intermédiaire de relais ou de distributeurs. Il existe sur le marché de nombreuses marques d'automates : Télémécanique, Siemens, Omron, Allen Bradley, Cetel, etc. . .

1.2.4 La logique câblée : commande pneumatique

L'élément principal s'appelle module séquenceur et l'association de modules constitue un ensemble appelé séquenceur. La détection est pneumatique, le pilotage des distributeurs se fait par une action de l'air comprimé sur un piston qui fait déplacer le tiroir du distributeur à droite ou à gauche. L'ensemble, appelé tout pneumatique, est homogène et fiable.

Forme commerciale des composants

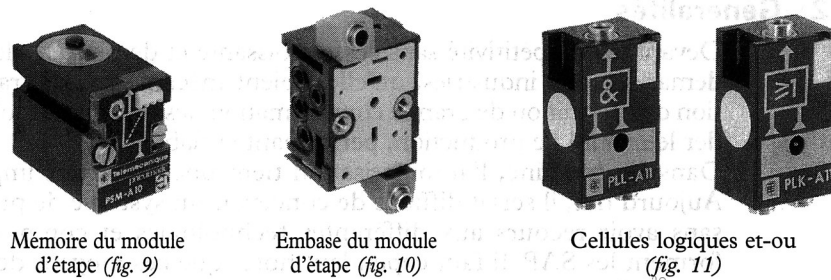


FIG. 1.6 – Éléments de logique câblée

1.2.5 Les systèmes asservis

Pour ces systèmes, on désire que la sortie suive avec précision les variations de l'entrée, et ceci avec un temps de réponse réduit. C'est par exemple le cas avec une direction assistée d'automobile ou la commande des gouvernes d'un avion. Applications : les robots industriels.

1.3 Domaines d'application des systèmes automatisés

Aujourd'hui, il serait difficile de concevoir un système de production sans avoir recours aux différentes technologies et composants qui forment

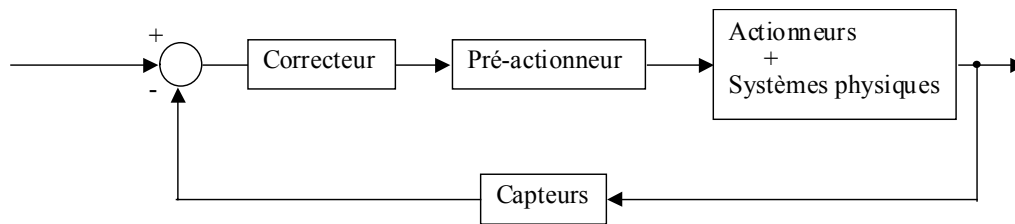


FIG. 1.7 – Schéma d'une boucle d'asservissement

les systèmes automatisés.

1.3.1 Les avantages

- La capacité de production accélérée ;
- L'aptitude à convenir à tous les milieux de production ;
- La souplesse d'utilisation ;
- La création de postes d'automaticiens.

1.3.2 Les inconvénients

- Le coût élevé du matériel, principalement avec les systèmes hydrauliques ;
- La maintenance doit être structurée ;
- La suppression d'emplois.

1.3.3 Exemples de domaines d'application

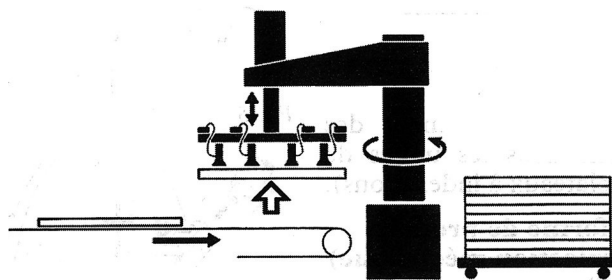


FIG. 1.8 – Le conditionnement, par ex. le déplacement d'objets suivant un angle quelconque ou le conditionnement sur palette après emballage

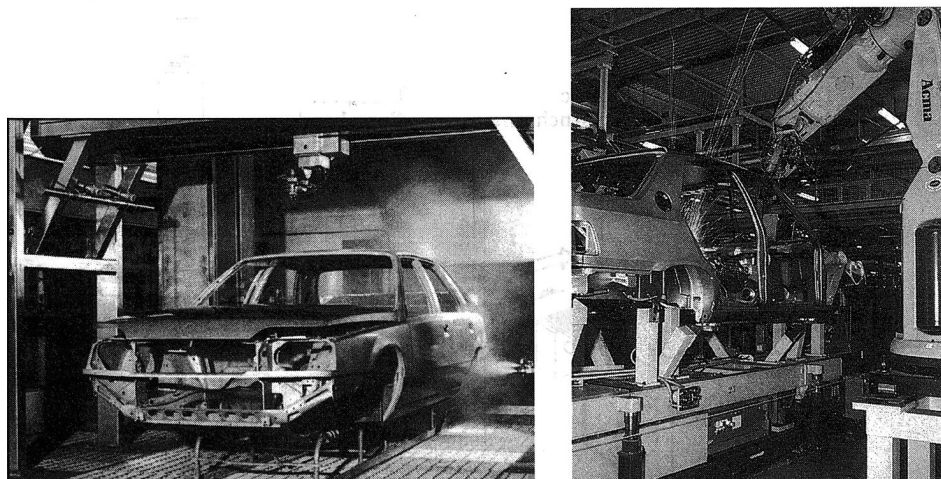


FIG. 1.9 – L'industrie automobile avec l'utilisation de robots industriels pour effectuer l'assemblage et la peinture des carrosseries

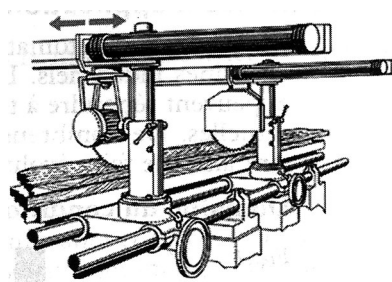


FIG. 1.10 – L'industrie du bois avec les opérations de débit, de sciage et d'usinage du bois

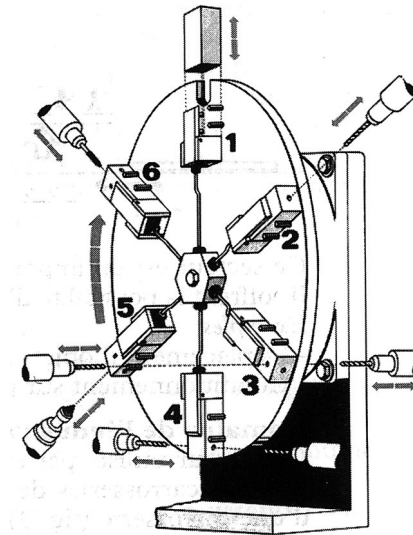


FIG. 1.11 – Machine outil : l'automatisation est ici assez importante. L'une des principales applications est dans les unités de perçage.

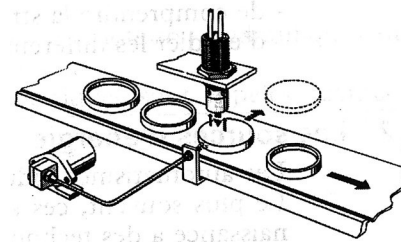


FIG. 1.12 – Contrôle de produits : Détection de défauts en bout de chaîne de production.

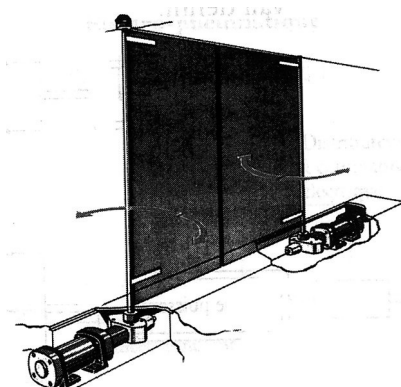


FIG. 1.13 – Automatisation de services : ouvertures programmées de portes et fenêtres, gestion centralisée de bâtiment.

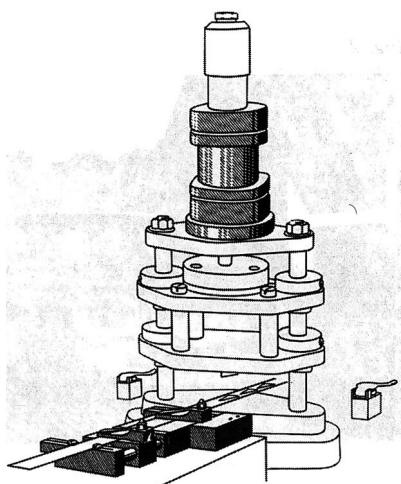


FIG. 1.14 – Mise en forme de produits finis : poinçonnage de tôles et leur mise en forme, travail dans des conditions extrêmes et environnement difficile.

1.4 Exemples de quelques capteurs

1.4.1 Les capteurs Tout Ou Rien

Dès qu'une grandeur physique est détectée (ou change d'état), ils délivrent en sortie un signal électrique ou une pression pneumatique. Il existe plusieurs capteurs T.O.R., par exemple :

- Les capteurs à commande manuelle destinés à l'équipement des pupitres et des postes de commande comme des boutons poussoirs, les boutons à 2 ou 3 positions, les arrêts "coup de poing". La commande est fournie par l'opérateur ;
- Les capteurs à commande mécanique ou interrupteurs de position situés sur la partie opérative, ils détectent par contact la présence d'une partie mobile (par ex : la tige d'un vérin).

1.4.2 Les capteurs sans contact

Citons par exemple :

- Les capteurs de proximité à jet d'air. Ces capteurs fluidiques permettent la détection sans contact mécanique, par réflexion d'un jet d'air, de la présence ou du passage d'une pièce. Le fonctionnement est assuré par un relais amplificateur. Détection de 0 à 6 mm ;
- Les détecteurs de proximité magnétique. Les Interrupteurs à Lamelle Souple (ILS) se ferment au passage d'un aimant. Ils ont comme avantages : l'élimination des problèmes mécaniques, un encombrement réduit (donc on peut monter plusieurs capteurs côte à côte), une endurance élevée (10^7 manœuvres). Dans cette catégorie, il existe aussi les détecteurs inductifs pour les métaux ferreux, capacitifs pour les métaux non ferreux, infrarouges pour localiser une source de chaleur.
- Les détecteurs de passage photoélectrique (barrage ou réflexion) pour lesquels la source de lumière est une diode électroluminescente infrarouge.

