

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
CENTRE UNIVERSITAIRE AHMED ZABANA DE RELIZANE



INSTITUT DE SCIENCES ET TECHNOLOGIES
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE

SUPPORT PÉDAGOGIQUE

Cours ; Travaux Dirigés (TDs) et Travaux Pratiques (TPs)

Rédigé par :

Dr. MILOUDI Mohamed

Maitre de Conférences A

mohamed.miloudi@cu-relizane.dz

Intitulé du Polycopié :

**A U T O M A T E S
P R O G R A M M A B L E S
I N D U S T R I E L S**

Niveau : 3^{ème} Année Licence en Génie Électrique

Option : Automatique



"إِنِّي رَأَيْتُ أَنَّهُ لَا يَكْتَبُ إِنْسَانٌ كِتَابًا فِي يَوْمِهِ؛ إِلَّا قَالَ فِي عَدِّهِ

لَوْ غَيَّرَ هَذَا كَانَ أَحْسَنَ، وَلَوْ زِيدَ كَذَا كَانَ يُسْتَحْسَنُ

وَلَوْ قَدَّمَ هَذَا كَانَ أَفْضَلَ، وَلَوْ تَرَكَ هَذَا كَانَ أَجْمَلَ

هَذَا مِنْ أَعْظَمِ الْعِبَرِ، وَهُوَ دَلِيلٌ عَلَى اسْتِيْلَاءِ الْمُتَقْصِ عَلَى جُمْلَةِ الْبَشَرِ"

القاضي الفاضل، أحد الأئمة الكتاب، ووزير السلطان صلاح الدين الأيوبي



Ce manuscrit a été réalisé pour servir comme support de cours aux étudiants de la troisième année Licence, option : Automatique du département de Génie Électrique de l'Université Ahmed Zabana de Relizane. Il comporte la philosophie de l'automatisme et l'informatique industrielle. Il peut également être utile à toute personne appelée à l'étude de l'automatisme industriel en génie électrique.

Son élaboration est le fruit d'un travail large de recueil, de lecture et de synthèse de diverses sources documentaires : livres et manuscrits de cours. L'auteur tient à mentionner que la plupart des figures et illustrations présentes dans ce cours sont à la propriété des détenteurs du droit d'auteur et que leur reproduction dans ce cours est uniquement pour une utilisation pédagogique.

Ce guide recense l'analyser d'un système automatisé de production dans le monde de génie électrique avec des travaux dirigés avec des exemples industriels, des travaux pratiques en automatismes et de références bibliographiques auxquelles l'étudiant doit se conformer pour des études approfondies dans cet axe de recherche. Il s'adresse également aux personnes voulant s'initier aux techniques d'automatisation de processus industriels.

TABLE DES MATIÈRES

TABLE DES MATIÈRES

INTRODUCTION GÉNÉRALE	01
------------------------------	----

CHAPITRE I GÉNÉRALITÉS SUR LES SYSTÈMES AUTOMATISÉS

I.1	INTRODUCTION	05
I.2	SYSTÈME	05
I.3	SYSTÈME DE PRODUCTION	06
I.4	SYSTÈME DE PRODUCTION AUTOMATISÉ (SAP)	06
I.4.1	Matières d'Œuvre	07
I.4.2	Valeur Ajoutée	08
I.4.3	Contexte	09
I.4.4	Objectifs de l'Automatisation	09
I.4.5	Conduite et Surveillance d'un Système Automatisé	10
I.5	MODÉLISATION D'UN SYSTÈME AUTOMATISÉ	11
I.6	CONVENTION ASSOCIÉE AU SYSTÈME AUTOMATISÉ	12
I.7	STRUCTURE DE BASE D'UN SYSTÈME AUTOMATISÉ	12
I.7.1	Partie Opérative (P.O)	13
I.7.2	Partie Commande (P.C)	15
I.7.3	Partie Relation (P.R)	16
I.7.4	Relations entre PC, PO et PR	18
I.8	STRUCTURE MATÉRIELLE DE LA PARTIE OPÉRATIVE	19
I.8.1	Pré-actionneurs	19
I.8.2	Actionneurs	23
I.8.3	Effecteurs	31
I.8.4	Capteurs	31
I.9	CONCLUSION	37

CHAPITRE II RÉSEAUX DE PÉTRI (RdP)

II.1	INTRODUCTION	38
II.2	CONCEPTS DE BASE DES RÉSEAUX DE PÉTRI	39
II.2.1	Notions de Base	39
II.2.2	Vocabulaire Associé au Réseau de Pétri	40

II.2.3	Réseau de Pétri	41
II.2.4	Marquage d'un Réseau de Pétri	43
II.3	ÉVOLUTION DES RÉSEAUX DE PÉTRI	46
II.3.1	Validation d'une Transition	46
II.3.2	Franchissement d'une Transition	46
II.3.3	Franchissement d'une Transition Source	47
II.3.4	Franchissement d'une Transition Puit	48
II.3.5	Franchissement d'une Transition avec Plusieurs Ressources	48
II.4	MARQUAGES ACCESSIBLES	52
II.5	SÉQUENCE DE FRANCHISSEMENT	56
II.6	PROPRIÉTÉS DES RÉSEAUX DE PÉTRI	58
II.6.1	Réseau de Pétri avec et sans Conflit	58
II.6.2	Réseau de Pétri à Choix Libre	59
II.6.3	RdP en Conflit Relatif au Marquage ou Transitions en Conflit Effectif	60
II.6.4	Réseau de Pétri Pur	60
II.6.5	Réseau de Pétri Propre	63
II.6.6	Réseau de Pétri Vivant	63
II.6.7	Réseau de Pétri Borné	65
II.6.8	Réseau de Pétri Conforme	65
II.6.9	Réseau de Pétri Généralisé	65
II.6.10	Réseau de Pétri Autonome et non Autonome	66
II.6.11	Réseau de Pétri à Capacités	67
II.6.12	Réseau de Pétri Interprété	69
II.7	STRUCTURES PARTICULIÈRES DES RÉSEAUX DE PÉTRI	69
II.7.1	Parallélisme	70
II.7.2	Synchronisation	70
II.7.3	Partage de Ressources	72
II.8	RÉSEAUX DE PÉTRI TEMPORISÉS	72
II.8.1	RdP P-Temporisé	73
II.8.2	RdP T-Temporisé	74
II.9	CONCLUSION	75

CHAPITRE III

GRAF CET OU SFC

III.1	INTRODUCTION	76
III.2	DÉFINITION DU GRAFCET	77
III.3	BUT DU GRAFCET	77
III.4	DOMAINE D'APPLICATION	78
III.5	NOTIONS DU POINT DE VUE SAP/PO/PC	78

III.5.1	GRAFCET Point de Vue Système Automatisé de Production (SAP)	79
III.5.2	GRAFCET Point de Vue Partie Opérative (PO)	79
III.5.3	GRAFCET Point de Vue Partie Commande (PC)	80
III.6	STRUCTURE GRAPHIQUE DU GRAFCET	80
III.7	ÉLÉMENTS DE BASE DU GRAFCET	81
III.7.1	Étape	81
III.7.2	Action	81
III.7.3	Transition	82
III.7.4	Réceptivité	82
III.7.5	Liaisons Orientée	82
III.8	CLASSIFICATION DES ACTIONS ASSOCIÉES AUX ÉTAPES	82
III.8.1	Action Continue	83
III.8.2	Action Conditionnelle	83
III.8.3	Action Maintenue	84
III.8.4	Action Méorisée	84
III.8.5	Action Manuelle	85
III.9	CLASSIFICATION DES RÉCEPTIVITÉS	85
III.9.1	Variables	85
III.9.2	Événements	87
III.10	STRUCTURE DE BASE DU GRAFCET	87
III.10.1	GRAFCET à Séquence Unique	87
III.10.2	GRAFCET à Sélection de Séquence	89
III.10.3	GRAFCET à Saut d'Étapes	90
III.10.4	GRAFCET à Reprise d'Étapes	91
III.10.5	GRAFCET à Exclusivité	92
III.10.6	GRAFCET à Séquences Simultanées	93
III.11	RÈGLES DE SYNTAXE DANS UN GRAFCET	94
III.12	RÈGLES D'ÉVOLUTION D'UN GRAFCET	95
III.12.1	Règle 1 : Situation initiale du GRAFCET	95
III.12.2	Règle 2 : Transition franchissable	96
III.12.3	Règle 3 : Franchissement d'une transition	97
III.12.4	Règle 4 : Franchissements simultanés	97
III.12.5	Règle 5 : Activation et désactivation simultanées	97
III.13	MISE EN ÉQUATION D'UN GRAFCET	98
III.13.1	Fonctions Transitions	99
III.13.2	Équations des Étapes	99
III.13.3	Équations des Actions	100
III.14	EXTENSION DU MODÈLE GRAFCET	103
III.14.1	Étapes Sources-Étapes Puits	103

III.14.2	Transitions Sources-Transitions Puits	104
III.15	MACRO-ÉTAPE	105
III.15.1	Principe	105
III.15.2	Symbole	105
III.16	TACHES SOUS-PROGRAMMES	107
III.17	FORÇAGE ET FIGEAGE	108
III.17.1	Forçage	109
III.17.2	Règles de Forçage	109
III.17.3	Représentation du Forçage	109
III.17.4	Types de Forçage	110
III.17.5	Figeage	111
III.18	CONCLUSION	111

CHAPITRE IV

GEMMA

IV.1	INTRODUCTION	113
IV.2	BESOIN D'OUTILS-MÉTHODES	114
IV.3	BESOIN D'UN VOCABULAIRE PRÉCIS	114
IV.4	BESOIN D'UNE APPROCHE GUIDÉE	115
IV.5	QU'EST QUE LE GEMMA ?	115
IV.6	CONCEPTS DE BASE DU GEMMA	116
IV.6.1	Système en Ordre de Marche	116
IV.6.2	Marche en Production	117
IV.6.3	Familles de Modes de Marches et d'Arrêts	117
IV.7	MODE D'EMPLOI DU GEMMA	123
IV.7.1	Sélection des Modes de Marches et d'Arrêts	124
IV.7.2	Conditions d'Évolution entre les Modes de Marches et d'Arrêts	124
IV.8	SÉLECTION DES MODES ET DES ÉVOLUTIONS	125
IV.9	EFFET DU GEMMA SUR LE GRAFCET	126
IV.10	FINALITÉ DU GEMMA	127
IV.10.1	GRAFCET de Conduite	127
IV.10.2	Pupitre de Commande	127
IV.11	BOUCLES OPÉRATIONNELLES	127
IV.11.1	Marche de Production Automatique	128
IV.11.2	Marche Automatique avec Arrêt d'Urgence	128
IV.11.3	Marche de Production Cycle par Cycle	129
IV.11.4	Marche de Production à Cycle Répété	129
IV.11.5	Marche de Vérification Manuelle dans l'Ordre	130
IV.11.6	Marche de Vérification Manuelle dans le Désordre	130

IV.II.7	Marche Automatique avec Marche de Préparation et de Clôture	131
IV.II.8	Marche de Redémarrage à Partir de d'État Initiale	132
IV.II.9	Marche de Redémarrage à Partir de d'État Quelconque	132
IV.I2	GRAFCETS ASSOCIÉS AUX GEMMA	133
IV.I2.1	GRAFCET de Conduite G.C.	133
IV.I2.2	GRAFCET de Sûreté G.S.	133
IV.I2.3	GRAFCET de Fonctionnement Normal G.F.N.	133
IV.I2.4	GRAFCET de Traitement de l'Arrêt d'Urgence G.T.A.U.	133
IV.I2.5	GRAFCET d'INITialisation G.I.N.I.T.	134
IV.I2.6	GRAFCET de REGlage G.R.E.G.	134
IV.I2.7	GRAFCET de Coordination de Tâche G.C.T.	134
IV.I3	MISE EN ŒUVRE DU GEMMA	134
IV.I4	CONCLUSION	138

CHAPITRE V **ARCHITECTURE DES API**

V.1	INTRODUCTION	140
V.2	CONTRAINTES DU MONDE INDUSTRIEL	141
V.3	POURQUOI L'AUTOMATISATION ?	141
V.4	DOMAINES D'EMPLOI DES AUTOMATES	142
V.5	AUTOMATES PROGRAMMABLES INDUSTRIELS	142
V.5.1	Définition	143
V.5.2	Rôle des APIs	143
V.5.3	Fonctions Principales des APIs	144
V.5.4	Environnement des APIs	145
V.5.5	Avantage des APIs	145
V.5.6	Inconvénient des APIs	146
V.6	ARCHITECTURE DES AUTOMATES	146
V.6.1	Aspect Extérieur des APIs	147
V.6.2	Aspect Intérieur des APIs	149
V.7	FONCTIONNEMENT D'UN AUTOMATE	156
V.7.1	Fonctions d'Acquisition des Informations	157
V.7.2	Fonctions de Traitement des Informations	157
V.7.3	Fonctions de Commande de la P.O.	160
V.7.4	Chaîne Fonctionnelle	160
V.7.5	Chaîne d'Action	161
V.7.6	Chaîne d'Acquisition	162
V.8	COMMUNICATIONS DE L'AUTOMATE	162
V.8.1	Dialogue de Programmation	163

V.8.2	Dialogue d'Exploitation	163
V.8.3	Dialogue de Supervision	164
V.9	PROTECTION DE L'AUTOMATE	164
V.10	CRITÈRES DE CHOIX D'UN AUTOMATE	165
V.11	CABLAGE DES AUTOMATES	166
V.11.1	Présentation de l'Automate	166
V.11.2	Alimentation de l'Automate	167
V.11.3	Câblage des Entrées/Sorties	167
V.11.4	Câblage de l'Automate/Actionneur	167
V.12	RÉSEAUX DES AUTOMOMES	169
V.12.1	Principe	169
V.12.2	Bus de Terrain	170
V.12.3	Différents Types de Réseaux d'Automates	171
V.13	CONCLUSION	173

CHAPITRE VI

PROGRAMMATION DES API

VI.1	INTRODUCTION	174
VI.2	NORME CEI 1131	174
VI.3	FONCTIONNEMENT DE L'API	177
VI.3.1	Entrées	177
VI.3.2	Exécution du Programme	178
VI.3.3	Mise à Jour des Sorties	178
VI.4	STRUCTURE LOGICIELLE MONOTACHE	179
VI.4.1	Exécution Cyclique	180
VI.4.2	Exécution Périodique	180
VI.5	STRUCTURE LOGICIELLE MULTI-TACHES	181
VI.6	LANGAGES DE PROGRAMMATION POUR API	182
VI.6.1	C'est quoi un Programme ?	182
VI.6.2	Adressages des Entrées et des Sorties d'un API	184
VI.6.3	Langage de Programmation LADDER	186
VI.6.4	Présentation du Langage LADDER	187
VI.6.5	Présentation du Langage FBD	187
VI.6.6	Présentation du Langage IL	188
VI.7	PROGRAMMATION DES API	190
VI.7.1	Fonctions Logiques	190
VI.7.2	Fonctions de Mémoire	193
VI.7.3	Temporisations	194
VI.7.4	Fonctions de Comptage	198

	VI.7.5 Fonctions de Régulation	200
VI.8	CONCLUSION	204

TRAVAUX DIRIGÉS

VII.1	TRAVAUX DIRIGÉS N° 1	205
VII.2	TRAVAUX DIRIGÉS N° 2	207
VII.3	TRAVAUX DIRIGÉS N° 3	209
VII.4	TRAVAUX DIRIGÉS N° 4	210
VII.5	TRAVAUX DIRIGÉS N° 5	211
VII.6	TRAVAUX DIRIGÉS N° 6	212
VII.7	TRAVAUX DIRIGÉS N° 7	213
VII.8	TRAVAUX DIRIGÉS N° 8	214
VII.9	TRAVAUX DIRIGÉS N° 9	216