1.4.3 Codes à barres

Ce procédé de détection, dont l'invention remonte en 1950, permet une identification précise et sans erreur d'un produit. Il est très employé dans les domaines agro-alimentaires, pharmaceutiques et industriels. Une étiquette, composée de barres noires et blanches alternées verticales, est lue par un crayon optique. Il existe deux familles de codes :

- Les codes dits "alimentaires" composés de barres et d'espaces dont la largeur peut varier de 1 à 3 ;
- Les codes industriels toujours composés de barres et d'espaces larges représentant le 1 logique et de barres et d'espaces minces représentant le 0 logique ;

Représentation des étiquettes

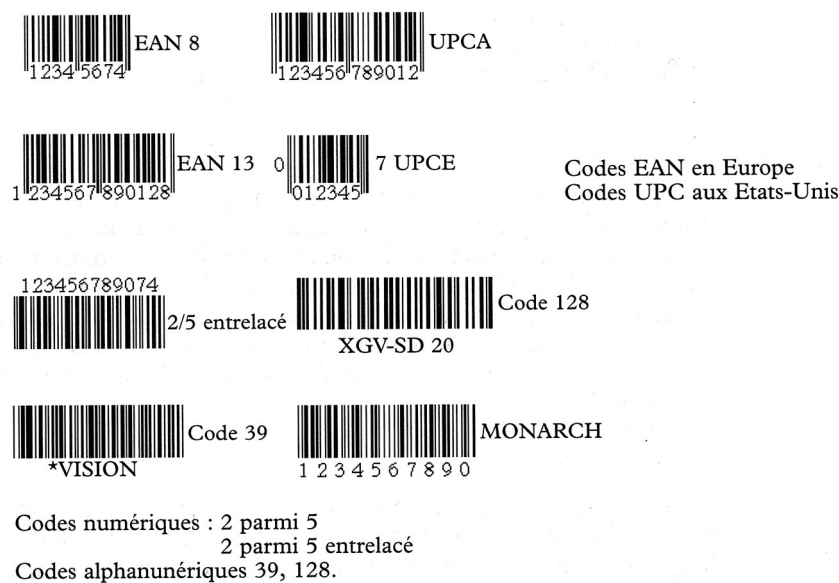


FIG. 1.15 – Quelques codes à barres

Le lecteur/décodeur a pour rôle de lire, de décoder puis de transmettre l'information d'une étiquette particulière vers une unité de traitement. Il fonctionne comme une caméra vidéo analysant une image électronique. Avantages : l'information portée par un code à barres est fiable (un code mal imprimé demeure lisible), facile à imprimer, économique, se limitant à une étiquette donc simple.

Chapitre 2

Les Automates Programmables Industriels

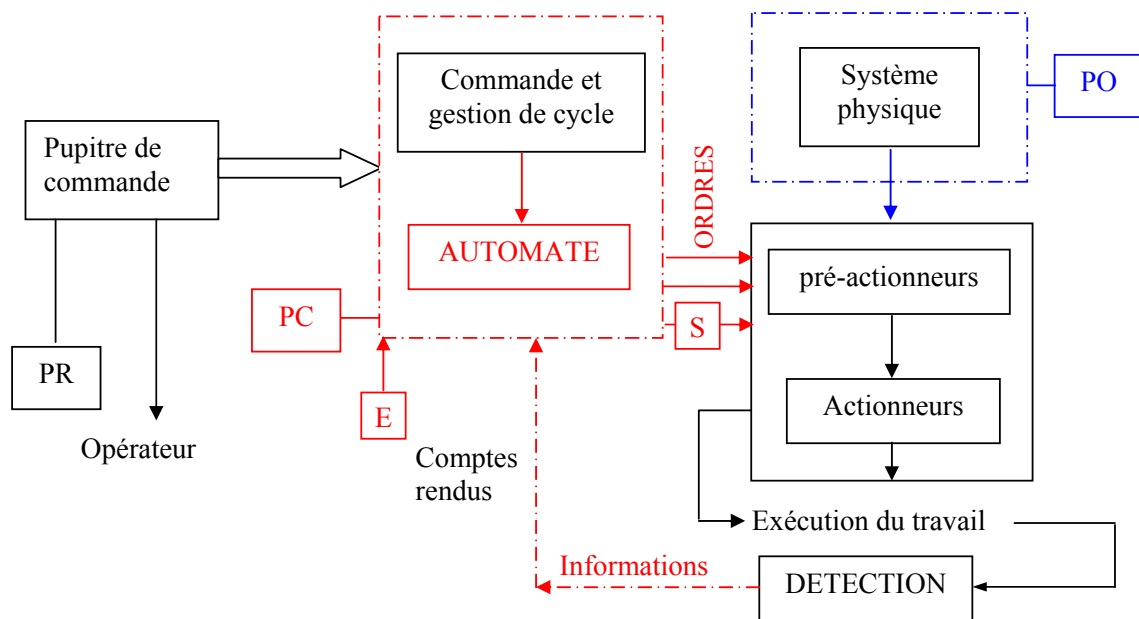


FIG. 2.1 – Situation de l'automate dans un système automatisé de production

2.1 La structure

Cet ensemble électronique gère et assure la commande d'un système automatisé. Il se compose de plusieurs parties et notamment d'une mémoire

programmable dans laquelle l'opérateur écrit, dans un langage propre à l'automate, des directives concernant le déroulement du processus à automatiser. Son rôle consiste donc à fournir des ordres à la partie opérative en vue d'exécuter un travail précis comme par exemple la sortie ou la rentrée d'une tige de vérin, l'ouverture ou la fermeture d'une vanne. La partie opérative lui donnera en retour des informations relatives à l'exécution du-dit travail.

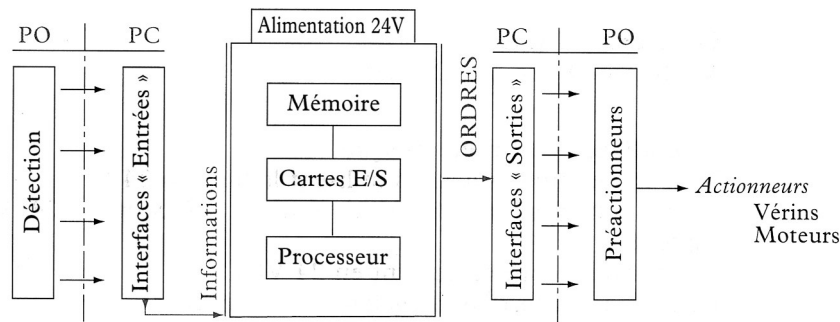


FIG. 2.2 – Structure interne d'un API

Les API comportent quatre parties principales :

- Une mémoire ;
- Un processeur ;
- Des interfaces d'Entrées/Sorties ;
- Une alimentation ($240 V_{ac} \rightarrow 24 V_{cc}$).

Ces quatre parties sont reliées entre elles par des bus (ensemble câblé autorisant le passage de l'information entre ces 4 secteurs de l'API). Ces quatre parties réunies forment un ensemble compact appelé automate. Elles sont détaillées dans l'article de M. Bertrand [16].

2.1.1 Description des éléments d'un API

Le processeur :

Son rôle consiste d'une part à organiser les différentes relations entre la zone mémoire et les interfaces d'E/S et d'autre part à gérer les instructions du programme.

Les interfaces :

L'interface d'Entrées comporte des adresses d'entrée, une pour chaque capteur relié. L'interface de Sorties comporte des adresses de sorties, une pour chaque pré-actionneur. Le nombre d'E/S varie suivant le type d'automate. Les cartes d'E/S ont une modularité de 8, 16 ou 32 voies. Elles admettent

ou délivrent des tensions continues 0 - 24 V_{cc} .

La mémoire :

Elle est conçue pour recevoir, gérer, stocker des informations issues des différents secteurs du système que sont le terminal de programmation (PC ou console) et le processeur, qui lui gère et exécute le programme. Elle reçoit également des informations en provenance des capteurs. Il existe dans les automates plusieurs types de mémoires qui remplissent des fonctions différentes :

- la conception et l'élaboration du programme font appel à la RAM et l'EEPROM ;
- la conservation du programme pendant l'exécution de celui-ci fait appel à une EPROM.

L'alimentation :

Tous les automates actuels utilisent un bloc d'alimentation alimenté en 240 V_{ac} et délivrant une tension de 24 V_{cc} .

2.1.2 Langages de programmation pour API

Chaque automate possède son propre langage. Mais par contre, les constructeurs proposent tous une interface logicielle répondant à la norme CEI¹ 1131-3. Cette norme définit cinq langages de programmation utilisables, [15] qui sont :

- GRAFCET ou SFC : ce langage de programmation de haut niveau permet la programmation aisée de tous les procédés séquentiels.
- Schéma par blocs ou FBD : ce langage permet de programmer graphiquement à l'aide de blocs, représentant des variables, des opérateurs ou des fonctions. Il permet de manipuler tous les types de variables.
- Schéma à relais ou LD : ce langage graphique est essentiellement dédié à la programmation d'équations booléennes (true/false).
- Texte structuré ou ST : ce langage est un langage textuel de haut niveau. Il permet la programmation de tout type d'algorithme plus ou moins complexe.

¹Commission Électrotechnique Internationale

- Liste d'instructions ou IL : ce langage textuel de bas niveau est un langage à une instruction par ligne. Il peut être comparé au langage assembleur.

Pour programmer l'automate, l'automaticien peut utiliser :

- une console de programmation ayant pour avantage la portabilité.
- un PC avec lequel la programmation est plus conviviale, communiquant avec l'automate par le biais d'une liaison série RS232 ou RS485 ou d'un réseau de terrain.

2.2 Conception d'un automatisme séquentiel

Pour un procédé donné, on doit tenir compte :

- des contraintes industrielles (rentabilité, sécurité),
- de la nature et des types de capteurs donc des signaux délivrés par les capteurs,
- de la nature des actionneurs donc des signaux de commande acceptés par les actionneurs.