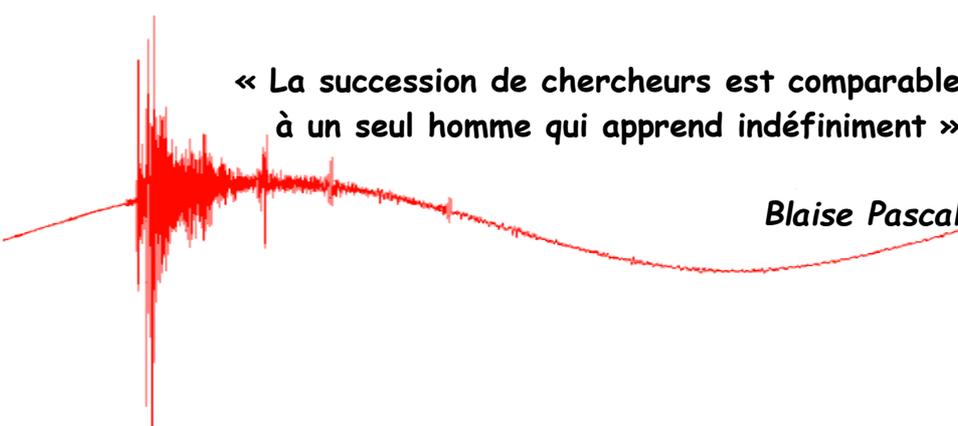
TRAVAUX PRATIQUES

VIII.1	TRAVAUX PRATIQUES N° 1	217
VIII.2	TRAVAUX PRATIQUES N° 2	219
VIII.3	TRAVAUX PRATIQUES N° 3	221
VIII.4	TRAVAUX PRATIQUES N° 4	222
VIII.5	TRAVAUX PRATIQUES N° 5	224
VIII.6	TRAVAUX PRATIQUES N° 6	225
VIII.7	TRAVAUX PRATIQUES N° 7	227

	CONCLUSION GÉNÉRALE	232
--	----------------------------	-----

	BIBLIOGRAPHIE	233
--	----------------------	-----

INTRODUCTION GÉNÉRALE



« La succession de chercheurs est comparable
à un seul homme qui apprend indéfiniment »

Blaise Pascal

INTRODUCTION GÉNÉRALE

L'Automatique est la discipline qui développe les méthodes et les moyens relatifs à la commande des systèmes. Elle concerne une grande majorité de métiers aujourd'hui et tout ingénieur se doit d'en posséder une culture minimale. Ses développements sont relativement récents et la fin du XX^{ième} siècle a vu l'accélération de ses champs d'application, à une vitesse telle que le monde technologique et industriel s'est trouvé créateur de savoirs, qui jusqu'ici étaient élaborés exclusivement dans les laboratoires de recherche des universités et des grandes écoles.

Une synergie entre les développements des industriels et la formalisation qui caractérise le monde universitaire s'est alors mise en place, aboutissant à une refonte dans la manière d'aborder l'automatique dans les différents cycles de formation. Celle-ci ne pouvait plus n'être abordée qu'en deuxième cycle, voire en troisième cycle, mais se devait de l'être au plus tôt dans la formation des ingénieurs, afin qu'ils en possèdent les bases, quels que soient leurs futurs parcours.

La difficulté pédagogique immédiate fut d'introduire une discipline, jusqu'ici de synthèse et enseignée en fin de formation, à de jeunes bacheliers ayant encore peu de bagages scientifiques et, pour la grande majorité d'entre eux, aucune culture technologique.

L'automatique se devait donc de se transformer en discipline initiale, conjuguant la rigueur mathématique maximale accessible aux étudiants de premier cycle à une approche intuitive, nécessaire à ce niveau, et permise à l'aide du puissant outil pédagogique que sont les travaux pratiques.

Ce polycopié est le fruit de plusieurs années d'expériences d'enseignement universitaire. Il est difficile de lister toutes celles ou ceux qui, d'une manière ou d'une autre, ont contribué aux réflexions qui ont permis sa réalisation. Mes pensées vont immédiatement à mes étudiants, dont la curiosité, l'intransigeance et la maturité intellectuelle sont le meilleur des ferments. Nombreuses ont été également les discussions pédagogiques avec mes collègues enseignants et les heures passées, ensemble, à mettre au point des travaux pratiques qui, de manière informelle, se retrouvent certainement dans ces pages.

Ce manuscrit est un cours d'automatique au sens large, transposable à tous les systèmes commandés, il s'adresse aux étudiants de la formation Licence L3, pour l'option Automatique (AUT). Par extension, ce cours s'adresse à toute personne, étudiants ou chercheurs concernés par le domaine de génie électrique (électrotechnique, électronique, télécommunication, informatique industrielle ... etc), et qui désire avoir un document de base ou approfondir leurs connaissances en matière de l'automatisme des systèmes de production et de l'informatique industrielle.

J'ai voulu que ce manuscrit soit un outil facile à utiliser, de produire des contributions de qualité, de permettre à l'étudiant d'améliorer ses compétences en ce qui a trait à décrire fonctionnellement un Système Automatisé de Production (SAP) ; faire le choix d'une technologie de commande ; analyser un système de production ; programmer un Automate Programmable Industriel (API). Il vise à fournir à l'étudiant les connaissances nécessaires qui lui permettent d'appréhender les concepts principaux utilisés pour mettre en œuvre les moyens modernes pour préparer dans les meilleures conditions leurs études.

Toutefois, les illustrations, ainsi que les exercices et travaux proposés, se font très majoritairement sur des systèmes industriels, afin de constituer un

ensemble cohérent et de forger, au fil des pages, une culture des solutions aujourd'hui mises en œuvre dans ce domaine.

Pour cela ce cours a été élaboré dans un esprit pédagogique de cours magistral, de manière à faciliter sa lecture et son assimilation, dans cet ordre j'exploite en plusieurs chapitres sur un système automatisé de production dont le but de stimuler les idées. L'étude a été conçu avec soin afin de maitre l'accent sur des différents applications d'automatisation conçue autour d'automates programmables industriels.

Le polycopié est organisé en huit chapitres :

- ✎ Le *premier chapitre* est une présentation de la fonction globale d'un système, Automatisation et structure des systèmes automatisés. La modélisation d'un système automatisé de production est très bien illustrée ainsi que la structure détaillée d'un système automatisé liées au vocabulaire technique, bien évidemment, le vocabulaire de l'Automatique est entièrement défini. Je décris d'abord une définition illustrée avec des exemples concrets sur les pré-actionneurs (contacteurs, triac, ...), les actionneurs (vérins, moteurs, ...) et les capteurs.
- ✎ Dans le *second chapitre* je présente les notions de base de la modélisation des systèmes automatisés par les réseaux de Pétri (Rdp) et on définit les différentes propriétés d'un Rdp avec les différents formalismes concernant cet outil, est ensuite les structures particulières des Réseaux de Pétri (RdP) sont présentées.
- ✎ Le *troisième chapitre* débute avec les notions sur la définition et notions de bases, règles d'établissement du GRAFCET (GRAPhe Fonctionnel de Commande par Etapes et Transitions) ou SFC (Sequential Fonction Chart), transitions et liaisons orientées, règles d'évolution, sélection de séquence et

séquences simultanées, organisation des niveaux de représentation, matérialisation d'un GRAFCET, est ensuite présentée.

- ✎ J'ai consacré le *quatrième chapitre* aux rappels sur le concept et la structuration du GEMMA (Guide d'Etude des Modes de Marches et d'Arrêts) d'un système automatisé de production, procédures de fonctionnement, d'arrêt et les procédures en défaillances et l'utilisation pratique du GEMMA et applications.
- ✎ J'ai réservé le *cinquième chapitre* aux rappels sur la structure interne et description des éléments d'un A.P.I. Le travail est présenté par étapes successives afin de mieux cerner la problématique de choix de l'unité de traitement, choix d'un automate programmable industriel, les interfaces d'entrées-sorties, outils graphiques et textuels de programmation, mise en œuvre d'un automate programmable industriel et le principes des réseaux d'automates.
- ✎ Le *sixième chapitre* s'adresse au trois types de langages utilisés dans les automates aujourd'hui : IL, FBD et LADDER. Chaque type de langage est illustré par des applications industrielles.
- ✎ Le *septième chapitre* est occupé par une large série sur les travaux dirigés de l'automatisme industriel des systèmes de production.
- ✎ Le *huitième chapitre* contient des travaux pratiques. Cependant, les illustrations, ainsi que ces travaux et activités suggérés, sont principalement réalisés sur des systèmes industriels.

Enfin, je termine ce polycopié par une conclusion sur l'étude de contrôle-commande, les systèmes industriels automatisés et de connaître la technologie des principaux constituants des systèmes automatisés de production.

CHAPITRE I

GÉNÉRALITÉS SUR LES SYSTÈMES AUTOMATISÉS



CHAPITRE I

GÉNÉRALITÉS SUR LES SYSTÈMES AUTOMATISÉS

I.1. INTRODUCTION

Un système est un produit qui devra vérifier les performances attendues du cahier des charges. On peut donc regarder un système suivant deux aspects (aspect structurel, aspect fonctionnel). Ces dernières années, les avancées technologiques ont conduit au développement des automates programmables industriels (API) et à une révolution importante dans l'automatique. Ce chapitre, une introduction aux API, a pour objectif de faciliter le travail des ingénieurs praticiens qui font leurs premiers pas dans le domaine des automates programmables industriels. Ce chapitre est de présenter les fonctions globales d'un système, automatisation et structure des systèmes automatisés, pré-actionneurs (contacteurs, triac, ...), actionneurs (vérins, moteurs, ...), capteurs, classification des systèmes automatisés, spécification des niveaux du cahier des charges, outils de représentation des spécifications fonctionnelles.

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

I.2. SYSTÈME

Le mot « *système* » en grec est « *systema* » qui signifie « *ensemble organisé* ». En général un système est un ensemble d'éléments en interaction dynamique, organisé en fonction d'un but. La fonction globale de tout système est de conférer une valeur ajoutée à un ensemble de matières d'œuvre dans un contexte donnée [1-5].

I.3. SYSTÈME DE PRODUCTION

Un Système de Production a pour rôle de conférer une valeur ajoutée à un ensemble de produits et matériaux bruts pour obtenir un produit fini ou un produit intermédiaire qui va, à son tour, participer à l'élaboration d'un produit fini. Le système de production reçoit donc un flux de matières d'œuvre (brutes), consomme divers types d'énergie (électrique, pneumatique, hydraulique, etc.) et consomme un certain nombre de composants auxiliaires (eau, liquide de refroidissement, huile, etc.) [6]. Il génère un flux de produits élaborés et un flux de déchets et de nuisances diverses. Un système de production est dit industriel si l'obtention de cette valeur ajoutée a un caractère reproductible et peut être exprimée et quantifiée en terme économique [7]. (M.O.S : matière d'œuvre entrante, M.E.S : matière d'œuvre sortante) (Fig. I-1).

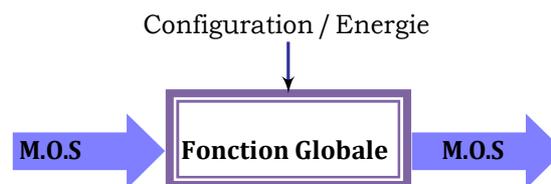


Figure I-1. Système de production

I.4. SYSTÈME DE PRODUCTION AUTOMATISÉ (SAP)

L'automatique est la discipline scientifique permettant de caractériser les systèmes automatisés et de choisir/concevoir/réaliser la commande des systèmes. Les systèmes de commande s'inspirent le plus souvent de l'homme.

Un système automatisé ou automatique est un système réalisant des opérations et pour lequel l'homme n'intervient que dans la programmation du système et dans son réglage. Il s'appelle aussi un système technique commandable. On dit qu'un système est commandable si en faisant varier uniquement les entrées, on peut faire subir des modifications au système, afin qu'il atteigne un objectif fixé en un temps fini [5-8].

L'automatisation de la production consiste à transférer tout ou partie des tâches de coordination, auparavant exécutées par des opérateurs humains, dans un ensemble d'objets techniques appelé *Partie Commande (PC)*.

La Partie Commande (PC) mémorise le *savoir-faire* des opérateurs pour obtenir la suite des actions à effectuer sur les matières d'œuvre afin d'élaborer la valeur ajoutée.

Elle exploite un ensemble d'informations prélevées sur la *Partie Opérative (PO)* pour élaborer la succession des ordres nécessaires pour obtenir les actions souhaitées.

La fonction globale de tout système automatisé est de conférer une *valeur ajoutée* à un ensemble de *matières d'œuvre* dans un *environnement* ou *contexte donné*.

I.4.1. Matières d'Œuvre

Une matière d'œuvre peut se présenter sous plusieurs formes [9,10] :

- **un produit**, c'est-à-dire de la matière, à l'état solide, liquide ou gazeux, et sous une forme plus ou moins transformée :
 - ▶ des objets techniques : lingot, roulement, moteur, véhicule ... ;
 - ▶ des produits chimiques : pétrole, éthylène, matière plastique ... ;
 - ▶ des produits textiles : fibre, tissu, vêtement ... ;
 - ▶ des produits électroniques : transistor, puce, microprocesseur, automate programmable ... ;
 - ▶ qu'il faut : concevoir, produire, stocker, transporter, emballer, utiliser ...

- **de l'énergie**
 - ▶ sous forme : électrique, thermique, hydraulique ... ;
 - ▶ qu'il faut : produire, stocker, transporter, convertir, utiliser ...

- **de l'information**
 - ▶ sous forme écrite, physique, audiovisuelle ... ;
 - ▶ qu'il faut : produire, stocker, transmettre, communiquer, utiliser ...

■ *des êtres humains*

- ▶ pris individuellement ou collectivement ;
- ▶ qu'il faut : former, informer, soigner, transporter, servir ...

I.4.2. Valeur Ajoutée

La valeur ajoutée à ces matières d'œuvre est *l'objectif global* pour lequel a été défini conçu, réalisé, puis éventuellement modifié, le système. Cette valeur ajoutée peut résulter par exemple :

■ d'une *modification physique* des matières d'œuvre :

- ▶ traitement mécanique : usinage, formage, broyage, impression ... ;
- ▶ traitement chimique ou biologique ;
- ▶ conversion d'énergie ;
- ▶ traitement thermique : cuisson, congélation ... ;
- ▶ traitement superficiel : peinture, teinture ... ;

■ d'un *arrangement particulier*, sans modification des matières d'œuvre :

- ▶ montage, emballage, assemblage ... ;
- ▶ couture, collage ...

■ d'une *mise en position* particulière, ou d'un *transfert*, de ces matières d'œuvre :

- ▶ manutention, transport, stockage ;
- ▶ commerce ;
- ▶ communication.

■ d'un *prélèvement d'information* sur ces matières d'œuvre :

- ▶ contrôle ;
- ▶ mesure ;
- ▶ lecture examens ...

I.4.3. Contexte

La *nature*, la *quantité* et la *qualité* de la valeur ajoutée peuvent varier pour tenir compte de l'évolution des besoins de la société dans laquelle s'insère le système. Ce qui peut conduire à modifier le système, voire l'abandonner pour en construire un nouveau.

L'environnement, c'est-à-dire le *contexte* physique, social, économique, politique, ... joue un rôle essentiel dans le fonctionnement du système et influe sur la qualité et/ou la quantité de la valeur ajoutée.

I.4.4. Objectifs de l'Automatisation

L'automatisation permet d'apporter des éléments supplémentaires à la valeur ajoutée par le système. Ces éléments sont exprimables en termes d'objectifs par :

- ◆ accroître la productivité du système c'est-à-dire augmenté la quantité de produits élaborés pendant une durée donnée. Cet accroissement de productivité exprime un gain de valeur ajoutée sous forme [3-5] :
 - d'une meilleure rentabilité ;
 - d'une meilleure compétitivité ;
 - améliorer la flexibilité de production ;
 - améliorer la qualité du produit grâce à une meilleure répétabilité de la valeur ajoutée.