Cette partie est la plus importante surtout pour la disposition des capteurs ! (Le capteur placé à cet endroit va-t-il délivrer une réponse pertinente ?). Elle détermine aussi :

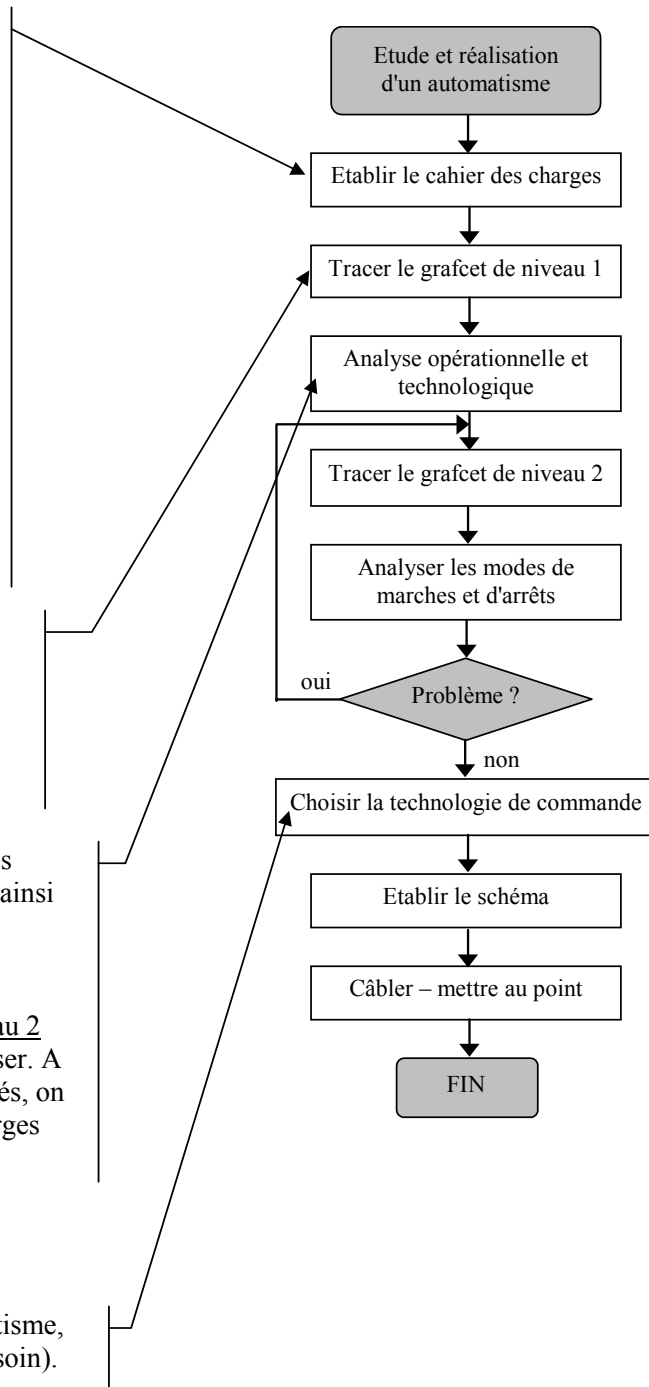
- le choix des interfaces d'E/S,
- le nombre et le type d'E/S,
- l'utilité ou pas d'un bus de terrain.

L'établissement des séquences de l'automatisme est ce que l'on appelle le grafcet de niveau 1. C'est le diagramme des actions à réaliser en fonction des informations reçues.

Cette analyse prend en compte les contraintes imposées par les capteurs et les actionneurs, ainsi que les modes de fonctionnement de l'automatisme (marche, arrêt, manuel, automatique, sécurité).

C'est ce qu'on appelle le GRAF CET de niveau 2 qui représente l'automatisme complet à réaliser. A ce stade, en fonction des problèmes rencontrés, on peut être obligé de revenir au cahier des charges et/ou au GRAFCET de niveau 1.

les différents matériels de l'automatisme, la technologie de commande (si besoin).



Chapitre 3

Le GRAFCET ou GRAPhe Fonctionnel de Commande Étape/Transition ou Sequential Function Chart

Norme internationale : [3] et [2], Norme française : [1]

”En 1975, le groupe de travail *Systèmes Logiques* de l’AFCET, décida, sous l’impulsion de Michel BLANCHARD[17] qui en était l’animateur, de créer une commission *Normalisation de la représentation du cahier des charges d’un automatisme logique*.

Les universitaires et industriels réunis dans ce groupe se sont alors attaqués à un problème ardu : tenter de définir un *formalisme* simple, accepté par tous, adapté à la représentation des évolutions séquentielles d’un système, intelligible à la fois par les concepteurs et les exploitants, et fournissant potentiellement des facilités de passage à une réalisation, à base matérielle et/ou logicielle de l’automatisme ainsi spécifié. Au début, le travail consista à dresser un état de l’art des différentes approches de modélisation du comportement de tels automatismes. Trois grandes classes d’outils de modélisation furent ainsi recensées :

- les organigrammes ;
- les représentations de systèmes logiques à évolutions simultanées (dont les Réseaux de Pétri) [8] ;
- les outils dérivés des graphes d’état.

L’analyse des avantages et inconvénients de ces outils mena, en 1977, à la définition du GRAFCET, ainsi nommé pour, à la fois marquer l’origine de ce nouvel outil de modélisation grAFCET et son identité GRAphe

Fonctionnel de Commande Etapes–Transitions). Les résultats de ces travaux firent l'objet, d'une part, d'un rapport interne à l'AFCECET, daté d'août 1977, et, d'autre part d'une publication officielle dans la revue "Automatique et Informatique Industrielle" en décembre 1977, date que la communauté considère aujourd'hui comme correspondant à la date de naissance effective du GRAFCET." extrait de "Généralisations Grafcet" de Pascal LHOSTE ¹, Animateur du Groupe "GRAFCET", Novembre 1997.

Le GRAFCET permet "...l'établissement des descriptions de la fonction et du comportement des systèmes de commandes en établissant une représentation graphique indépendante de la réalisation technologique..." (norme [3]).

Le GRAFCET est destiné à représenter des automatismes logiques (ou discrets), c'est à dire des systèmes à événements discrets dans lesquels les informations sont de type booléennes ou peuvent s'y ramener (par exemple la comparaison d'une température avec un seuil). Le GRAFCET est utilisé généralement pour spécifier et concevoir le comportement souhaité de la partie commande d'un système, mais il peut également être utilisé pour spécifier le comportement attendu de la partie opérative ou bien de tout le système de commande.

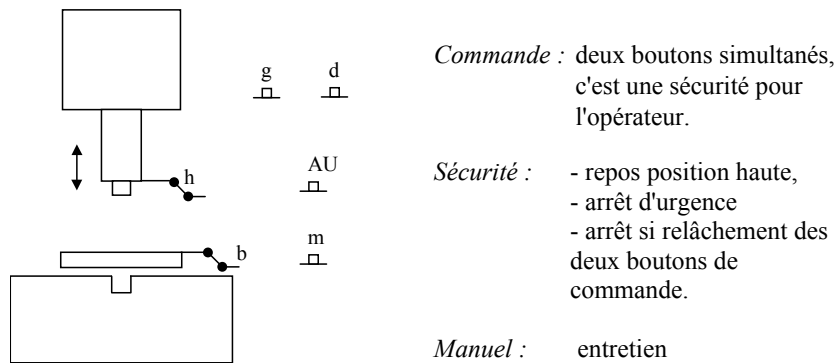
3.1 But du GRAFCET

Lorsque le mot GRAFCET (en lettre capitale) est utilisé, il fait référence à l'outil de MODÉLISATION. Lorsque le mot grafcet est écrit en minuscule, il fait alors référence à un modèle obtenu à l'aide des règles du GRAFCET. Présenté sous forme d'organigramme, son but est la description du fonctionnement de l'automatisme contrôlant le procédé. C'est tout d'abord un outil graphique, puissant, directement exploitable, car c'est aussi un langage pour la plupart des API existants sur le marché.

Il comprend :

- des étapes associées à des actions ;
- des transitions associées à des réceptivités ;
- des liaisons orientées reliant étapes et transitions.

¹http://www.lurpa.ens-cachan.fr/grafcet/groupe/gen_g7/geng7.html



Fonctionnement :

repos : position haute (et h à 1)

action sur g et d ?

descente poinçon

jusqu'à b ?

remontée poinçon

jusqu'à h ?

repos

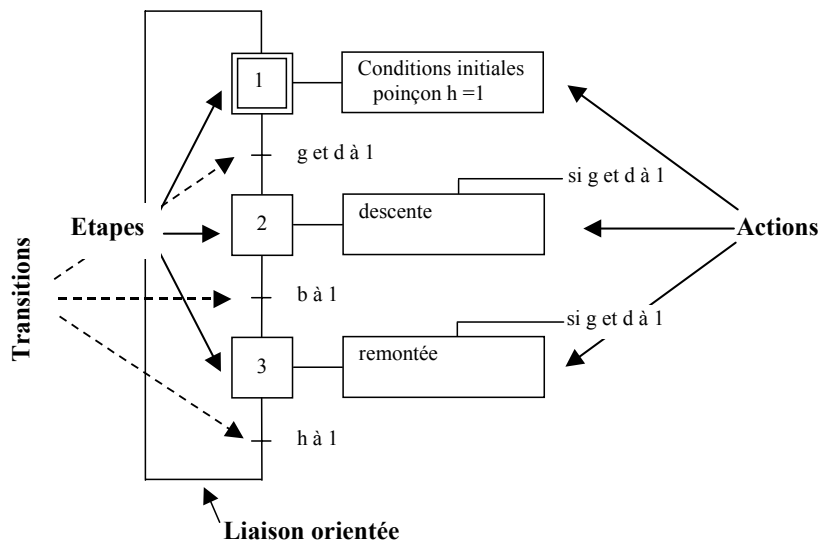


FIG. 3.1 – Exemple de la presse à emboutir

Attention ! Ce grafcet n'est pas complet, il manque l'arrêt d'urgence et la commande manuelle. Il faut noter que :

- suivant les capteurs, les conditions de transitions changent (ouvert ou fermé au repos) ;
- suivant les actionneurs, les actions sont différentes ;
- la sécurité opérateur est effectuée par une action, si la condition g et d à 1 est vraie.