- ◆ s'adapter à des contextes particuliers [11-13] :
 - adaptation à des environnements hostiles pour l'homme (milieu salin, spatial, nucléaire ...) ;
 - adaptation à des tâches physiques ou intellectuelles pénibles pour l'homme (manipulation de lourdes charges, tâches répétitives parallélisées ...).

- ◆ augmenter la sécurité, etc.

I.4.5. Conduite et Surveillance d'un Système Automatisé

Il s'avère très difficile en pratique d'intégrer dans une Partie Commande (P.C) la totalité des savoir-faire humains de sorte que l'automatisation reste souvent partielle : certaines tâches restent confiées à des intervenants humains.

A ces causes techniques viennent s'ajouter des considérations économiques de compétitivité, des considérations financières imposant un fractionnement des investissements, des considérations sociales d'automatisation douce [14].

En effet, l'automatisation reste souvent partielle c'est-à-dire certaines tâches restent confiées à des intervenants humains. Ces tâches peuvent être classées en deux catégories : *conduite* et *surveillance*.

- **Conduite** : Cette catégorie regroupe les opérations de mise en marche du système, d'initialisation, de spécifications des consignes de fonctionnement, etc.
- **Surveillance** : Le modèle de fonctionnement de la Partie Commande (P.C) (choisi par le concepteur) correspond à un ensemble de situations prévues c'est-à-dire retenues par le concepteur parmi un ensemble de situations possibles. De ce fait, il est indispensable de pouvoir faire face à des situations non prévues (non retenues pour des raisons économiques compte tenu de leur faible probabilité d'apparition). A ce niveau, seul l'opérateur est appelé à intervenir et à prendre les décisions requises par cette situation. Il assure donc une fonction de surveillance.

Certaines tâches restent donc manuelles et l'automatisation devra donc prendre en compte la spécificité du travail humain, c'est-à-dire en particulier :

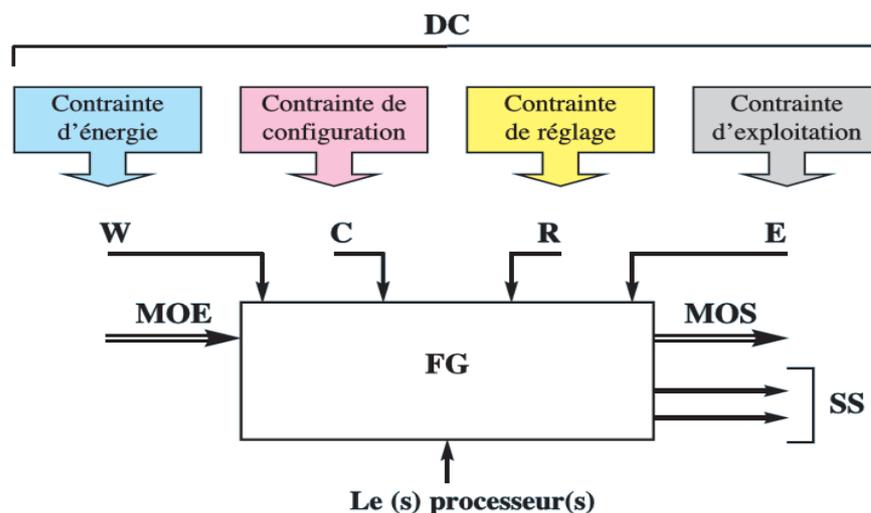
- ▶ assurer les dialogues nécessaires à l'analyse de la situation entre les intervenants et le système automatisé ;
- ▶ assurer la sécurité de ces intervenants dans l'exécution de leurs tâches manuelles.

En outre le modèle de fonctionnement de la Partie Commande, choisi par le concepteur du système, ne correspond qu'à un ensemble de situations prévues, c'est-à-dire retenues par le concepteur parmi un ensemble de situations possibles. Or il est impératif de pouvoir faire face à des situations non prévues (donc non retenues en général pour des raisons économiques compte tenu de leur faible probabilité), voire imprévisibles.

1.5. MODÉLISATION D'UN SYSTÈME AUTOMATISÉ

La modélisation d'un système technique consiste à lui donner une représentation graphique qui énumère les quatre ensembles d'éléments ci-dessous en les distinguant les uns des autres et en montrant les relations entre elles (Fig. I-2) [15,16] :

- ❶ La fonction globale, qui apporte la valeur ajoutée à la matière d'œuvre ;
- ❷ Les éléments constitutifs (le ou les processeurs) qui sont inclus dans la frontière et qui supportent la fonction globale ;
- ❸ La matière d'œuvre sur laquelle s'exerce son action ;
- ❹ Les données de contrôle, qui provoquent ou modifient la mise en œuvre de la fonction.



- | | |
|--------------------------------------|----------------------------------|
| - DC : Données de Contrôle | - MOE : Matière d'Œuvre Entrante |
| - FG : Fonction Globale | - MOS : Matière d'Œuvre Sortante |
| - Le processeur : Support de travail | - SS : Sorties Secondaires |

Figure I-2. Modélisation d'un système technique de production automatisé

I.6. CONVENTION ASSOCIÉE AU SYSTÈME AUTOMATISÉ

On représente la structure d'un système technique automatisé par le schéma suivant (Fig. I-3) :

Un system automatisé est généralement constitue par [I-II] :

- × Une partie commande (P.C) ;
- × Une partie opérative (P.O) ;
- × Des éléments d'interfaces qui relient la PC à la PO ;
- × Un pupitre permettant le dialogue entre l'opérateur et le système automatisé.

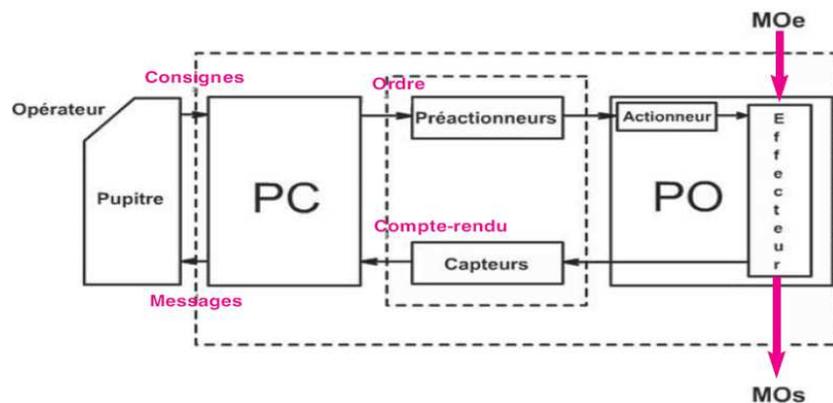


Figure I-3. Structure d'un système technique de production automatisé

I.7. STRUCTURE DE BASE D'UN SYSTÈME AUTOMATISÉ

Un système de production est dit automatisé lorsqu'il peut gérer de manière autonome un cycle de travail pré établi 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 (Fig. I-4) [8-16]:

- la Partie Opérative (P.O) ;
- la Partie Commande (P.C) ou Système de Contrôle/Commande (S.C.C) ;
- la Partie Relation (P.R) de plus en plus intégrée dans la partie commande.

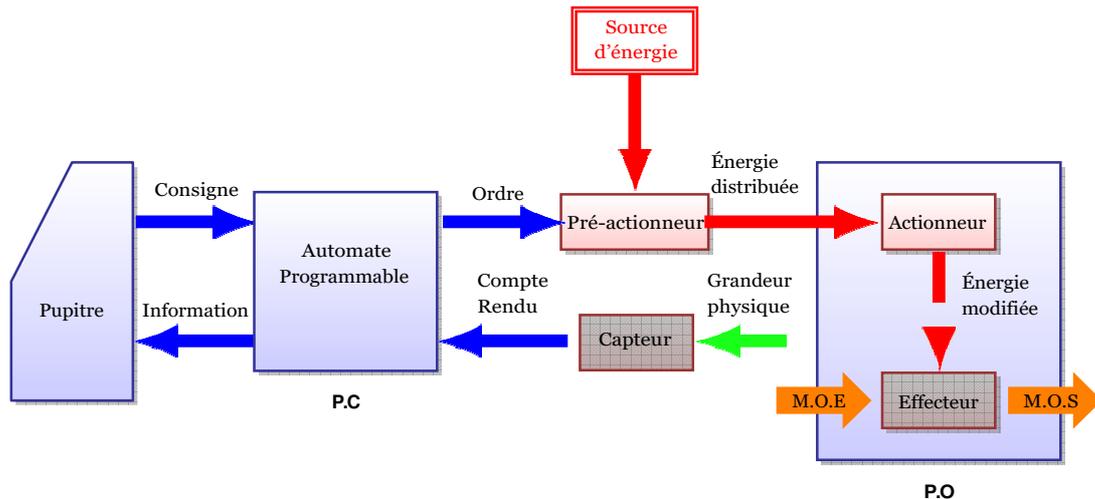


Figure I-4. Structure d'un système automatisé

I.7.1. Partie Opérative (P.O)

C'est la partie visible du système. Elle comporte les éléments du procédé, c'est à dire les éléments mécaniques du mécanisme. Elle procède au traitement des matières d'œuvre afin d'élaborer la valeur ajoutée. Elle est formée de l'ensemble des divers organes physiques qui interagissent sur le produit pour lui conférer une valeur ajoutée : les pré-actionneurs, les actionneurs, les effecteurs et les capteurs (Fig. I-5, Fig. I-6).

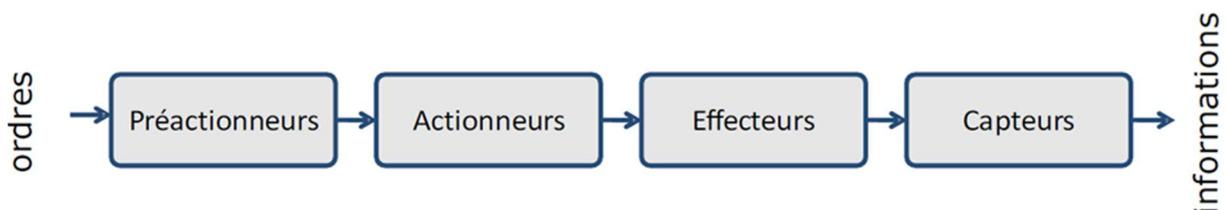
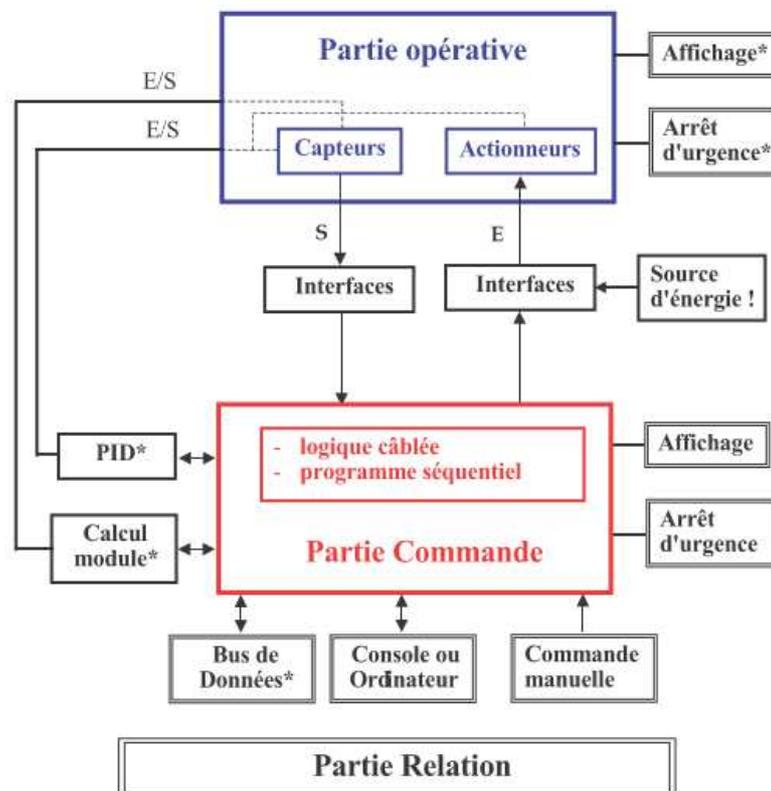


Figure I-5. Partie Opérative (PO)

- ⊕ **des pré-actionneurs** (*distributeurs, contacteurs*) qui reçoivent des ordres de la partie commande, qui permet l'adaptation des ordres de faible énergie en ordres adaptés aux actionneurs (ils servent de relais de puissance entre la commande et les actionneurs) ;
- ⊕ **des actionneurs** (*vérins, moteurs, électrovannes*) qui ont pour rôle d'exécuter ces ordres. Ils transforment l'énergie pneumatique (air comprimé), hydraulique (huile sous pression) ou électrique en énergie mécanique (ils

transforment l'énergie reçue en une énergie exploitable par les effecteurs qui agissent et transforment le produit) ;

- ✦ **des capteurs** (*fin de course, détecteur de position, détecteur de présence, capteur de température ...*) qui informent la partie commande de l'exécution du travail. Par exemple, on va trouver des capteurs mécaniques, pneumatique 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 (ils recueillent les informations : état ou position du produit, alarmes, etc. traduisant un changement d'état du procédé. Ces changements induisent le calcul de la commande du procédé de production par son Système de Contrôle/Commande ou la Partie Commande) ;
- ✦ **Les effecteurs** (*outils de coupe, tête de soudure, bras, pince ...*) sont des dispositifs terminal qui agit sur la matière d'œuvre pour lui donner sa valeur ajoutée.



* En option

Figure I-6. Procédé automatisé

I.7.2. Partie Commande (P.C)

La Partie Commande (P.C) ou Système de Contrôle/Commande (S.C.C) est le 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 (P.O), 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 Etape / Transition (GRAFCET). La Partie Commande (P.C) coordonne la succession des actions sur la Partie Opérative (P.O) avec la finalité d'obtenir la valeur ajoutée [1-5].

La Partie Commande regroupe l'ensemble des composants permettant le traitement des informations reçues de la Partie Opérative (P.O) et des ordres envoyés par la Partie Relation (P.R) [4,5].

La Partie Commande (P.C) se compose de :

- ▶ L'automate programmable industriel (Fig. I-7) ;
- ▶ Les interfaces d'entrées qui transforment les informations issues des capteurs placés dans la Partie Opérative (P.O) ou dans la partie dialogue en informations de nature et d'amplitude compatibles avec les caractéristiques technologiques de l'automate ;
- ▶ Les interfaces de sorties qui transforment les informations élaborées l'unité de traitement en informations de nature et d'amplitude compatibles avec les caractéristiques technologiques des pré-actionneurs d'une part, des visualisations et avertisseurs d'une part.



Figure I-7. Procédé automatisé

La Partie Commande (P.C) :

- ⊕ Élabore les ordres à partir de ces informations et informe l'opérateur de l'état du système ;
- ⊕ pilote le fonctionnement du système automatisé ;
- ⊕ Ses autres fonctions sont :
 - ✱ Dialoguer avec l'opérateur via la partie relation ;
 - ✱ Acquérir des données en provenance des capteurs de la partie opérative ;
 - ✱ Traiter les données acquises en entrée :
 - Surveillance (sûreté de fonctionnement, diagnostic ...) ;
 - Régulation du comportement des systèmes de la partie opérative ;
 - ...
- ⊕ Une fois que la PC a terminé son traitement, elle transmet des ordres aux pré-actionneurs de la PO et des informations aux composants de signalisation de la PR.
- ⊕ Pour communiquer avec la PO et la PR, la PC utilise des interfaces d'entrées et de sorties.

I.7.3. Partie Relation (P.R)

La Partie Relation (P.R), la Partie Supervision (P.S), la Partie Echange (P.E) ou bien l'Interface Homme Machine (I.H.M) 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, arrêt d'urgence, marche automatique, etc. L'outil de description s'appelle le Guide d'Etudes des Modes de Marches et d'Arrêts (G.E.M.M.A) [3].

Les outils graphiques, que sont le GRAFCET et le GEMMA, sont utilisés par les automaticiens et les techniciens de maintenance, pour la recherche des pannes sur les SAP (Système Automatisé de Production) [6].

Pendant le fonctionnement, un dialogue continu s'établit entre les trois secteurs du système, permettent ainsi le déroulement correct du cycle défini dans le cahier de charges.

La Partie Relation (P.R) se compose de deux ensembles [2-8] :

- ▶ Les visualisations et avertisseurs qui transforment les informations fournies par l'automate en informations perceptibles par l'homme (informations optiques ou sonores) ;
- ▶ Les capteurs qui transforment les informations fournies par l'homme (action manuelle sur un bouton poussoir, par exemple) et informations exploitables par l'automate.

L'Interface Homme Machine (I.H.M) est équipée d'organes permettant (Fig. I-8) [6] :

- ⊕ La mise en/hors énergie de l'installation ;
- ⊕ La sélection des modes de marche ;
- ⊕ La commande manuelle des actionneurs ;
- ⊕ Le départ des cycles de fonctionnement ;
- ⊕ L'arrêt d'urgence ;
- ⊕ D'informer l'opérateur de l'état de l'installation : *voyants lumineux, afficheurs, écrans, vidéos, klaxons ...*



Figure I-8. Interface Homme-Machine (IHM)

I.7.4. Relations entre PC, PO et PR

Un procédé industriel passe d'un état initial à un état final via une succession d'états intermédiaires qui marquent son évolution. Cette évolution est surveillée grâce aux capteurs qui renvoient des comptes rendus de l'état du système. La Partie Commande (P.C) traite ces informations et élabore les ordres renvoyés vers les actionneurs (consignes, ordres, comptes rendus, signalisations) :

- ✘ La PR émet à la PC des consignes (demande d'ouverture, de fermeture, d'arrêt ...).
- ✘ La PC émet à la PO des ordres pour coordonner le déroulement des opérations.
- ✘ La PO émet des comptes rendus caractérisant les états des matières d'œuvres ou des parties mécaniques.

Tout système automatisé comporte les fonctions suivantes [7,8] :

- ❑ **Agir sur la matière d'œuvre** : C'est la partie opérative qui réalise ce pour quoi le système a été conçu.
- ❑ **Acquérir les informations** : Ce sont les capteurs qui permettent de connaître toutes les informations nécessaires au bon fonctionnement du système.
- ❑ **Dialoguer avec l'opérateur** : C'est les ordres et les comptes rendus qui permettent à l'opérateur de savoir à chaque instant l'état du système et son évolution.
- ❑ **Communiquer** : C'est tout ce qui permet au système de communiquer avec d'autre système pour une gestion automatisé de la production par exemple.
- ❑ **Traiter les données** : C'est le cœur du système, cette fonction est celle où l'on effectue tous les calculs nécessaires au bon fonctionnement du

ystème. Les signaux entrants et sortants de cette fonction sont adaptés du point de vue énergétique par des circuits d'interfaçage.

I.8. STRUCTURE MATÉRIELLE DE LA PARTIE OPÉRATIVE

I.8.1. Pré-actionneurs

Le pré-actionneur est l'élément qui distribue l'énergie disponible aux actionneurs sur ordre de la partie commande. Les pré-actionneurs sont principalement de type tout ou rien (TOR). Ils commandent exclusivement l'établissement et l'interruption de la circulation de l'énergie la source et l'actionneur [17-19].

I.8.1.1. Contacteur et relai

Un contacteur (pré-actionneur électrique pour les moteurs) (Fig. I-9) est un appareil électrotechnique destiné à établir ou interrompre le passage du courant, à partir d'une commande.

Le contacteur est un appareil de commande capable d'établir ou d'interrompre le passage de l'énergie électrique (c'est sa fonction). Il a donc un pouvoir de coupure non nul. Vous deviez appeler ce type de constituant pré-actionneur puisqu'il se trouve avant l'actionneur dans la chaîne des énergies. Ce dernier peut être commandé à distance au moyen de contacts actionnés manuellement (bouton poussoir) ou automatiquement (asservi à une grandeur physique : pression, température, vitesse, etc.) [12].

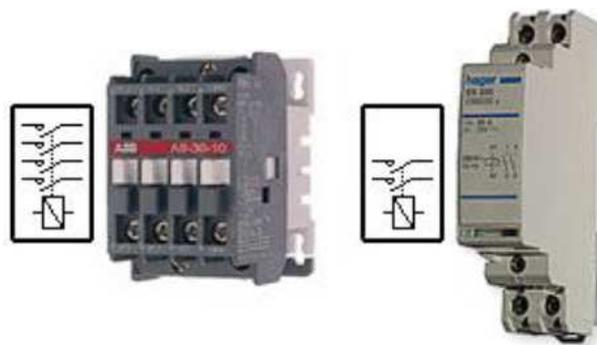


Figure I-9. Pré-actionneur électrique type contacteur

1.8.1.2. Variateur de vitesse

Le variateur de vitesse fait aussi partie des pré-actionneurs proportionnels pour les moteurs électriques, ce n'est pas d'un type tout ou rien (TOR). Il laisse passer seulement une partie de l'énergie source, c'est à dire qu'il régule le débit d'énergie, on parle alors de pré-actionneur proportionnel (Fig. I-10) [5].



Figure I-10. Pré-actionneur électrique-proportionnel type variateur de vitesse

1.8.1.3. Distributeur

Le distributeur (pré-actionneur pneumatique ou hydraulique) (Fig. I-11) est un pré-actionneur associé à un vérin pneumatique (ou hydraulique). Il commande la circulation de l'énergie entre la source et l'actionneur [4].

Un distributeur pneumatique que l'on caractérise par [20] :

- ▶ son dispositif de commande (mécanique, électrique ou pneumatique) ;
- ▶ sa stabilité (monostable ou bistable) ;
- ▶ le nombre d'orifices de passage de fluide qu'il présente dans chaque position.



Figure I-11. Pré-actionneur pneumatique /hydraulique type distributeur

Les distributeurs sont utilisés pour commuter et contrôler le débit du fluide sous pression, comme des sortes d'aiguillage, à la réception d'un signal de commande qui peut être mécanique, électrique ou pneumatique. Ils permettent de [20] :

- ✖ contrôler le mouvement de la tige d'un vérin ou la rotation d'un moteur hydraulique ou pneumatique (distributeurs de puissance).
- ✖ choisir le sens de circulation d'un fluide (aiguiller, dériver, etc.) ;
- ✖ exécuter, à partir d'un fluide, des fonctions logiques (fonctions ET, OU, mémoire, etc.) ;
- ✖ démarrer ou arrêter la circulation d'un fluide (robinet d'arrêt, bloqueur, ...).

Selon le critère de stabilité, il existe deux types de pré-actionneurs [20] :

- **Un pré-actionneur monostable** : on parle d'un pré-actionneur monostable, s'il nécessite un ordre pour passer de sa position repos à sa position travail et il revient automatiquement à sa position repos lorsque l'ordre disparaît : donc il est stable uniquement dans sa position repos. *Exemples* : un bouton poussoir, un relais électromagnétique.
- **Un pré-actionneur bistable** : on parle d'un pré-actionneur bistable s'il nécessite un ordre pour passer de sa position repos à sa position travail, il restera ensuite dans cette position à la disparition de cet ordre. Il ne peut regagner sa position repos que s'il reçoit un nouvel ordre : il est stable dans la position repos et la position travail. *Exemples* : un interrupteur, un télérupteur.

I.8.1.4. Principe de la symbolisation

① Nombre de cases

Il représente le nombre de positions de commutation possibles, une case par position. Chaque position du distributeur est symbolisée par un carré (Fig. I-12).

Distributeur à deux (02) positions



Distributeur à trois (03) positions



Figure I-12. Cases (positions) de distributeur

② Flèches

Dans chaque case ou position, les voies sont figurées par des flèches indiquant le sens de circulation du fluide entre les orifices. T : les orifices non utilisés dans une position sont symboliquement obturés par un T droit ou inversé. Le nombre des orifices est déterminé pour une position et est égal pour toutes les positions (Fig. I-13) [20].



Figure I-13. Flèches de distributeur

③ Source de pression

Elle est indiquée par un cercle noirci en hydraulique, clair en pneumatique. Echappement est symbolisé par un triangle noirci en hydraulique, clair en pneumatique (Fig. I-14, Fig. I-15) [20].



Figure I-14. Source de pression de distributeur

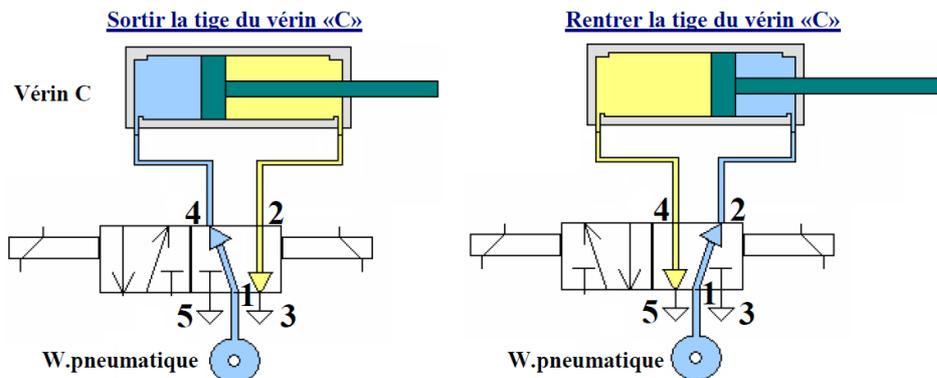


Figure I-15. Ordre d'exécution vers pré-actionneurs pneumatique (Distributeur 5/2)

Tableau I-1. Principaux types de distributeurs

Désignation	Fonction	Symbole
Distributeur 2/2	Distributeur avec : deux positions de commutation et deux raccords	
Distributeur 3/2	Distributeur avec : deux positions de commutation et trois raccords	
Distributeur 4/2	Distributeur avec : deux positions de commutation et quatre raccords	
Distributeur 5/2	Distributeur avec : deux positions de commutation et cinq raccords	
Distributeur 5/3	Le piston du vérin d'entraînement n'exerce aucune force sur la tige du piston. La tige se déplace librement	
Distributeur 5/3	La tige du piston ne bouge pas, centre fermé	
Distributeur 5/3	La tige du piston des vérins à simple tige se déploie avec une force réduite.	

I.8.2. Actionneurs

Les actionneurs sont des convertisseurs électromécaniques conçus pour mettre en mouvement des systèmes mécanique à partir de commande électrique. Ce sont essentiellement des moteurs et des vérins. Ils produisent de l'énergie mécanique à partir d'énergie électrique, hydraulique ou pneumatique mais toujours contrôlés par des signaux de commande électrique [10,20].

L'actionneur est un élément qui convertit une énergie d'entrée non directement utilisable par les mécanismes agissant sur la matière d'œuvre en

une énergie de sortie utilisable par ces mécanismes pour obtenir une action définie. Pour exécuter les ordres de la partie commande, la partie opérative est équipée des actionneurs. Les actionneurs sont le plus souvent des composants électroniques capables de produire un phénomène physique (déplacement, dégagement de chaleur, émission de lumière ...) à partir de l'énergie qu'il reçoit.

1.8.2.1. Actionneur électriques

Les moteurs (actionneurs électriques) sont des appareils très utilisés dans les ateliers de production industrielle. Ils trouvent leur utilisation dans des milieux très variés comme les secteurs alimentaires ou pharmaceutiques, le montage automatisé de certains ensembles mécaniques dans l'industrie automobile, l'outillage à main (perceuses, visseuses pneumatiques, etc.) [10].

Les moteurs électriques sont devenus incontournables dans le monde actuel. À la différence des entraînements pneumatiques, principalement employés pour la production industrielle du fait de leur simplicité et de leur fiabilité, les moteurs électriques sont également utilisés dans le cercle privé. Les exemples ne manquent pas : lave-linge, sèche-cheveux, lecteurs de CD, jouets, robots ménagers, ventilateurs, pour n'en citer que quelques-uns. Les moteurs électriques trouvent également de nombreuses applications dans le secteur automobile, puisqu'ils sont associés à une foule de systèmes de confort, comme les lève-vitres ou le réglage des sièges.

Selon la nature de l'action sur l'effecteur ou la matière d'œuvre elle-même, il existe plusieurs grands types d'actionneurs électriques (Fig. I-16) :

- ***Les moteurs rotatifs*** qui transforment l'énergie électrique en énergie mécanique de rotation : moteurs à courant continu, moteurs à courant alternatif (synchrone, asynchrone...), moteurs pas à pas (pour les dispositifs de positionnement à faible puissance).

- *Les moteurs linéaires* ou *électro-aimants* qui transforment l'énergie électrique en énergie mécanique de translation.
- *Les résistances chauffantes* qui transforment l'énergie électrique en énergie calorifique [8].



Figure I-16. Actionneurs électriques

Avantages

- ✓ réponse rapide et précise ;
- ✓ commande simple ;
- ✓ faciles à mettre en application dans des conceptions modulaires et nouvelles ;
- ✓ relativement pas cher.

Inconvénients

- ✘ il peut créer des dégâts dans des secteurs inflammables à cause de l'arc électrique ;
- ✘ surchauffe une fois calé ;
- ✘ exige des freins pour se bloquer.

I.8.2.2. Actionneurs pneumatiques

Un vérin pneumatique est un actionneur qui permet de transformer l'énergie de l'air comprimé en un travail mécanique. Un vérin pneumatique est soumis à des pressions d'air comprimé qui permettent d'obtenir des mouvements dans un sens puis dans l'autre. Les mouvements obtenus peuvent être linéaires ou rotatifs [8].

Les vérins réalisent des mouvements linéaires. Ils sont munis d'un piston avec une tige qui se déplace librement à l'intérieur d'un tube. En fonction du type ils ont un ou deux orifices permettant l'admission ou l'échappement de l'air. La longueur du mouvement définit la course du vérin, le diamètre est lié à la force à exécuter au cours du mouvement.

Ces actionneurs peuvent soulever, pousser, tirer, serrer, tourner, bloquer, percuter, abloquer. Ils sont utilisés généralement dans les applications qui nécessitent des mouvements séquentiels simples. Son principe de fonctionnement consiste à convertir une énergie d'entrée pneumatique en énergie utilisatrice mécanique. Il existe des actionneurs pneumatiques linéaires ou vérins et des actionneurs pneumatiques rotatifs. La classification des actionneurs pneumatiques tient compte du mode d'action de la tige : simple effet ou double effet (Fig. I-17).



Figure I-17. Actionneurs pneumatiques (vérins)

Avantages

- ✓ vitesse de fonctionnement élevée ;
- ✓ disponibilité d'air ;
- ✓ n'exige aucune canalisation de retour ;
- ✓ propre ;
- ✓ peut être utilisé dans un milieu explosif ;
- ✓ robuste ;
- ✓ relativement moins cher.

Inconvénients

- ✘ la compressibilité d'air entrave la commande de vitesse ;
- ✘ moins précis ;
- ✘ l'échappement crée le bruit ;
- ✘ exige le séchage additionnel et le filtrage.

I.8.2.3. Actionneurs hydrauliques

Un vérin hydraulique est un tube cylindrique (le cylindre) dans lequel une pièce mobile (le piston) sépare le volume du cylindre en deux chambres isolées l'une de l'autre. Un ou plusieurs orifices permettent d'introduire ou d'évacuer un fluide dans l'une ou l'autre des chambres et ainsi de déplacer le piston, le piston muni d'une tige se déplace librement à l'intérieur d'un tube. Pour faire sortir la tige, on applique une pression sur la face avant du piston, et sur la face arrière pour faire rentrer la tige (Fig. I-18) [17].