3.2 Description du GRAFCET

3.2.1 Les étapes

L'étape symbolise un état ou une partie de l'état du système. L'étape possède deux états possibles : active représentée par un jeton dans l'étape ou inactive. L'étape i , repérée numériquement, possède ainsi une variable d'état, appelée variable d'étape X_i . Cette variable est une variable booléenne valant 1 si l'étape est active, 0 sinon.



Étape initiale

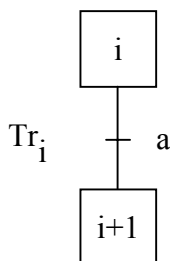
Elle représente le système à l'état de repos initial. Elle est activée au début du cycle.



Étape

A chaque étape est associée une action ou plusieurs, c'est à dire un ordre vers la partie opérative ou vers d'autres grafkets.

3.2.2 Les transitions



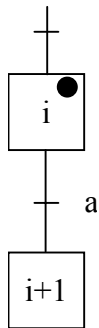
Une transition indique la possibilité d'évolution qui existe entre deux étapes et donc la succession de deux activités dans la partie opérative. Lors de son franchissement, elle va permettre l'évolution du système. A chaque transition est associée une réceptivité qui exprime la condition nécessaire pour passer d'une étape à une autre.

Cette condition est écrite sous forme d'une proposition logique, une fonction combinatoire calculée à partir :

- des variables d'entrées traduisant l'état des capteurs, des boutons poussoirs, etc.,
- du temps,
- de l'état courant des étapes du grafcet (les X_i).

Si la réceptivité n'est pas précisée, alors cela signifie qu'elle est toujours vraie.

Règle : si l'étape i est inactive, $X_i = 0$, la transition Tr_i est sans effet. Cependant, attention, valider sans raison une transition peut avoir des conséquences graves, perturbant le cycle dans certains cas !



Si l'étape i est active, $X_i = 1$, la transition a est validée, alors :

Si $a = 0$, alors attente

Si $a = 1$, alors l'étape i est dévalidée $X_i = 0$ et l'étape suivante $i+1$ est activée, $X_{i+1} = 1$.

3.2.3 Les liaisons orientées

Une liaison orientée est le lien qui lie une étape à une transition ou l'inverse. Par convention, étapes et transitions sont placées suivant un axe vertical. Les liaisons orientées sont de simples traits verticaux lorsque la liaison est orientée de haut en bas, et sont munis d'une flèche vers le haut lorsque la liaison est orientée vers le haut.

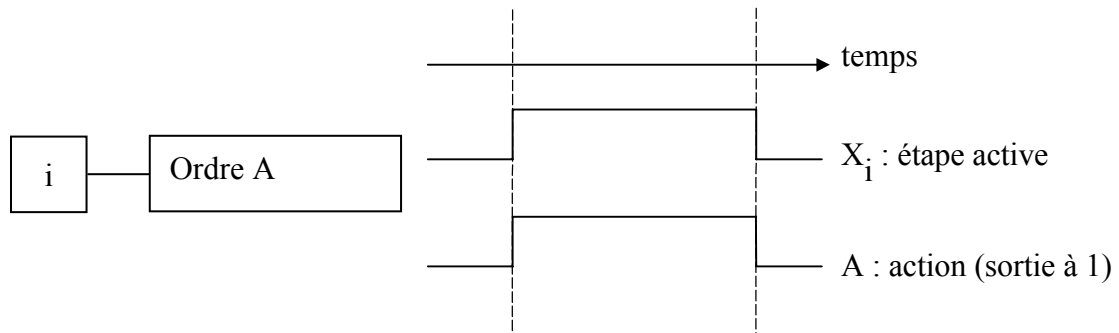
3.2.4 Notation des entrées/sorties

Lors de l'établissement du grafcet de spécification (grafcet de niveau 1), on utilise des noms explicites pour les entrées du système modélisé ainsi que pour les sorties. Lors du passage au grafcet de réalisation, on utilise plutôt des noms logiques : E_i pour les entrées et S_i pour les sorties.

3.2.5 Les actions

L'action associée à l'étape peut être de 3 types : continue, conditionnelle ou mémorisée.

Actions continues

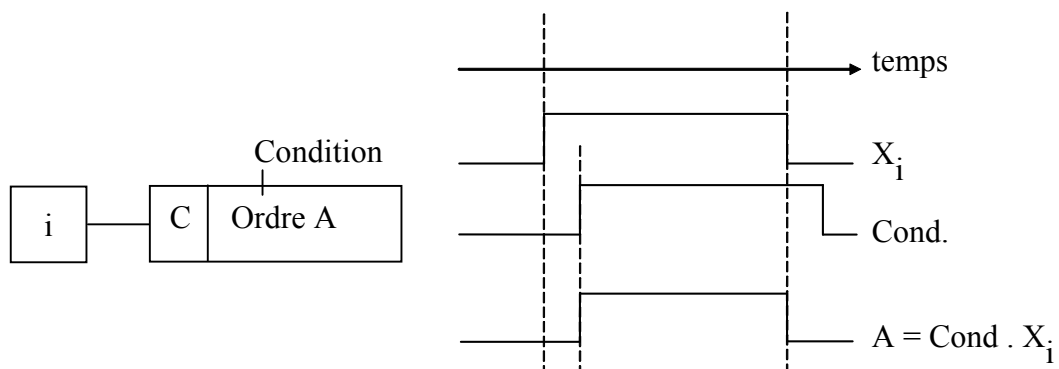


La ou les sorties correspondant à l'ordre A sont mises à 1 tant que l'étape associée est active. Lorsque l'étape devient inactive, la ou les sorties sont mises à 0.

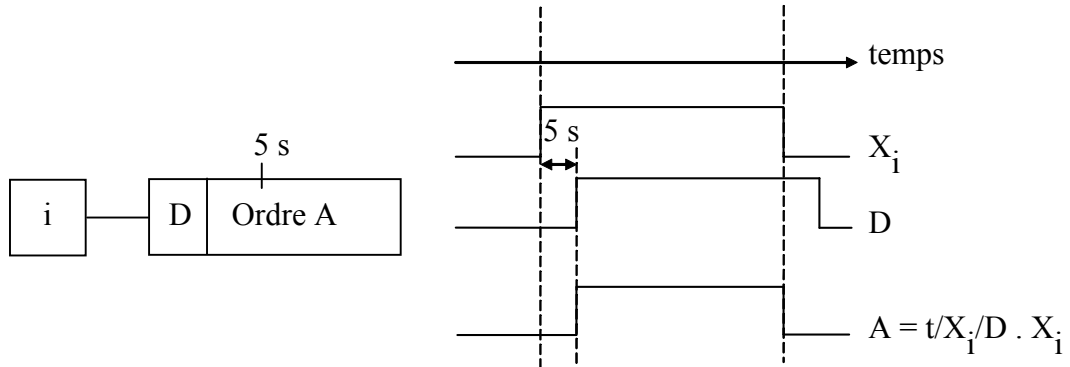
Actions conditionnelles

Une action conditionnelle n'est exécutée que si l'étape associée est active et si la condition associée est vraie. Elles peuvent être décomposées en 3 cas particuliers :

Action conditionnelle simple : Type C

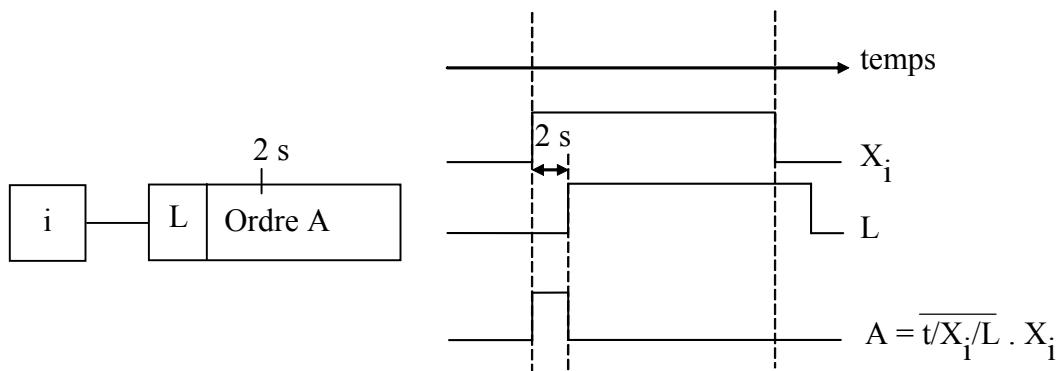


Action retardée : Type D (delay)

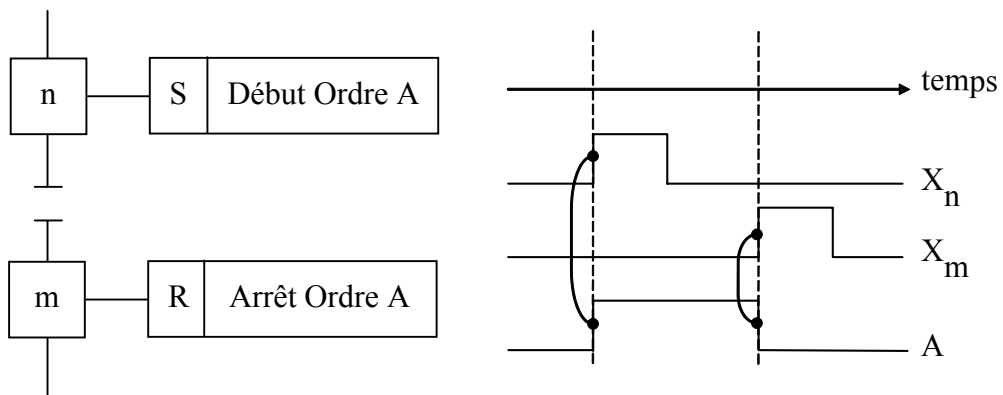


D est la durée de la temporisation, c'est à dire le retard mis pour passer à 1 après que X_i soit passé à 1.