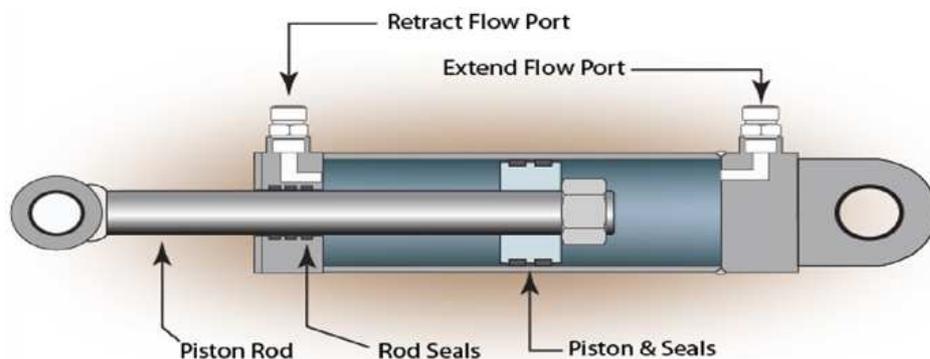


Figure I-18. Actionneur hydraulique (vérin)

Ces actionneurs trouvent leur utilité dans les mouvements qui exigent des forces très élevées à faible vitesse. Leur principe de fonctionnement consiste à convertir une énergie d'entrée hydraulique (débit d'huile + pression) en énergie utilisatrice mécanique. Une pompe électrique fournit de l'huile à l'actionneur via des tuyaux. L'actionneur est équipé d'une canalisation de retour sert à récupérer l'huile qui s'échappe lors du fonctionnement et la ramener vers la

pompe après filtrage et refroidissement éventuel. Nous pouvons trouver les actionneurs hydrauliques comme des vérins linéaires ou rotatifs.

Avantages

- ✓ vitesse de fonctionnements modérés ;
- ✓ fonctionnement doux à de basses vitesses ;
- ✓ Supporter les charges lourdes ;
- ✓ rapport élevé de puissance-à-poids ;
- ✓ l'incompressibilité du fluide aide à réaliser une commande précise ;
- ✓ robuste ;
- ✓ réponse rapide ;
- ✓ inflammable et peut être utilisé dans un milieu explosif.

Inconvénients

- ✗ fuites par des joints ;
- ✗ exige la canalisation de retour et la source d'énergie à distance ;
- ✗ difficile à miniaturiser en raison de la haute pression et débit ;
- ✗ cher.

① Vérin à simple effet

Les vérins à simple effet ne sont alimentés en air comprimé que d'un seul côté. Ils ne comportent donc qu'un raccord pneumatique d'alimentation et peuvent fonctionner uniquement dans un sens. L'air doit s'échapper de la chambre du vérin pour qu'il puisse revenir à sa position initiale. La tige du piston peut alors rentrer dans le vérin, sous l'effet d'un ressort incorporé ou d'une force extérieure. L'échappement s'effectue à travers un alésage au niveau de la culasse [20].

Le travail dans un sens : L'arrivée de la pression se fait sur un seul orifice d'alimentation, ce qui entraîne le piston dans un sens, le retour s'effectuant sous l'action d'un ressort de rappel (Fig.I-19).

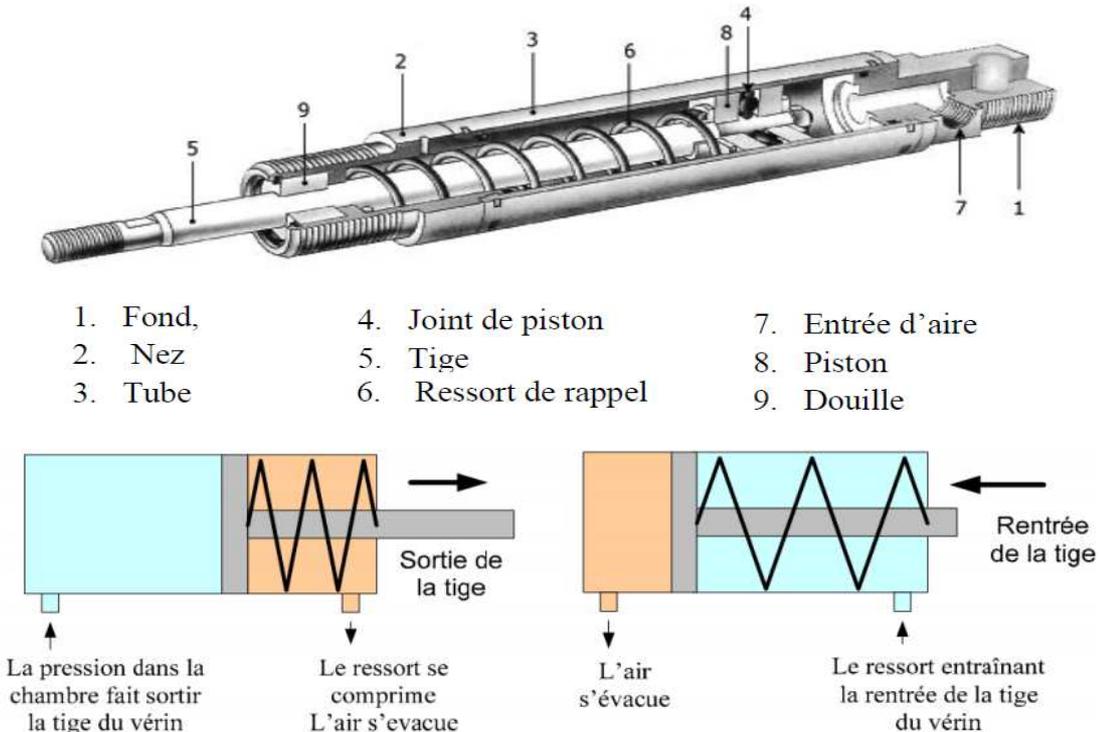


Figure I-19. Vérin à simple effet

② Vérin à double effet

Les vérins à double effet sont alimentés en air comprimé des deux côtés. Ils peuvent donc fonctionner dans les deux sens. La force transmise à la tige de piston pour l'aller est sensiblement supérieure à celle du retour, puisque la surface exposée à l'air comprimé du côté du piston est plus grande que du côté de la tige de piston (Fig. I-20) [20].

Contrairement à la version à simple effet, ce type de vérin comporte deux orifices répartis sur les deux chambres du vérin et peut donc développer un effort en *tirant* et en *poussant*. La majeure partie de ces vérins peuvent être équipés de capteurs de position à détection magnétique à condition que le vérin dispose d'un piston magnétique.

Sur les vérins à double effet, chaque chambre de pression comporte un raccord. L'inversion du mouvement suppose donc l'échappement de l'air qui se trouve dans la chambre correspondante (côté piston ou côté tige de piston).

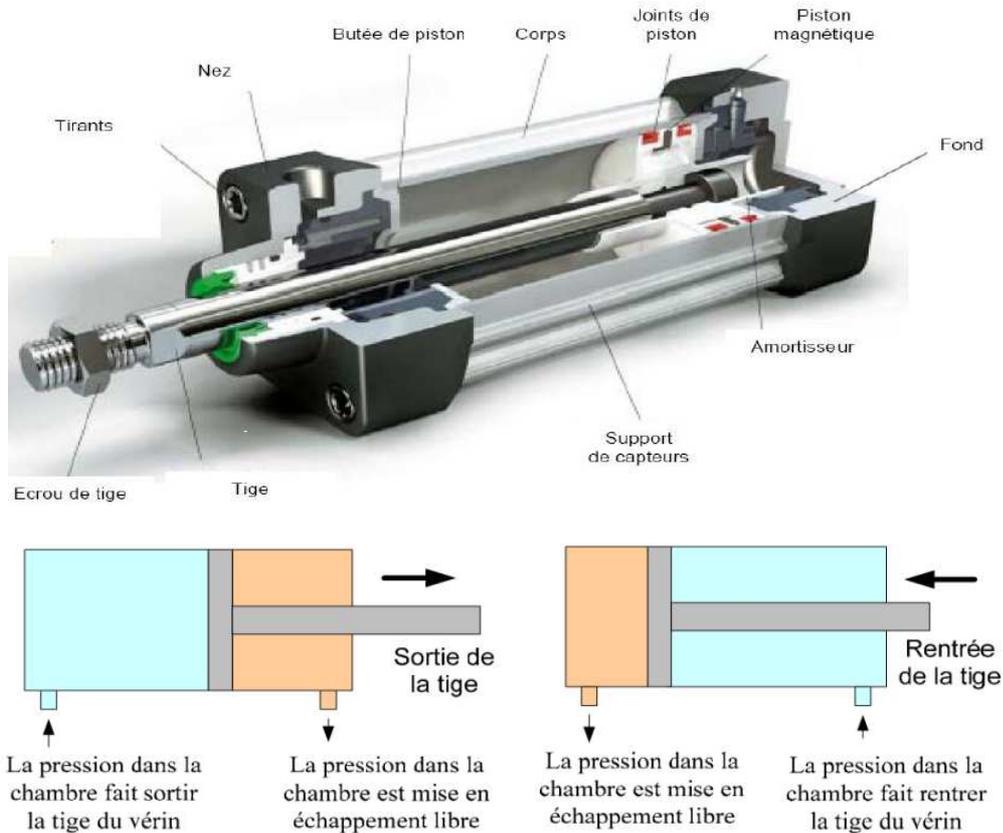


Figure I-20. Vérin à double effet

③ Vérin rotatif

L'énergie du fluide est transformée en mouvement de rotation ; par exemple, vérin double effet entraînant un système pignon-crémaillère. L'angle de rotation peut varier entre 90° et 360° (Fig. I-21) [20].

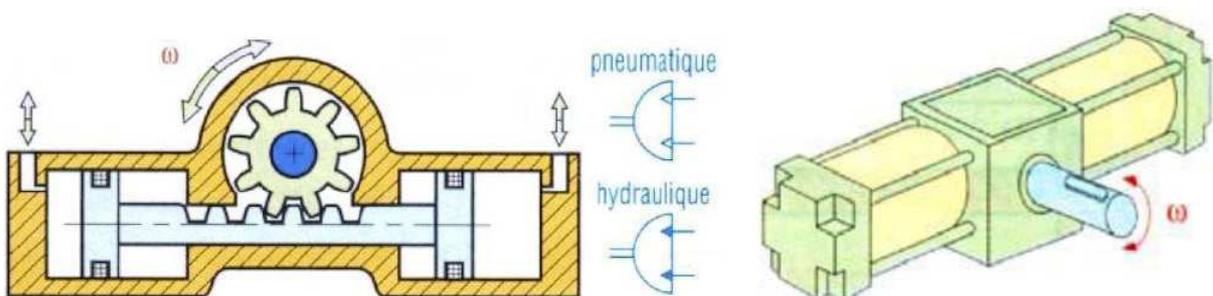


Figure I-21. Vérin rotatif

I.8.3. Effecteurs

Tout élément de la partie opérative qui agit dans le système (cabine d'ascenseur, convoyeur, wagon, ...). Il est situé à la suite de l'actionneur pour finaliser le travail, il produit l'effet attendu. Par exemple : Les chenilles du robot ou la cabine d'ascenseur. L'effecteur est l'élément terminal de la chaîne d'énergie qui agit directement sur la matière d'œuvre traitée par le système, sa fonction technique peut se décomposer en plusieurs actions (Fig. I-22) [8,20] :

- Transmettre l'énergie ;
- Transformer l'énergie ;
- Adapter l'énergie.

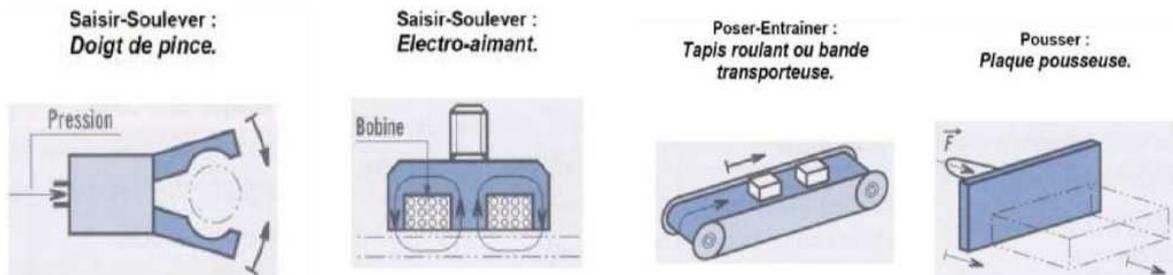


Figure I-22. Effecteurs

I.8.4. Capteurs

Dans de nombreux domaines (industrie, recherche scientifique, services ...), on a besoin de contrôler nombreux paramètres physiques (température, force, position, présence, vitesse ...). Le capteur est un élément indispensable à la mesure de ces grandeurs physiques.

Un capteur est un dispositif qui transforme une grandeur physique (pression, vitesse ...) à qui il est soumis en image électrique (tension, courant ...). Si la grandeur physique désigné par m comme mesurable, l'image électrique est désigné par S , on a $S=f(m)$. (S : grandeur de sortie ou réponse du capteur, m : grandeur d'entrée ou mesurande).



Figure I-23. Capteur

I.8.4.1. Mesurande

C'est une grandeur physique d'entrée du capteur ou la grandeur directe ou intermédiaire que l'on recherche à mesurer. Dans les automatismes industriels on cherche souvent à mesurer : la température, la force, la position, la présence, la vitesse, l'accélération, la pression, le niveau, le débit, le couple, le déplacement, la distance ... [18,19].

I.8.4.2. Grandeur de sortie

Elle généralement de type électrique, elle peut être une tension, un courant ou une impédance. La grandeur de sortie est donc un signal électrique qui pourrait être de type analogique ou numérique. La Figure I-24 montre les différents types de signaux que l'on peut rencontrer [19].

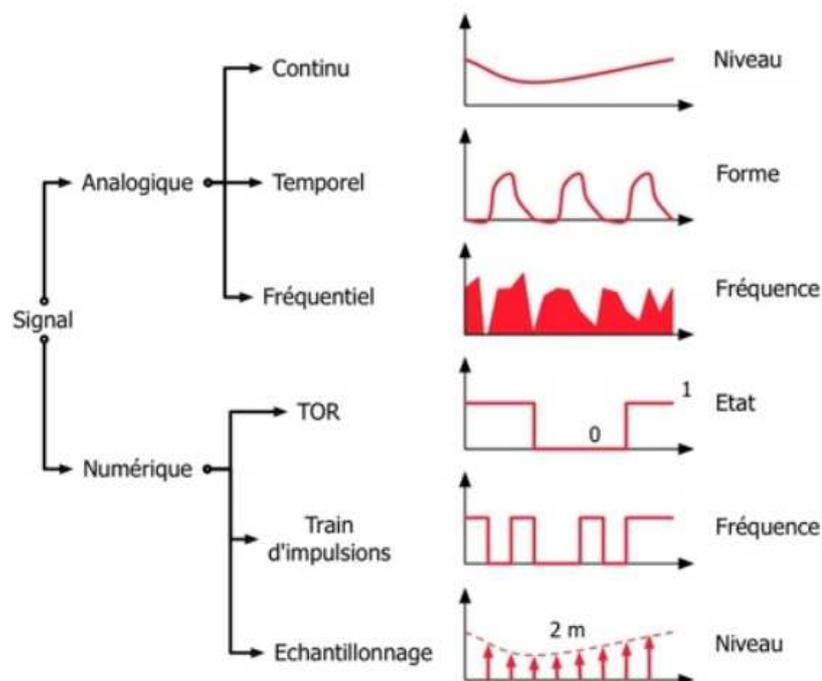


Figure I-24. Différents types des signaux

I.8.4.3. Caractéristiques du capteur

Certains paramètres sont communs à tous les capteurs. Ils caractérisent les contraintes de mise en œuvre et permettent le choix d'un capteur :

- ❑ **L'étendue de la mesure** : c'est la différence entre le plus petit signal détecté et le plus grand perceptible sans risque de destruction pour le capteur.
- ❑ **La sensibilité** : ce paramètre caractérise la capacité du capteur à détecter la plus petite variation de la grandeur mesurée. C'est le rapport entre la variation ΔS du signal électrique de sortie pour une variation donnée Δm de la grandeur physique d'entrée : $S = \Delta S / \Delta m$.
- ❑ **La fidélité** : un capteur est dit fidèle si le signal qu'il délivre en sortie ne varie pas dans le temps pour une série de mesures concernant la même valeur de la grandeur physique d'entrée. Il caractérise l'influence du vieillissement.
- ❑ **Le temps de réponse** : c'est le temps de réaction d'un capteur entre la variation de la grandeur physique qu'il mesure et l'instant où l'information est prise en compte par la partie commande.
- ❑ **La linéarité** : le capteur est linéaire dans une plage déterminée de mesure, si sa sensibilité est indépendante de la valeur du mesurand, alors le signal électrique tout le long de la chaîne de mesure est proportionnel à la variation du mesurand.