Action limitée dans le temps : Type L (limited)



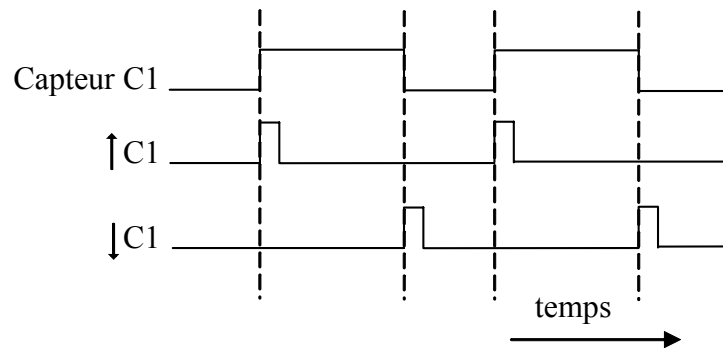
Actions mémorisées



On peut ainsi donner l'équation d'un ordre A en fonction des états des étapes, des conditions éventuelles et du temps.

3.2.6 Les réceptivités

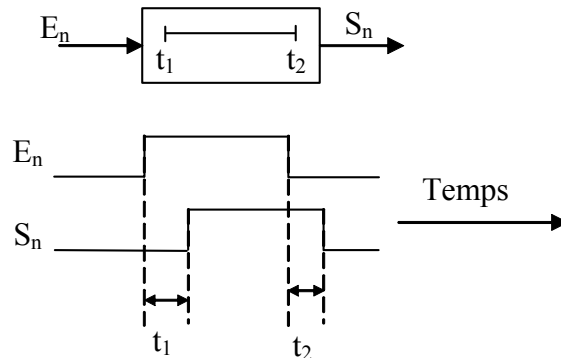
Une réceptivité est associée à chaque transition. C'est une fonction booléenne calculée à partir des entrées du graphe, des états des étapes ou des temporisations. Une réceptivité est donc écrite en utilisant les opérateurs ET, OU, NON et front. Le front montant ou descendant d'une variable permet de situer dans le temps le changement de valeur de ce capteur. On les note respectivement E et \bar{E} . Les fronts ne sont à 1 que durant un délai d , qui correspond au temps de prise en compte de l'événement, c'est à dire le temps de franchissement d'une transition.



3.2.7 Les temporisations

Les temporisations utilisées en GRAFCET font référence aux variables d'étapes (X_n). Une variable temporisation S_n est une variable booléenne dont le mode d'évaluation permet de prendre en compte le temps. Elle peut s'écrire de manière générale $S_n = t_1/X_n/t_2$ où :

- E_n désigne la variable d'entrée,
- t_1 désigne le retard apporté au changement de l'état logique $0 \rightarrow 1$ de la variable d'entrée E_n . Si $t_1 = 0$, alors la temporisation s'écrit $S_n = X_n/t_2$.
- t_2 désigne le retard apporté au changement de l'état logique $1 \rightarrow 0$ de la variable d'entrée E_n . Si $t_2 = 0$, alors la temporisation s'écrit $S_n = t_1/X_n$.



La variable X_n (l'étape n associée) doit être active pendant un temps supérieur ou égal à t_1 pour que S_n puisse être évaluée.

3.3 Les règles d'évolution du GRAFCET

Règle N°1 – Condition initiale

A l'instant initial, seules les étapes initiales sont actives.

Règle N°2 – Franchissement d'une transition

Pour qu'une transition soit validée, il faut que toutes ses étapes amont (immédiatement précédentes reliées à cette transition) soient actives. Le franchissement d'une transition se produit lorsque la transition est validée, ET seulement si la réceptivité associée est vraie.

Règle N°3 – Évolution des étapes actives

Le franchissement d'une transition entraîne obligatoirement dans cet ordre la désactivation de toutes ces étapes amont et l'activation de ses étapes aval.

Règle N°4 – Franchissement simultané

Toutes les transitions simultanément franchissables à un instant donné sont simultanément franchies.

Règle N°5 – Conflit d'activation

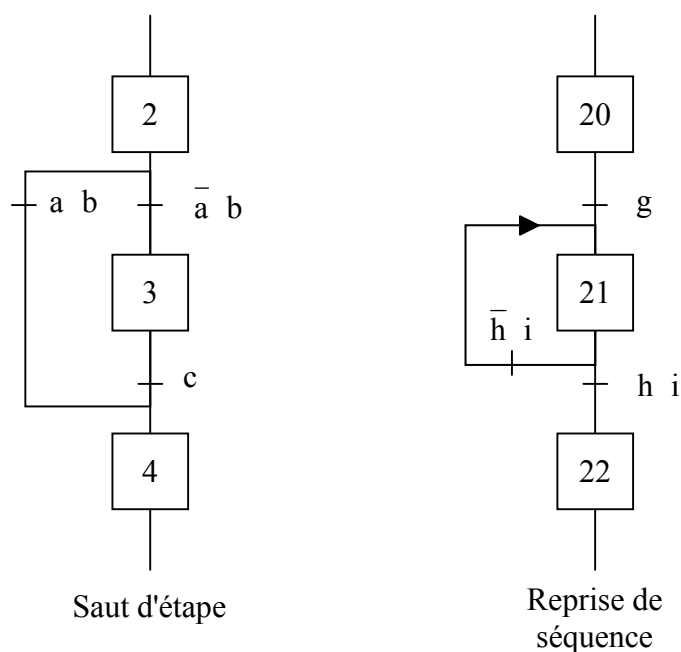
Si une étape doit être simultanément désactivée par le franchissement d'une transition aval, et activée par le franchissement d'une transition amont, alors elle reste active. On évite ainsi des commandes transitoires (néfastes au procédé) non désirées.

Durée de franchissement d'une transition ou d'activité d'une étape

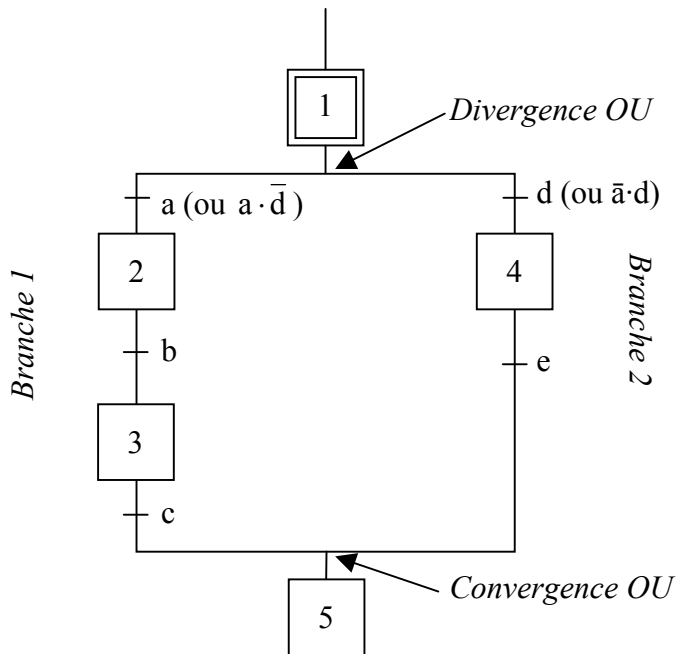
La durée de franchissement d'une transition est considérée théoriquement comme aussi petite que l'on veut, mais non nulle, même si, en pratique, cette durée peut être imposée par la technologie utilisée pour la réalisation de l'automatisme. De même, la durée d'activité d'une étape ne peut pas être nulle, mais suffisante, si besoin est, pour effectuer une action fugitive à la vitesse de la partie commande.

3.4 Les structures de base

3.4.1 Saut d'étape et reprise de séquence



3.4.2 Aiguillage entre deux ou plusieurs séquences



Soit X_1 active et $a = 0$ $d = 0$
 \rightarrow attente que a ou d passe à 1

Branche 1 :

- si $a = 1 \rightarrow X_2$
- si $b = 1 \rightarrow X_3$
- si $c = 1 \rightarrow X_5$

Branche 2 :

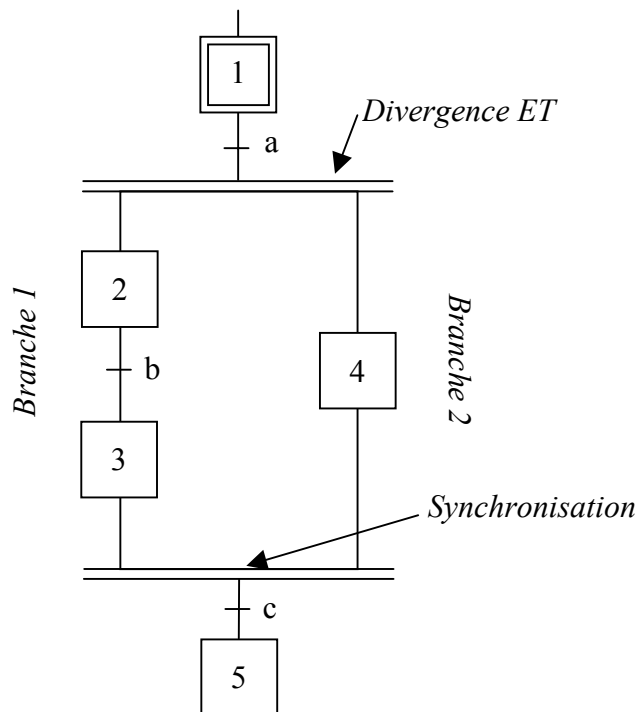
- si $d = 1 \rightarrow X_4$
- si $e = 1 \rightarrow X_5$

Remarque : chaque branche de la divergence peut être terminée par un saut, auquel cas il n'y a pas de convergence en OU.

Attention !!! Si a et d sont à 1 simultanément, alors nous sommes confronté à un cas de parallélisme interprété. Les étapes 2 et 4 vont devenir actives simultanément et donc provoquer l'exécution d'actions en parallèle, situation non voulue par le concepteur. a et d doivent être des conditions exclusives : l'une ou l'autre sera vraie, mais pas les deux à la fois. On résout ce problème en mettant, par exemple, comme condition sur la branche 1.

3.4.3 Parallélisme

Au contraire de l'aiguillage où ne peut se dérouler qu'une seule activité à la fois, il peut y avoir plusieurs activités indépendantes pouvant se dérouler en parallèle. On parle alors de parallélisme structurel.



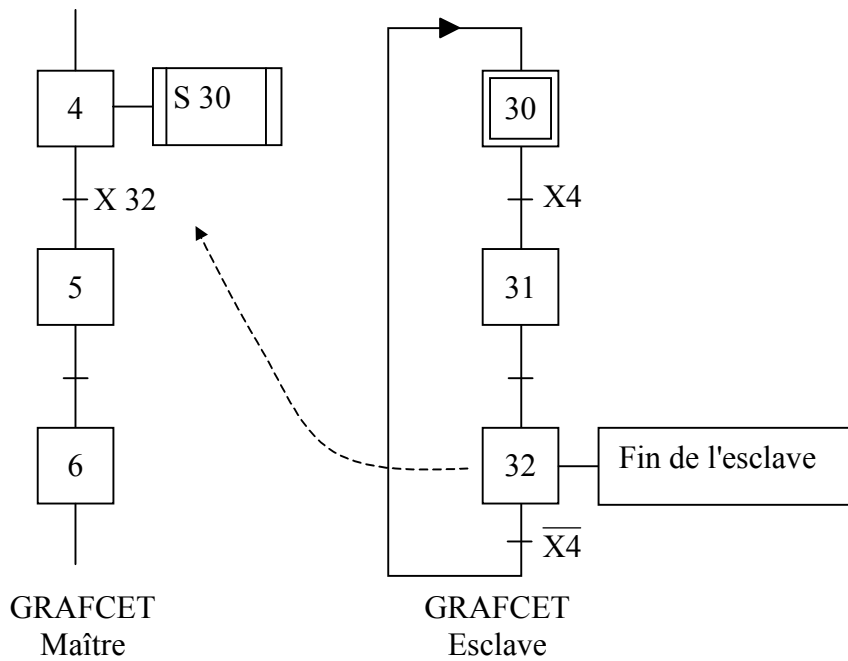
Après le franchissement de a, nous obtenons ainsi deux activités simultanées ou parallèles.

La synchronisation permet d'attendre la fin de plusieurs activités se déroulant en parallèle, pour continuer par une seule. On parle alors de synchronisation puisque, l'activité de l'étape 4 qui se termine avant celle de 3 doit attendre et se synchroniser avec elle.

Ainsi, pour la compréhension du GRAFCET, une étape dite de synchronisation peut être rajoutée dans la branche la plus rapide de la divergence.

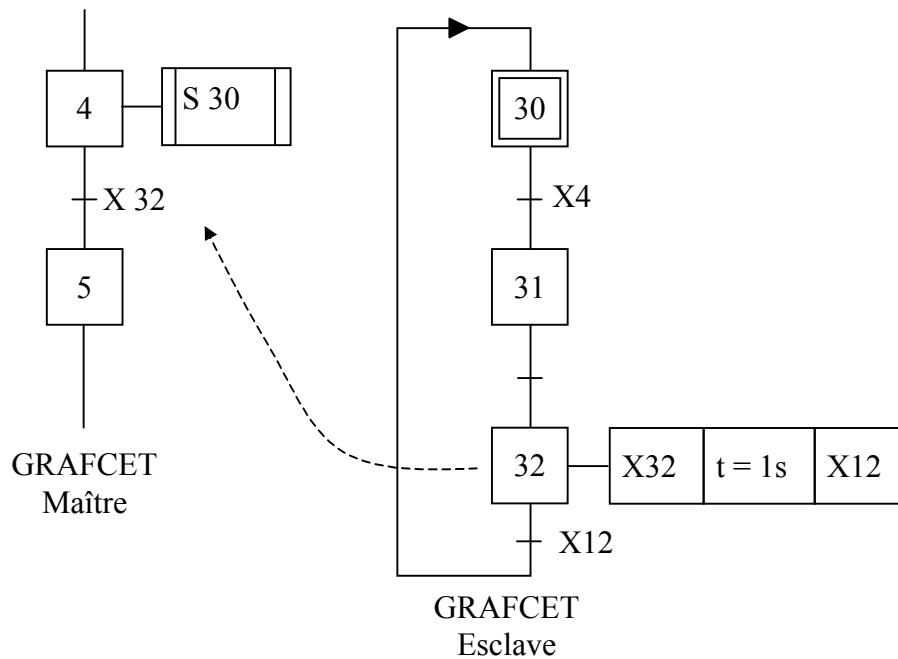
3.4.4 Sous-grafcet ou séquence répétée

Dans un système automatisé, certaines séquences peuvent revenir de façon répétitive dans le cycle. Pour éviter de répéter ces mêmes actions, il est possible d'utiliser un sous-programme. Celui-ci est écrit sous la forme d'un grafcet indépendant, connecté au grafcet principal.



Au départ, les étapes initiales des deux grafjets sont activées. Dans le maître, l'activation de l'étape 4, $X_4 = 1$, va activer le déroulement du grafjet esclave en validant la première transition, et provoquer le déclenchement de la réceptivité X_{32} . Dans l'esclave, l'activation de l'étape 32, $X_{32} = 1$, provoque l'apparition dans le maître de la réceptivité X_{32} qui autorise le déroulement du programme principal, en dévalidant X_4 . Cette étape 32 "rend la main au maître".

Il existe une autre méthode pour rendre la main au grafjet maître, qui d'ailleurs est la plus utilisée. Elle est illustrée ci-dessous et utilise une temporisation pour maintenir la variable X_{32} à 1 pendant 1 s afin de rendre la main au maître.



Les grafjets esclaves sont très utiles pour les séquences répétées, arrêt d'urgence, commande manuelle. Ils permettent de commander plusieurs automatismes sur un même API.

3.4.5 Gestion de conflits

Suivant la structure de programmation choisie, et pour des automatismes donnés, il existe des situations conflictuelles qui peuvent avoir des conséquences graves. Comment y remédier ?

Aiguillage

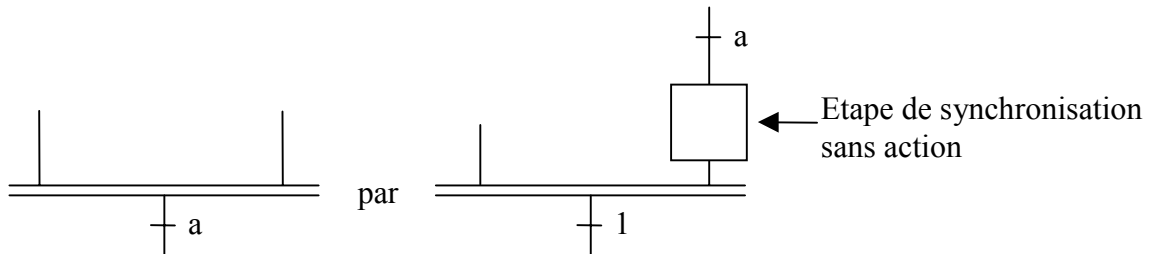
Si les conditions d'aiguillage sont toutes validées simultanément, l'automatisme est face à un problème de préséance, qui se résout en doublant l'une des conditions d'aiguillage (cf. 3.4.2).

Parallélisme où 3 types de conflits peuvent se présenter

- le déroulement simultané de séquences avec attente commune. Ceci se produit, si les durées des actions des séquences simultanées sont différentes ! On ajoute alors une étape de synchronisation sans action. C'est ce que l'on appelle la synchronisation de durée.

- l'état d'avancement de plusieurs séquences simultanées. Une séquence a son déroulement tributaire de l'état d'avancement d'une autre séquence. L'utilisateur a alors besoin d'une synchronisation de décalage temporel qui peut se faire au niveau d'une transition ou d'un grafcet esclave activé en fonction des contraintes de synchronisation.
- la mémorisation de disponibilité d'une partie de la machine. Une séquence particulière n'est utilisée qu'alternativement par plusieurs séquences. Il faut alors faire un choix de priorité ou utiliser des étapes d'attente si la séquence commune est déjà utilisée.

Remarque : sur certains API, la convergence en ET est suivie d'une transition qui est toujours validée à 1. Pour résoudre cela, on remplace donc :



3.5 GRAFCET de niveau 1 et 2

3.5.1 GRAFCET de niveau 1 du point de vue de la partie opérative

C'est en général la description de l'automatisme seul, c'est à dire l'enchaînement des actions et des transitions permettant de contrôler le procédé. Lorsque l'on aborde l'analyse et la description d'un système, on ne sait pas quelle technologie sera retenue pour les actionneurs, les capteurs et la commande. On décrira dans ce GRAFCET les actions et les événements en termes généraux.

3.5.2 GRAFCET de niveau 2 du point de vue de la partie commande

C'est la description complète de l'automatisme qui tient compte de toutes les contraintes du procédé. Les points essentiels du GRAFCET de niveau 2 sont :

- la simplification du GRAFCET niveau 1, c'est à dire les parallélismes et les séquences répétées ;
- les modes de fonctionnement de l'automatisme, c'est à dire la prise en compte de la marche automatique, de la marche par cycle et de la marche manuelle utile pour la maintenance et les réglages ;
- les arrêts d'urgence entraînant la coupure d'alimentation, l'utilisation d'une variable AU (si arrêt d'urgence $AU = 1$) à rajouter dans les conditions logiques pour les différentes actions, une procédure de dégagement si $AU = 0$;
- les sécurités opérateur, c'est à dire la mise en place de protections, de commandes à deux mains, qui doivent être activées pour le départ du cycle et pendant le cycle ;
- les sécurités procédé, c'est à dire l'arrêt de l'automatisme si une condition anormale est détectée, par exemple la détection d'un objet incorrect ou mal positionné ;
- les conditions initiales, c'est à dire les différentes positions au repos (au départ du cycle), l'état de l'alimentation et l'ensemble des tests destinés à vérifier les conditions initiales.

La complexité de cette tâche peut être grande ! Le GRAFCET de niveau 2 doit être étudié avec le plus grand soin. Ceci peut éventuellement conduire à modifier le procédé, et à modifier si besoin les capteurs et les actionneurs !