I.8.4.4. Différents types de capteurs

Le capteur se présente vu de sa sortie :

- Soit comme un **générateur**, S étant alors une charge, une tension ou un courant, le capteur dit alors **capteur actif**.
- Soit comme une **impédance**, S étant alors une résistance, une inductance ou une capacité. Le capteur est dit alors **capteur passif**.

Il existe différents types de sortie de capteur [19] :

- **Capteur tout ou rien** : la sortie présente un niveau bas et un niveau haut.
- **Capteur analogique** : les informations acquises par le capteur sont délivrées sous forme analogique.
- **Capteur numérique** : les informations acquises par le capteur sont délivrées sous forme numérique et peuvent être traitées directement par un ordinateur.

① Capteur de position à action mécanique

La détection de présence est réalisée lorsque l'objet à détecter entre en contact avec la tête de commande au niveau de son dispositif d'attaque. Le mouvement engendré sur la tête d'attaque provoque la fermeture du contact électrique situé dans le corps du capteur (Fig. I-25).



Figure I-25. Capteur de position à action mécanique

② Capteur de fin de course à action magnétique

Lorsqu'un champ magnétique est dirigé sur la face sensible du capteur, le contact s'établit entre les deux bornes du capteur (Fig. I-26). Ce type de capteur est souvent monté directement sur les corps de vérins en tant que fin de course.



Figure I-26. Capteur à action magnétique

③ Capteur de proximité inductif

Un capteur inductif détecte sans contact tous les objets de matériaux conducteurs. A l'approche d'un objet conducteur, de la face sensible le capteur délivre un signal. Le signal se sortie correspondant à un contact électrique.

Les bobinages de l'oscillateur constituent la face sensible du capteur. A l'avant de celle-ci est créé un champ magnétique alternatif (Fig. I-27).

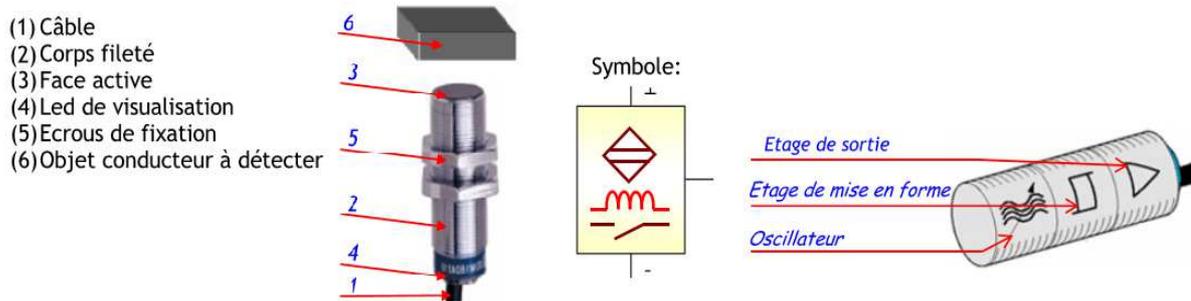


Figure I-27. Capteur de proximité inductif

④ Capteur de proximité capacitif

Un capteur capacitif de proximité détecte sans contact tous les objets de matériaux conducteurs ou isolants. Il se compose principalement d'un oscillateur dont les condensateurs la face sensible. Le signal se sortie correspondant à un contact électrique (Fig. I-28).

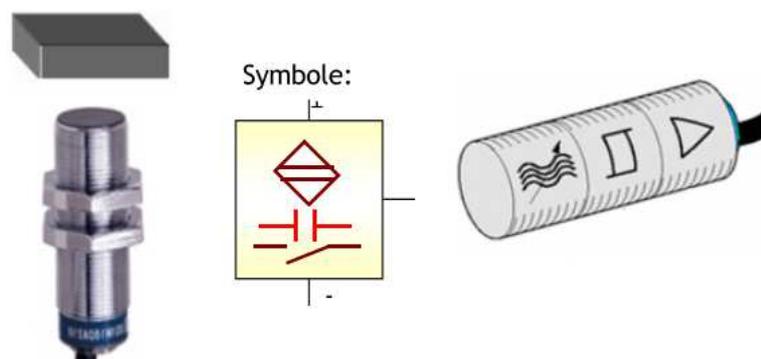


Figure I-28. Capteur de proximité capacitif

④ Capteur de proximité photoélectrique

Un capteur de proximité photoélectrique est constitué d'un émetteur de lumière à diode électroluminescente et un récepteur de lumière à phototransistor qui convertit le signal lumineux en signale électrique. La

détection est obtenue lorsque le faisceau lumineux émis, n'arrive pas au récepteur car l'objet se trouve entre les deux modules (Fig. I-29).

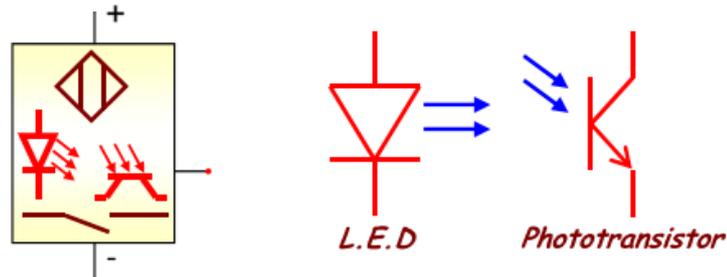


Figure I-29. Capteur de proximité photoélectrique

Il existe trois systèmes de base pour détecter un objet à l'aide d'un émetteur-récepteur de lumière.

A. Le système barrage

L'émetteur et le récepteur sont dans deux boîtiers séparés. L'objet est détecté lorsqu'il interrompt le faisceau lumineux (Fig. I-30).

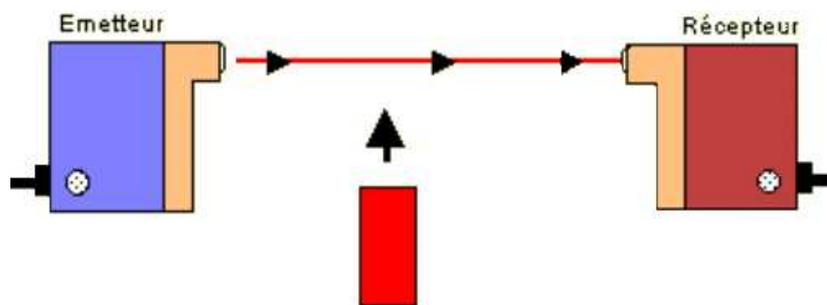


Figure I-30. Système barrage d'un capteur de proximité photoélectrique

B. Le système reflex

L'émetteur et le récepteur sont dans le même boîtier. Le faisceau lumineux émis est envoyé vers le récepteur par réflecteur. La détection se fait par coupure du faisceau (Fig. I-31).

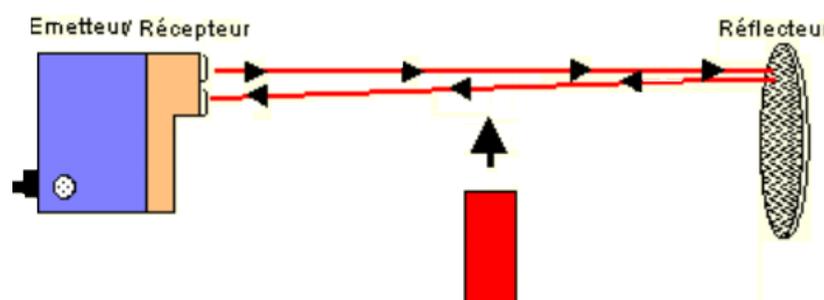


Figure I-31. Système barrage d'un capteur de proximité photoélectrique

C. Le système de proximité

L'émetteur et le récepteur sont dans le même boîtier. Le faisceau lumineux émis est envoyé par l'objet (Fig. I-32).

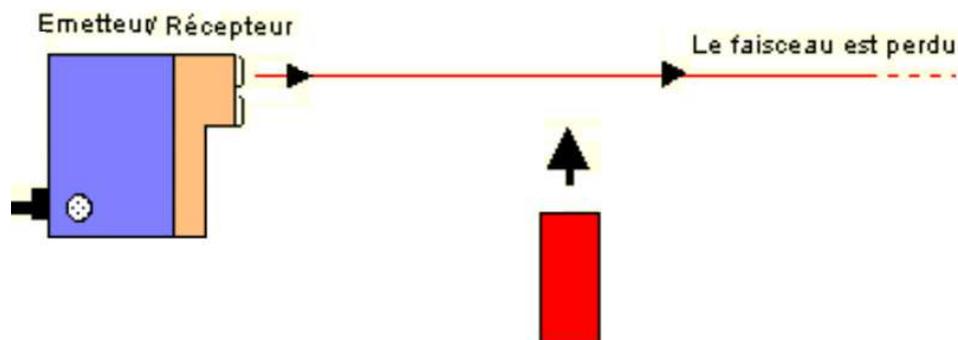


Figure I-32. Système barrage d'un capteur de proximité photoélectrique

I.9. CONCLUSION

J'ai pu décrire dans ce chapitre que dans un système automatisé, l'énergie nécessaire à la transformation du produit est fournie par une source extérieure. Un automate dirige la succession des opérations. L'homme surveille le système et peut dialoguer avec lui par l'intermédiaire d'un pupitre.

Après avoir présenté l'automatisation et la structure des systèmes automatisés, on peut passer à l'étude d'un outil de modélisation qui est le réseau de Pétri.

CHAPITRE II

RÉSEAUX DE PÉTRI (RdP)



« La simplicité est la sophistication suprême.
Oui ne doute pas acquit peu »

Leonardo Da Vinci

CHAPITRE II

RÉSEAUX DE PÉTRI (RdP)

II.1. INTRODUCTION

C'est en 1960 que *Carl Adam Pétri* (mathématicien allemand) définissait les réseaux qui portent depuis son nom. Vingt ans ont passé sans que cet outil de modélisation, réputé puissant dans tous les laboratoires de recherche, parvienne à pénétrer le monde industriel si ce n'est sous la forme voisine, mais conceptuellement différente, du GRAFCET. Cet insuccès est sans doute imputable, pour une part, à l'absence de norme et à une utilisation, dans les écoles d'ingénieurs et les universités, plus tournée vers la recherche que vers l'industrie. Mais la raison essentielle réside sans doute dans le fait que les concepteurs de systèmes automatisés n'exigeaient pas, jusqu'à une date récente, d'outils de modélisation aussi puissants que les réseaux de Pétri. Le formalisme des réseaux de Pétri est un outil permettant l'étude de systèmes dynamiques et discrets. Il permet d'obtenir une représentation mathématique modélisant le système. L'analyse de cette représentation (Réseau de Pétri) peut révéler des caractéristiques importantes du système concernant sa structure et son comportement dynamique.

Dans ce chapitre vous permettra de vous initier à un outil de modélisation très puissant et très répondu notamment dans le monde recherche et développement : RdP (Réseaux de Pétri). Cet outil permet essentiellement la modélisation des systèmes quel que soit leurs domaines d'application (Informatique, Télécommunication, Production, ...).

Dans le présent chapitre, Je vous présenterai dans un premier temps les notions de base de la modélisation par les réseaux de Pétri et les formalismes concernant cet outil. Ensuite je discute les différentes règles d'évolution et les

différentes propriétés de cet outil graphique. Je vous présenterai également les structures de base fréquemment utilisées notamment dans les systèmes de production industrielle. À la fin de ce chapitre je cite les structures particulières des Réseaux de Pétri (RdP).

II.2. CONCEPTS DE BASE DES RÉSEAUX DE PÉTRI

Les réseaux de Pétri sont des outils à la fois graphiques et mathématiques permettant de modéliser le comportement dynamique des systèmes à évènements discrets. Leur représentation graphique permet de visualiser d'une manière naturelle le parallélisme, la synchronisation, le partage de ressources, les choix (conflits), ... etc. Leur représentation mathématique permet d'analyser le modèle pour étudier ses propriétés et de les comparer avec le comportement du système réel [5].

Les RdP ont été développés pour permettre la modélisation de classes importantes de systèmes qui recouvrent des classes de systèmes de production, de systèmes automatisés, de systèmes informatiques et de systèmes de communication, pour n'en citer que quelques-uns, afin de permettre leur conception, leur évaluation et leur amélioration.

II.2.1. Notions de Base

Un *réseau de Pétri (RdP)* est un *graphe biparti* constitué de deux sortes de nœuds : les *places* (représentées par *des ronds*) et les *transitions* (représentées par *des barres*). Le graphe est *orienté*, *des arcs* vont d'une sorte de nœuds à l'autre (jamais de places à places, ou de transitions à transitions directement). Un graphe orienté comporte (Fig. II-1) [21] :

- ① un ensemble fini de *places*, $P = \{P_1, P_2, P_3, \dots, P_i\}$, symbolisées par des *cercles* et représentant des *conditions* qui traduit l'état d'une ressource du système (machine libre, stock vide, convoyeur à l'arrêt, ...)

- ② un ensemble fini de *transitions*, $T = \{T_1, T_2, T_3, \dots, T_j\}$, symbolisées par des *tirets* et représentant l'ensemble des *événements* (les actions se déroulant dans le système) dont l'occurrence provoque la modification de l'état du système ;
- ③ un ensemble fini d'*arcs orientés* qui assurent la *liaison* d'une place vers une transition ou d'une transition vers une place ;

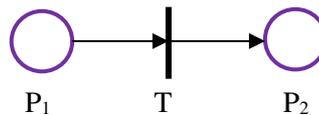


Figure II-1. Liaison orientée du réseau de Pétri

Un graphe orienté est dit *biparti*, c'est-à-dire qu'un arc relie alternativement une place à une transition et une transition à une place. Ainsi les situations suivantes sont interdites (Fig. II-2).



Figure II-2. Liaison orientée interdite dans un réseau de Pétri

- ④ Un nombre M_i de *jetons* dans les places. On appelle M le vecteur des marquages défini par les M_i , $M = \{M_1, M_2, \dots, M_n\}$.

II.2.2. Vocabulaire Associé au Réseau de Pétri

II.2.2.1. Condition

Une condition est la description de l'état d'une ressource du système modélisé [5-9] :

- ☞ une machine est au repos ;
- ☞ une machine est en réparation ;
- ☞ une commande est en attente.

Une condition est un prédicat logique d'un état du système. Elle est soit *vraie*, soit *fausse*. Un état du système peut être décrit comme un ensemble de

conditions. Dans le formalisme des réseaux de Pétri, la condition est modélisée à l'aide d'une place.

II.2.2.2. Événement

Un événement est une action qui se déroule au sein du système et dont la réalisation dépend de l'état du système :

- ☞ début de traitement sur une machine ;
- ☞ panne sur une machine ;
- ☞ début de traitement d'une commande.

Dans le formalisme des réseaux de Pétri, l'événement est modélisé à l'aide d'une transition.

II.2.3. Réseau de Pétri

Un réseau de Pétri est un outil à la fois graphique et mathématique défini par un quadruplet $R = (P, T, \text{Entrée}, \text{Sortie}, M)$, où :

- ❶ P est un ensemble fini de *places* : $P = \{P_1, P_2, P_3, \dots, P_i\}$;
- ❷ T est un ensemble fini de *transitions* : $T = \{T_1, T_2, T_3, \dots, T_j\}$;
 - **Transition source** : Une transition source est une transition qui ne comporte aucune place (Fig. II-3).