Dans certains cas, on peut être amené (après étude) à rechercher d'autres solutions que l'automatisme séquentiel, par exemple la logique floue. Ces cas limites sont :

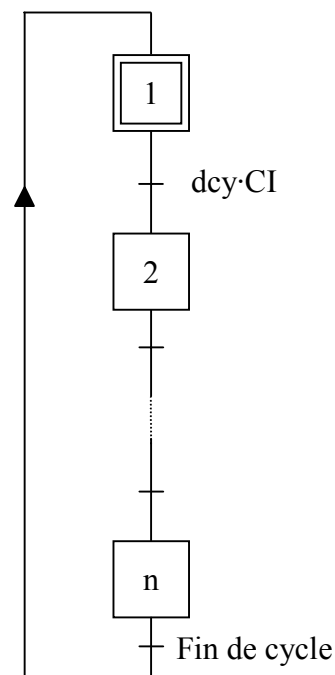
- une indétermination logique au niveau du procédé ;
- une description logique inadaptée ;
- une durée de cycle de scrutation ou d'acquisition trop longue ;
- la difficulté à définir une séquence de dégagement pour l'arrêt d'urgence.

Attention!!! En milieu industriel, les essais erreurs peuvent être dangereux ou catastrophiques : d'où l'intérêt de faire une étude du grafcet de niveau 2 très approfondie!!!

3.6 Modes de marches et d'arrêts

On considère un ensemble partie opérative – partie commande autonome, avec éventuellement une partie opérative défaillante, mais une partie commande toujours en ordre de marche.

3.6.1 Cycle unique ou cycle par cycle

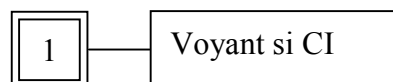


Voilà le grafcet type pour un cycle unique, avec comme condition de validation : $dcy \cdot CI$

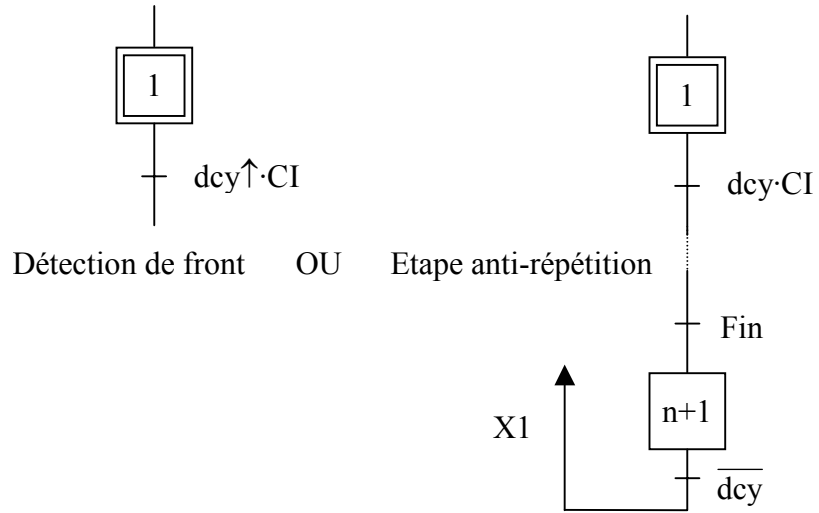
- dcy étant l'ordre de départ donné par l'opérateur ;
- CI la variable représentant les conditions initiales.

A ce grafcet, on peut rajouter :

- Un voyant de contrôle activé lors de l'étape d'initialisation ;

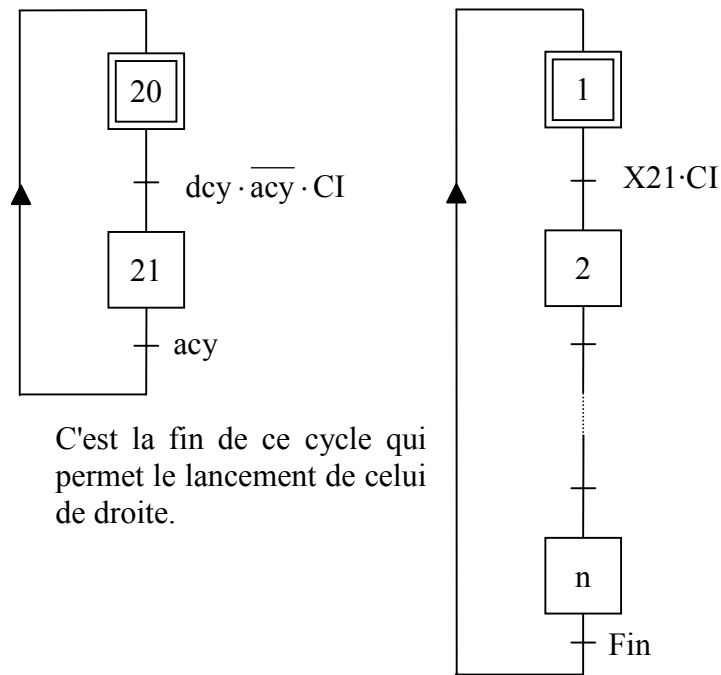


- Un test anti-répétition du cycle.

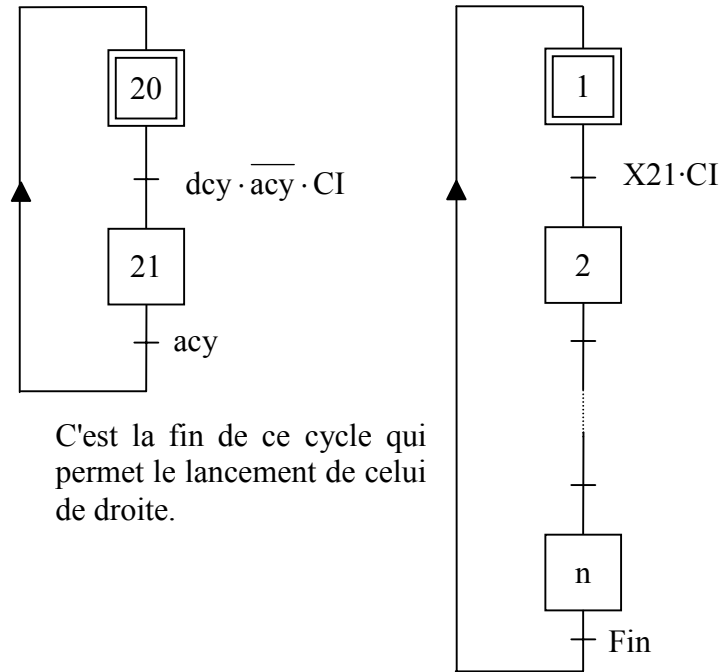


3.6.2 Cycle par cycle ou continu

Il existe deux solutions pour réaliser ce type de grafcet. Ces deux solutions font appel aux deux variables associées au cycle : dcy pour le départ du cycle, et acy pour l'arrêt du cycle.



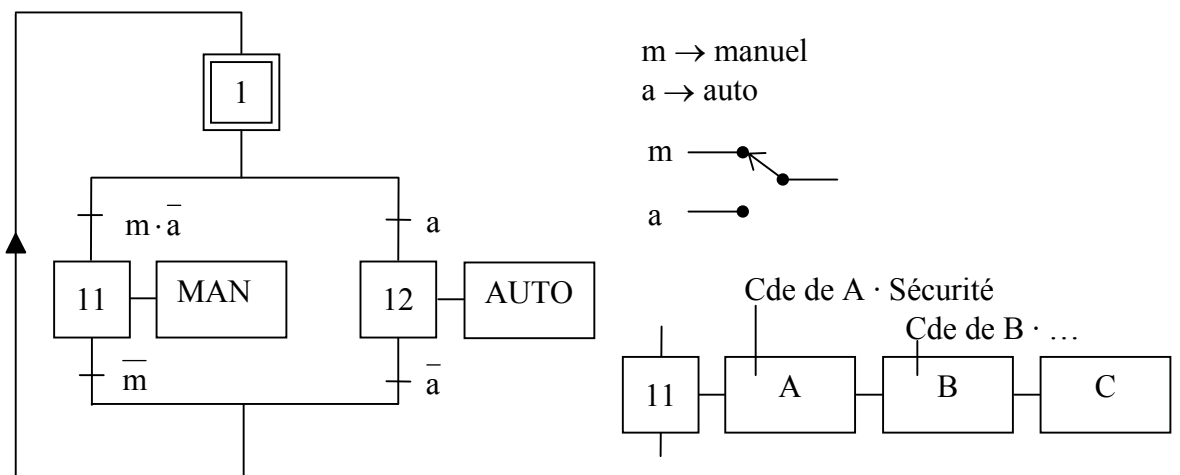
Solution N°1



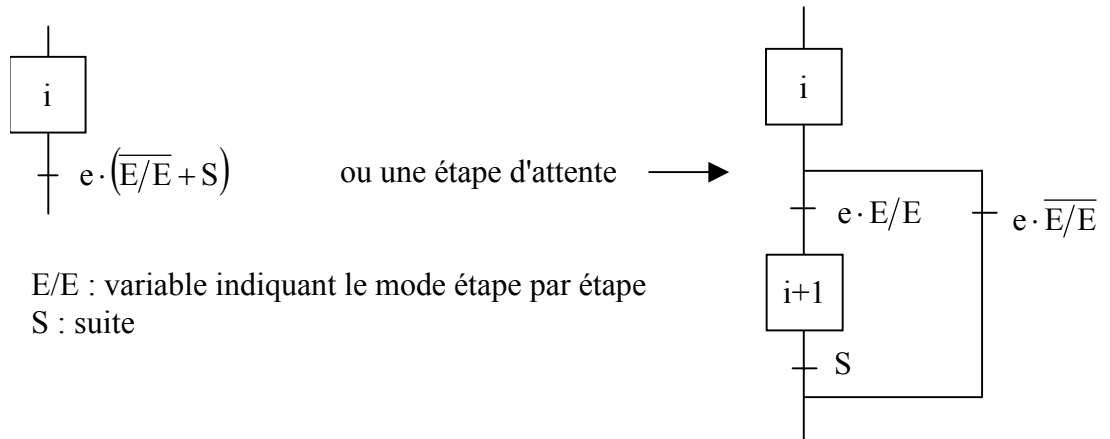
Solution N°2 : un interrupteur permet de choisir entre le mode automatique ou celui cycle par cycle.

3.6.3 Intervention réglages et dégagement

– Marche manuelle



- Marche séquence par séquence ou étape par étape

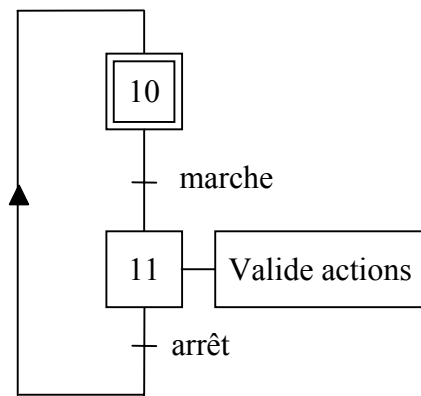


On ajoute, si possible, une transition permettant soit de sélectionner une séquence, soit de s'arrêter sur une étape donnée. Ceci est très difficile à mettre en œuvre.

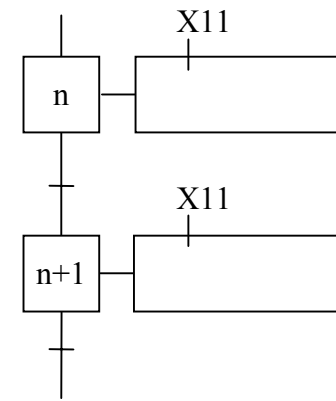
Attention : tous ces modes présentent de grand danger matériel et opérateur !

3.6.4 Arrêts

- Arrêt momentané

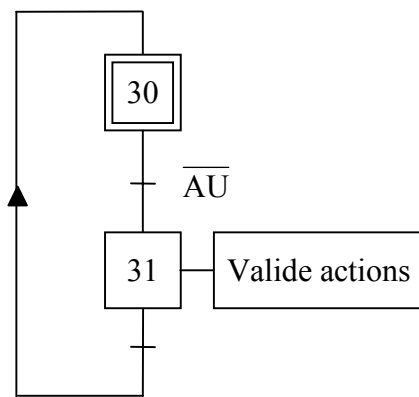


grafcet esclave :
arrêt momentané

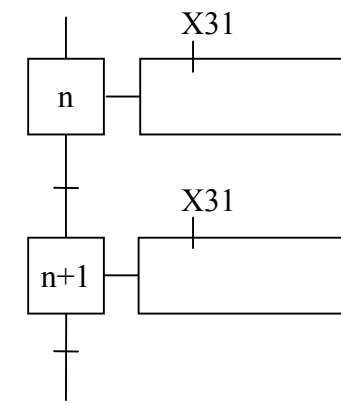


grafcet maître

- Arrêt d'urgence

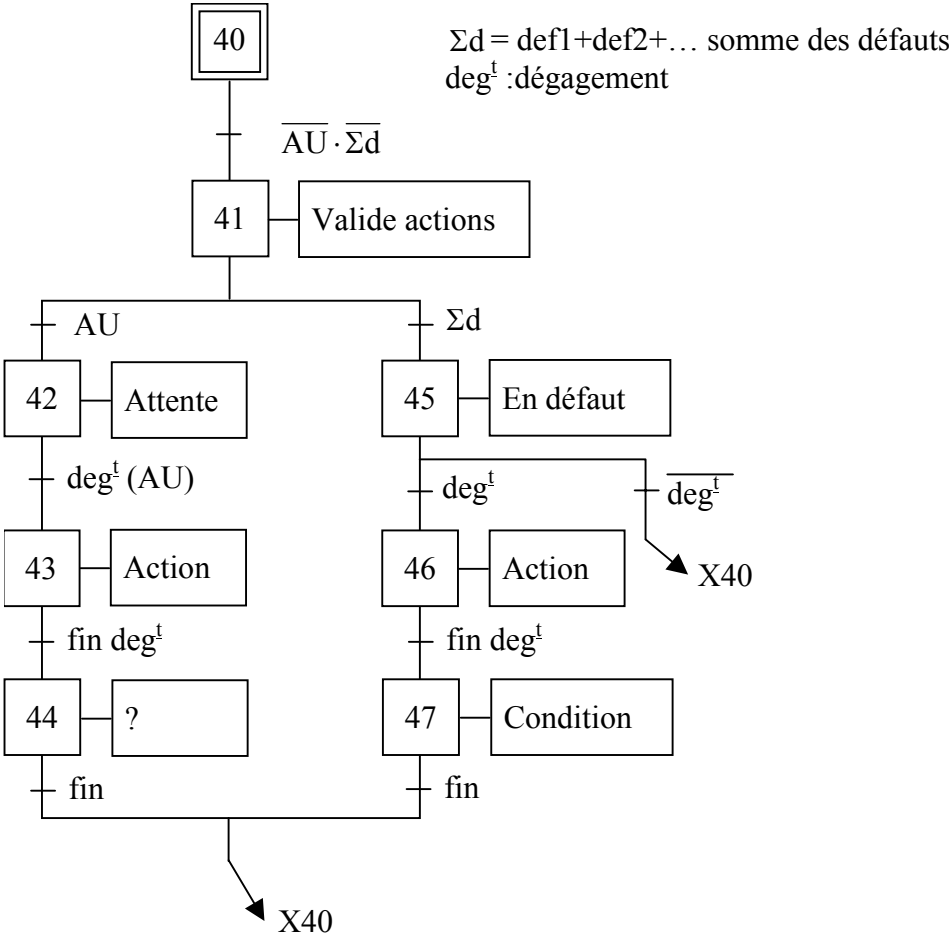


grafcet esclave :
arrêt d'urgence



grafcet maître

- Arrêt d'urgence + défauts + dégagement



Glossaire

API : Automates Programmables Industriels

EPROM : Erasable Programmable Read-Only Memory, ROM qui peut être effacée (sous UV) et reprogrammée.

EEPROM : Electrically Erasable Programmable Read Only Memory, ROM qui peut être effacée par un signal électrique et reprogrammée.

FBD : Function Block Diagram, en français schéma par blocs, langage de programmation d'automates.

GEMMA : Guide d'Etudes des Modes de Marches et d'Arrêts

GRAFCET : GRAPhe Fonctionnel de Commande Etape/Transition – langage de programmation d'automates.

ILS : Interrupteur à Lame Souple

IL : Instruction List, en français liste d'instructions, langage de programmation d'automates.

LD : Ladder Diagram, en français schéma à relais, langage de programmation d'automates.

PC : Partie Commande

PO : Partie Opérative

PR : Partie Relation

RAM : Random Access Memory

ROM : Read Only Memory

SAP : Système Automatisé de Production

SCC : Système Contrôle/Commande

SFC : Sequential Function Chart, en français GRAFCET, langage de programmation d'automates.

ST : Structured Text, en français texte structuré, langage de programmation d'automates.

TOR : Tout Ou Rien

Bibliographie

- [1] NF C 03-190. *Diagramme fonctionnel GRAFCET pour la description des systèmes logiques de commande*. Union Technique de l'Électricité, 1995.
- [2] Norme CEI 1131-3. *Automates programmables - Partie 3 : Langages de programmation*. Commission électrotechnique internationale, 1993.
- [3] Norme CEI 848. *Etablissement des diagrammes fonctionnels pour systèmes de commande*. Commission électrotechnique internationale, 1988.
- [4] BODART A., CARRE-MENETRIER V., DE LOOR P., J.B. Deluche, J. Dupont, D. Gendreau, J. Hancq, A. Kril, and J. Nido. *7 facettes du GRAFCET – Approches pratiques de la conception à l'exploitation*. Cepadues éditions, 2000.
- [5] ADEPA/AFCEC. *Le GRAFCET : ouvrage de référence*. Collection Automatisation et Production, Cepadues Editions, 1991.
- [6] P. BRARD and G. COLOMBARI. Outil de description des automatismes séquentiels : Le GRAFCET. *Techniques de l'Ingénieur*, R 7250, 30 pages, 1995.
- [7] Cours d'AUTOMATIQUE de D. CHAPPE à l'ENSMM (Besançon)
. http://www.ens2m.fr/cours_automatique/index.html.
- [8] R. David and H. Alla. *Du Grafcet aux réseaux de Pétri, 2^e édition*. Série Automatique, Hermes, 1997.
- [9] Cours de GRAFCET de V. Chapurlat au Mines d'Alès
. <http://www.lgi2p.ema.fr/%7Echapurla/enseignements.html>.
- [10] Exemples de grafkets sur le site LIMI (Brest)
. <http://doelan-gw.univ-brest.fr:8080/grafcet/index.html>.
- [11] Cours GRAFCET de P. Trau / ULP IPST / mars97
. <http://www-ipst.u-strasbg.fr/pat/autom/grafcet.htm>.
- [12] G. DÉCHENAUX. API et PC : solutions concurrentes ou complémentaires? *Techniques de l'Ingénieur*, R 8022, 11 pages, 1998.

- [13] Serveur du Groupe de recherche GRAFCET de l'AFCEC (comporte énormément de liens)
. [http ://www.lurpa.ens-cachan.fr/grafcet/grafcet_fr.html](http://www.lurpa.ens-cachan.fr/grafcet/grafcet_fr.html).
- [14] D. DUPONT and D. DUBOIS. Réalisation technologique du GRAFCET. *Techniques de l'Ingénieur*, S 8032, 24 pages, 2002.
- [15] P. JARGOT. Langage de programmation pour API. Norme CIE 1131-3. *Techniques de l'Ingénieur*, S 8030, 23 pages, 1999.
- [16] BERTRAND M. Automates programmables industriels. *Techniques de l'Ingénieur*, S 8015, 14 pages, 2001.
- [17] BLANCHARD M. *Comprendre, maîtriser et appliquer le GRAFCET*. Collection Automatisation et Production, Cepadues éditions, 1979.
- [18] S. MORENO and E. PEULOT. *LE GEMMA : Modes de marches et d'arrêts, Grafcet de coordination de tâches, Conception des systèmes automatisés de production sûrs*. Editions Casteilla, 1997.
- [19] J. PERRIN, M. MAGNIEZ, S. SINIBALDI, A. CORTIAL, S. BADEAU, F. FRONTERI, and F. AGATE. *Automatique Informatique Industrielle*. Collection H. LONGEOT – L. JOURDAN, Dunod, 1989.
- [20] C. POURCEL. *Systèmes Automatisés de Production*. Collection Automatisation et Production, Cepadues éditions, 1986.