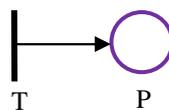


Figure II-3. Transition source

- **Transition puit** : Une transition puit est une transition qui ne comporte aucune place de sortie (Fig. II-4).

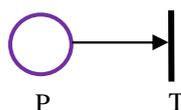


Figure II-4. Transition puit

- ③ Les arcs orientés qui assurent la **liaison** d'une place vers une transition ou bien d'une transition vers une place sont appelées les **poids**. Par convention, lorsque le poids n'est pas précisé sur un arc, alors ce poids vaut 1. Si aucun arc ne relie une transition "T" à une place "P", alors la Sortie $(p,t) = 0$. De même, si aucun arc ne relie une place "P" à une transition "T", alors l'Entrée $(p,t) = 0$.
- ④ **Entrée** (ou Pré) est une application, Entrée : $P \times T \rightarrow \mathbb{N}$, appelée *matrice d'incidence avant*. Elle est notée Pré (p,t) ou Entrée (p,t) ou encore $I(p,t)$, contient la valeur entière "n" associée à l'arc allant de "P" à "T".

- "P" est une place d'entrée de la transition "T" si Pré $(p,t) > 0 \Rightarrow k_1 = \text{Entrée}(p,t) > 0$ (Fig. I-5).

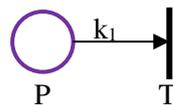


Figure II-5. Place d'entrée de d'une transition

- "T" est une transition d'entrée d'une place "P" si Post $(p,t) > 0 \Rightarrow i_1 = \text{Sortie}(p,t) > 0$ (Fig. I-6).

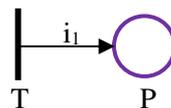


Figure II-6. Transition d'entrée d'une place

- ⑤ **Sortie** (ou Post) est une application, Sortie : $P \times T \rightarrow \mathbb{N}$, appelée *matrice d'incidence arrière*. Elle est notée Post (p,t) ou Sortie (p,t) ou encore $O(p,t)$ contient la valeur entière "n" associée à l'arc allant de "T" à "P".

- "P" est une place de sortie de la transition "T" si Post $(p,t) > 0 \Rightarrow k_2 = \text{Sortie}(p,t) > 0$ (Fig. I-7).

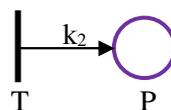


Figure II-7. Place de sortie d'une transition

- "T" est une transition de sortie d'une place "P" si $\text{Pré}(p,t) > 0 \Rightarrow i_2 = \text{Sortie}(p,t) > 0$ (Fig. I-8).

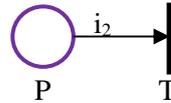


Figure II-8. Transition de sortie d'une place

II.2.4. Marquage d'un Réseau de Pétri

Le marquage d'un réseau de Pétri est une fonction M de P dans \mathbb{N}^+ (ensemble des entiers non négatifs). On représente graphiquement cette fonction M en plaçant des *marques* ou *jetons (points noirs)* dans les places du réseau de Pétri. Ainsi, une place peut être soit marquée (elle possède alors une ou plusieurs marques) soit non marquée (elle est alors vide). Ces marques ou jetons sont *indivisibles* et *infusionnables* (Fig. I-9). Lorsque la place représente une condition logique (machine à l'arrêt, convoyeur en panne), la présence d'un jeton indique que cette condition est vraie ; fausse dans le cas contraire [21].

Une place donc peut représenter une ressource du système (un stock par exemple), elle peut contenir plusieurs jetons (dans l'exemple du stock, le nombre de jetons peut indiquer le nombre de pièces stockés).

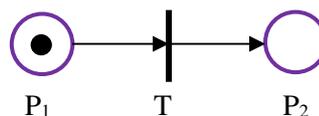


Figure II-9. Place marquée et autre vide

Les jetons circulent dans les places selon certaines règles (définies dans le § II-3). Cette circulation symbolise l'évolution dynamique du système. Le marquage initial, M_0 , d'un RdP correspond à la distribution initiale des jetons dans chacune des places du RdP, qui précise l'état initial du système. Dans ce cas, on parle du RdP marqué par opposition à un RdP non marqué, c'est-à-dire pour lequel le marquage initial n'est pas précisé.

On note $M(p)$ le nombre de jetons contenu dans la place "P" pour le marquage M. Dans le réseau RdP de la Figure II-10, l'état initial correspond à la répartition des jetons de la façon suivante : les places 1 et 3 sont respectivement marquées par 3 et 1 jeton, la place 2 est vide, alors le marquage initial : $M_0 = [M_0(p_1), M_0(p_2), M_0(p_3)]^T = [3, 0, 1]^T$. Un marquage peut être représenté sous forme d'un vecteur colonne : $M_0 = [M_0(p_1), M_0(p_2), M_0(p_3)]^T = \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix}$.

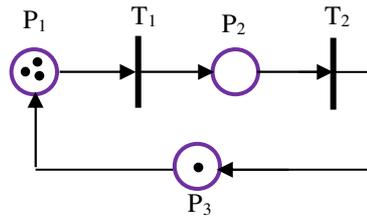


Figure II-10. Marquage initial : $M_0 = [3, 0, 1]^T$

Un jeton peut avoir plusieurs significations en fonction de la place dans lequel il se trouve.

Par exemple dans le RdP de la Figure II-10 on considère que :

- P1 représente un stock : le nombre de jetons en P1 indique le nombre de pièces stockées ;
- P2 représente une machine en cours de traitement : un jeton en P2 indique que la machine traite une pièce ;
- P3 représente une machine libre : un jeton en P3 indique que la machine est libre.

Les jetons d'une place n'ont pas d'identité individuelle, ils sont indiscernables. Ces jetons sont consommés et produits par les événements du système.

La présentation faite des réseaux de Pétri est une représentation graphique, alors qu'il est possible d'en faire une représentation mathématique grâce à l'algèbre linéaire. Cette notation permet de représenter un RdP de manière plus formelle mais moins lisible que la représentation graphique.

Ainsi, **Pré** et **Post** sont représentés par des matrices à **n** lignes (nombre de places), **m** colonnes (nombre de transitions).

II.2.4.1. Exercice d'application

Pour le réseau de Pétri de la Figure II-II :

- Indiquer le marquage initial ;
- Etablir la matrice d'entrée (d'incidence avant) E ;
- Etablir la matrice de sortie (d'incidence arrière) S.

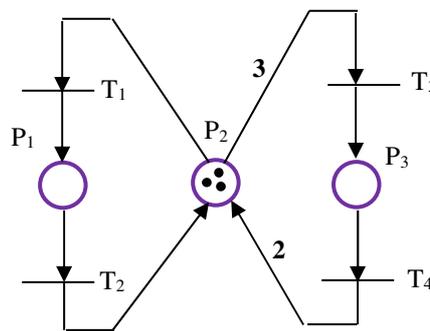


Figure II-II. Réseau de Pétri

II.2.4.2. Solution

Le marquage initial est : $M_0 = [M_0(p_1), M_0(p_2), M_0(p_3)]^T = \begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix}$

La matrice d'entrée (d'incidence avant) E :

T_1	T_2	T_3	T_4	\leftarrow
0	1	0	0	P_1
1	0	3	0	P_2
0	0	0	1	P_3

La matrice de sortie (d'incidence arrière) S :

T_1	T_2	T_3	T_4	\rightarrow
1	0	0	0	P_1
0	1	0	2	P_2
0	0	1	0	P_3

II.3. ÉVOLUTION DES RÉSEAUX DE PÉTRI

II.3.1. Validation d'une Transition

L'évolution l'état du réseau de Pétri correspond à une évolution du marquage. Les jetons, qui matérialisent l'état du réseau à un instant donné, peuvent passer d'une place à l'autre par franchissement ou tir d'une transition. Une transition est *validée* (on dit aussi *sensibilisée* ou *franchissable* ou encore *tirable*) si et seulement si toutes les places d'entrée possèdent *au moins* un nombre de marques égal au poids de l'arc correspondant. (Fig. II-12) [17].

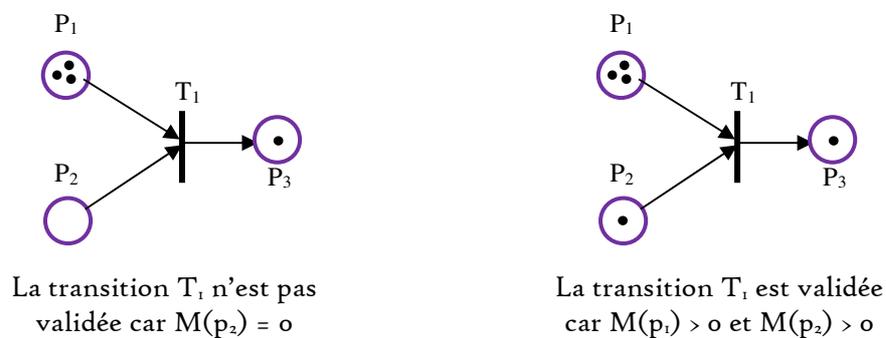


Figure II-12. Transition est validée

II.3.2. Franchissement d'une Transition

Pour qu'une transition soit franchissable, il faut et il suffit que l'on trouve dans toutes les places immédiatement amont à cette transition, le nombre de marques correspondant au poids des arcs reliant respectivement chacune de ces places à cette transition. Une transition peut être *franchie* si et seulement si elle est *sensibilisée* (Fig. II-13) [11].



Figure II-13. Franchissement d'une transition

Le *franchissement* d'une transition ou le *tir* d'une transition, consiste à enlever un jeton dans chacune des places d'entrée de la transition et à ajouter un jeton dans chacune des places de sortie de la même transition.

Le franchissement de la transition T_1 à partir du marquage M_1 conduit au marquage M_2 , se note : $M_1 (T_1 \rightarrow M_2)$.

La Figure II-14 montre un autre exemple de franchissement de la transition T_1 à partir du marquage $M_0 = [2 \ 1 \ 1 \ 1]$ qui conduit le marquage $M_1 = [1 \ 0 \ 2 \ 2]$.



Figure II-14. Franchissement d'une transition

Les conséquences du franchissement (*toujours supposé instantané*) de la transition sont les suivantes :

- ☞ On *prélevé dans chacune des places d'entrée* un nombre de jetons égal au poids respectif de l'arc reliant la place à la transition.
- ☞ On *dépose dans chacune des places de sortie* un nombre de jetons égal au poids respectif de l'arc reliant la transition à la place.

II.3.3. Franchissement d'une Transition Source

Une transition source est une transition qui ne comporte aucune place d'entrée ; c'est une transition toujours franchissable et le franchissement a lieu lorsque l'événement associé se produit.

Dans le réseau de Pétri de la Figure II-15, la transition T_1 représente une transition source.

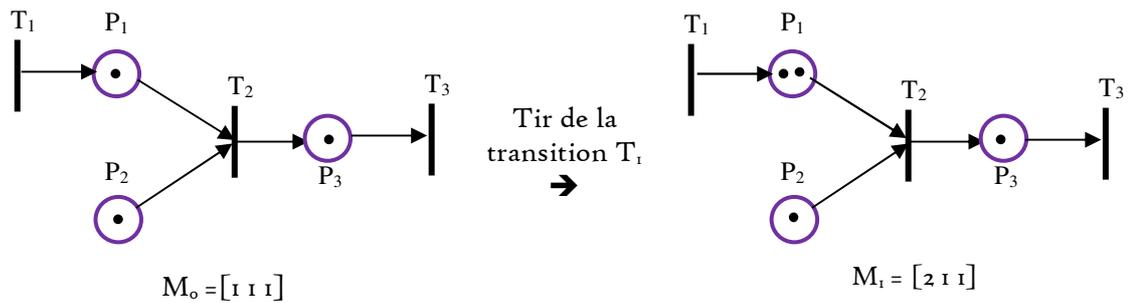


Figure II-15. Franchissement d'une transition source

II.3.4. Franchissement d'une Transition Puit

Une transition puits est une transition qui ne comporte aucune place de sortie, le franchissement d'une transition puits enlève des jetons de toutes les places d'entrée de la transition (Fig. II-16) (T_3 est une transition puits).

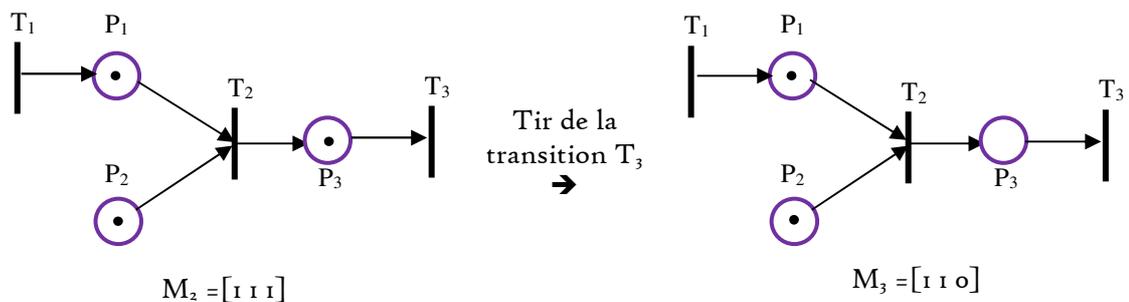


Figure II-16. Franchissement d'une transition puits

II.3.5. Franchissement d'une Transition avec Plusieurs Ressources

De manière formelle, une transition est dite sensibilisée si les places situées en amont (places d'entrée) possèdent chacune un nombre de jetons supérieur ou égal au poids des arcs joignant ces places à la transition (Fig. II-18).

Franchir ou tirer une transition consiste à enlever de chaque place d'entrée un nombre de jetons égal au poids de l'arc joignant cette place à la transition et à déposer, dans chaque place aval ou place de sortie, un nombre de jetons égal au poids de l'arc joignant la transition à chacune de ces places. Le franchissement des transitions et les modifications du marquage qu'il entraîne permettent d'analyser la dynamique du système modélisé [21].

La Figure II-17 montre une transition non tirable.

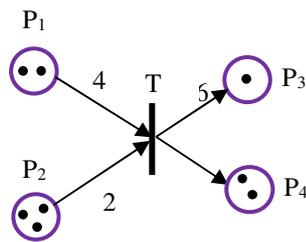


Figure II-17. Transition T non tirable

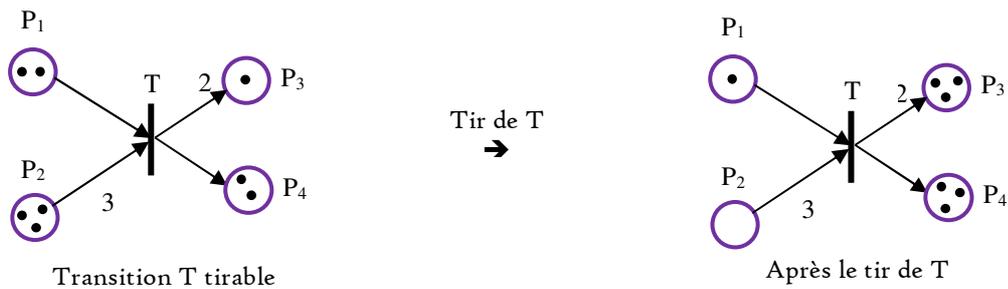


Figure II-18. Transition T tirable

II.3.5.1. Exemple concret

Une molécule d'eau est constituée de deux atomes d'hydrogène H_2 et d'un atome d'oxygène O . Cette réaction chimique peut être modélisée par un RdP de la Figure II-19.



Figure II-19. RdP de la réaction chimique d'une molécule H_2O

II.3.5.2. Remarques

- ⊙ Lorsqu'une transition est validée, cela n'implique pas qu'elle sera immédiatement franchie ; cela ne représente qu'une possibilité de franchissement ou d'évolution du RdP.
- ⊙ Pour les RdP, il y a un seul franchissement à la fois : le réseau ne peut évoluer que par franchissement d'une seule transition à la fois, transition choisie parmi toutes celles qui sont validées à cet instant.
- ⊙ Le franchissement d'une transition est indivisible et de durée nulle.

- ⊙ En utilisant la notion d'algèbre linéaire, le franchissement (tir) d'une transition T_j ne peut s'effectuer que si le marquage de chacune des places P_i directement en amont de cette transition est tel que :

$$M(P_i) > \text{Pré}(P_i, T_j) \text{ (condition nécessaire)}$$

- ⊙ Le franchissement (tir) de T_j consiste à retirer $\text{Pré}(P_i, T_j)$ jetons dans chacune des places directement en amont de T_j et à ajouter $\text{Post}(P_k, T_j)$ jetons dans chacune des places P_k directement en aval de T_j . Le franchissement de T_j conduit au nouveau marquage M' tel que :

$$\forall p_i \in P, M'(P_i) = M(P_i) - \text{Pré}(P_i, T_j) + \text{Post}(P_i, T_j)$$

- ⊙ La différence des marquages M et M' tels que $M \xrightarrow{T_j} M'$ est égale à $\text{Post} - \text{Pré}$. On introduit la matrice W , dite *matrice d'incidence* telle que :

$$\forall p_i \in P, \forall t_j \in T : W(P_i, T_j) = \text{Post}(P_i, T_j) - \text{Pré}(P_i, T_j)$$

- ⊙ La $j^{\text{ème}}$ colonne de la matrice d'incidence W donne donc la variation du nombre de jetons dans chaque place lors du franchissement de la transition T_j . On peut par conséquent établir le nouveau marquage M' à partir du marquage M par franchissement de la transition T_j en appliquant la règle suivante :

$$M' = M + W \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow \text{la transition } T_j$$

II.3.5.3. Exercice d'application

Pour le réseau de Pétri de la Figure II-20 :

- Indiquer le marquage initial ;
- Etablir la matrice d'entrée (d'incidence avant) E ;
- Etablir la matrice de sortie (d'incidence arrière) S ;

- Etablir la matrice d'incidence W ;
- Etablir les marquages en utilisant l'algèbre linéaire.

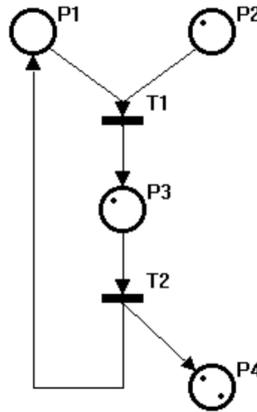


Figure II-20. Réseau de Pétri (RdP)

Le marquage initial est : $M_0 = [M_0(p_1), M_0(p_2), M_0(p_3), M_0(p_4)] = [0 \ 1 \ 1 \ 2]$

La matrice d'entrée (d'incidence avant) E :

T_1	T_2	\leftarrow
1	0	P_1
1	0	P_2
0	1	P_3
0	0	P_4

La matrice de sortie (d'incidence arrière) S :

T_1	T_2	\rightarrow
0	1	P_1
0	0	P_2
1	0	P_3
0	1	P_4

La matrice d'incidence $W=S-E$:

T_1	T_2	
-1	1	P_1
-1	0	P_2
1	-1	P_3
0	1	P_4

Le franchissement de la transition T_1 enlève un jeton à P_1 et un jeton à P_2 et ajoute un jeton à P_3 . Le franchissement de la transition T_2 enlève un jeton à P_3 et ajoute un jeton à P_1 et un jeton à P_4 . Un zéro dans la matrice W correspondant à une transition T_i et une place P_j s'interprète comme l'absence d'un arc reliant ces deux nœuds. Mais attention, cette dernière interprétation ne concerne que les RdP dits purs.

Par ailleurs, la matrice d'incidence ne donne aucune indication sur la possibilité de franchir une transition puisqu'elle est indépendante du marquage.

II.4. MARQUAGES ACCESSIBLES

Pour définir l'état d'un système modélisé par un réseau de Pétri, il est nécessaire de compléter le réseau de Pétri par un marquage. Ce marquage consiste à disposer un nombre entier (positif ou nul) de jetons dans chaque place du réseau de Pétri [5,16].

L'ensemble des marquages accessibles d'un RdP à partir d'un marquage initial donné, correspond à l'ensemble des marquages atteint après franchissement de transitions sensibilisées les unes après les autres ; ce qui correspond à toutes les situations possibles du RdP au cours de son évolution à partir du marquage initial.

On appelle marquage M d'un RdP le vecteur dont les composantes représentent le nombre de jetons dans chaque place : la $i^{\text{ème}}$ composante de ce vecteur correspond au nombre de jetons dans la $i^{\text{ème}}$ place. Il indique à un instant donné l'état du RdP. On note le marquage initial, M_0 , le marquage à l'instant initial ($t = 0$).

L'ensemble des marquages accessibles, $M(R; M_0)$, pour le réseau de Pétri de la Figure II-21 est : $M(R; M_0) = \{M_0, M_1, M_2, M_3, M_4\}$.

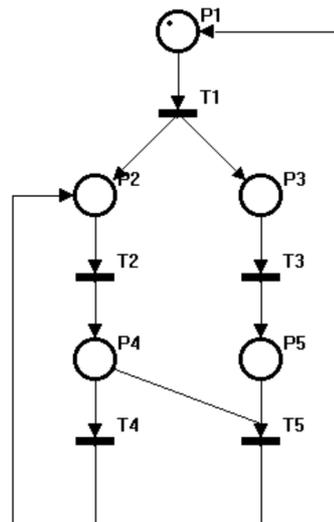


Figure II-21. Réseau de Pétri (RdP)

Etant donné un marquage, une transition est sensibilisée (ou déclenchable ou franchissable) si dans chacune de ses place d'entrée il y a au moins le nombre de jetons indiqué par le poids de l'arc correspondant [15,21].

Plusieurs transitions peuvent être sensibilisées dans un même marquage, l'évolution du réseau dépend du choix de la transition à déclencher. Ainsi, à partir d'un état le réseau peut évoluer selon différents scénarios, en fonction des choix faits à chaque étape. L'idée et l'intérêt du réseau de Pétri est précisément de représenter potentiellement toutes les évolutions possibles d'un système puis de calculer des propriétés qui et restent valables quelle que soit l'évolution.

Le marquage d'un RdP une distribution de jetons sur les places. Le marquage est une application vectorielle $M : P \rightarrow \mathbb{N}$ donnant pour chaque place le nombre de jetons qu'elle contient, la dimension du vecteur de marquage $M(p)$ est égale au nombre de places.

L'évolution du RdP est représentée dans la Figure II-22 avec les marquages représentés sous la forme de vecteurs colonnes.

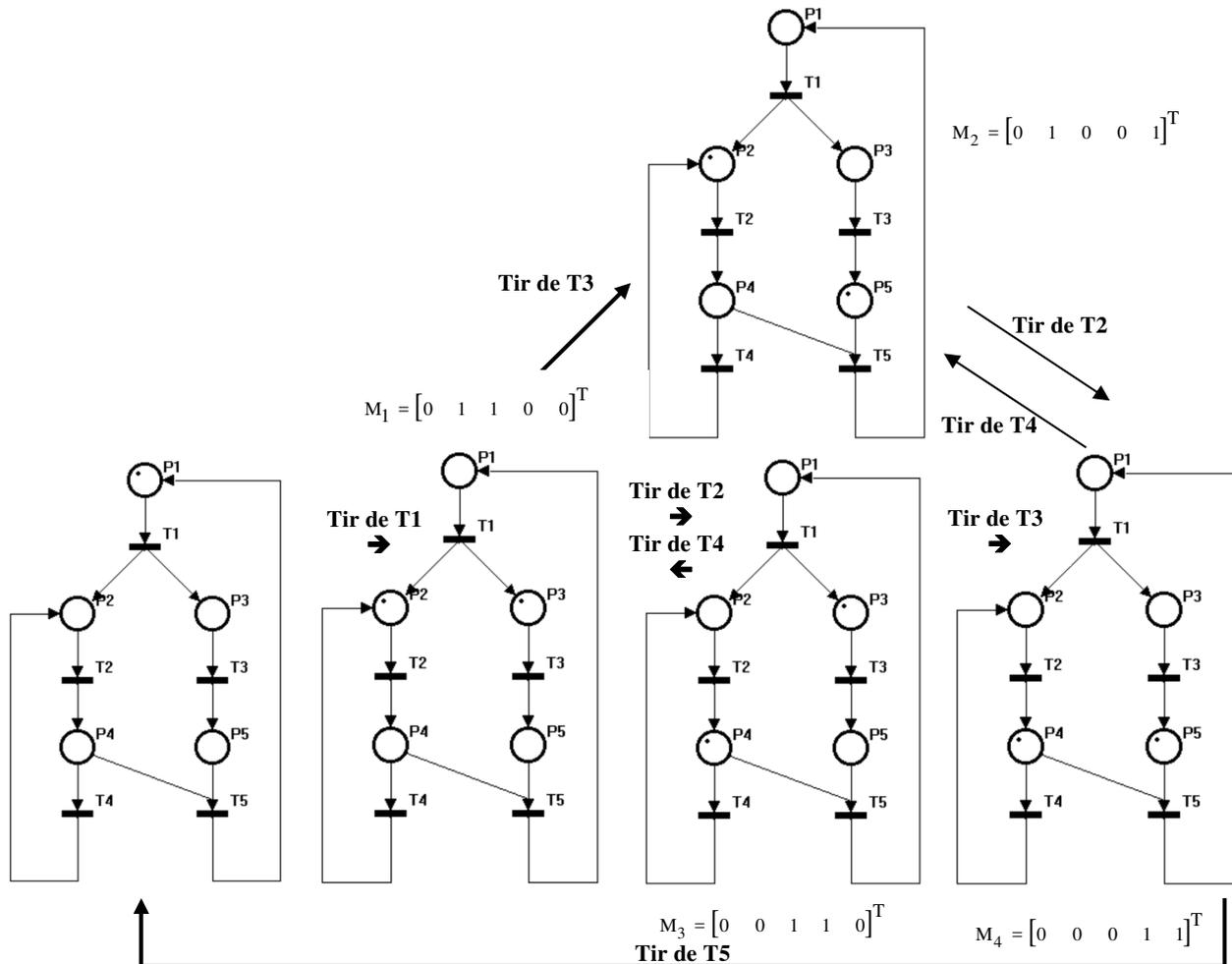


Figure II-22. Evolution du réseau de Pétri (RdP)

L'évolution du RdP peut être représenté sous la forme d'un graphe des marquages $GM(R; M_0)$, dont les sommets correspondent aux marquages accessibles (Fig. II-23).

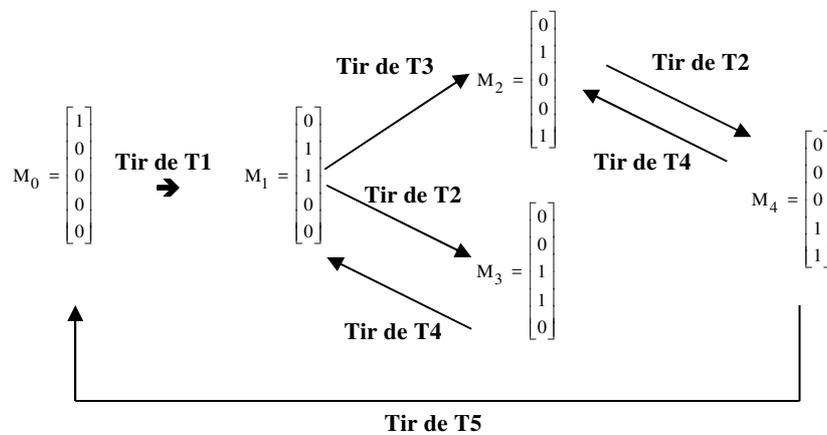


Figure II-23. Graphe des marquages

Enfin, il est possible de représenter un graphe de marquage sous forme d'un organigramme (Fig. II-24).

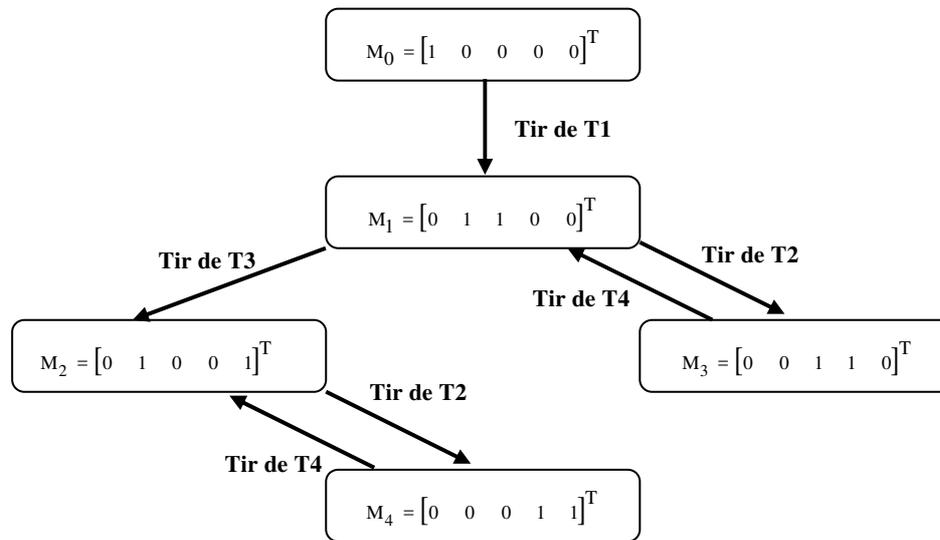


Figure II-24. Graphe des marquages

Le déclenchement (ou franchissement) d'une transition sensibilisée consomme des jetons de ses places d'entrée et ajoute des jetons dans ses places de sortie. Le nombre de jetons consommés et produits correspond aux poids des arcs. À la suite d'un déclenchement on obtient un nouveau marquage, qui n'a pas forcément le même nombre total de jetons que le précédent.

L'évolution temporelle d'un RdP peut être décrite par un graphe de marquage représentant l'ensemble des marquages accessibles et d'arcs correspondant aux franchissements des transitions faisant passer d'un marquage à l'autre pour un marquage initial M_0 .

Cette représentation permet de déterminer certaines propriétés d'un graphe. Par exemple si le graphe présente une zone non bouclé, cette partie du marquage une fois atteinte constitue un arrêt de l'évolution du RdP et celui-ci sera déclaré avec blocage.

II.5. SÉQUENCE DE FRANCHISSEMENT

La séquence est une situation dans laquelle une série de transitions doivent obligatoirement se dérouler l'une après l'autre. Le franchissement successif de transitions (sensibilisées) dans un ordre donné à partir d'un marquage donné constitue une séquence de franchissements [12,21].

Un réseau de Pétri est un graphe d'états si et seulement si toute transition a exactement une place d'entrée et une place de sortie. Si l'ensemble des places du graphe d'états ne contient qu'un seul jeton, on retrouve le graphe d'état classique introduit pour la description et la conception des machines logiques séquentielles. Il permet de visualiser des phénomènes de concurrence (décision), mais pas de synchronisation. Un réseau de Pétri est un graphe d'événements si et seulement si toute place a exactement une transition d'entrée et une transition de sortie. Aussi, un graphe d'événements peut modéliser des phénomènes de synchronisation, mais pas de concurrence.

Pour le RdP de la Figure II-21, les séquences de franchissement peuvent être notées différemment selon les auteurs. Par exemple, la séquence qui conduit du marquage M_0 au marquage M_4 peut être notée :

- ▶ $S_1 = T_1 T_2 T_4$ ou $M_0(T_1 T_2 T_4 \rightarrow M_4, M_0(S_1 \rightarrow M_4))$, ou encore $M_0[S_1 > M_4$.
- ▶ $S_2 = T_1 T_3 T_5$ ou $M_0(T_1 T_3 T_5 \rightarrow M_0, M_0(S_2 \rightarrow M_5))$, ou encore $M_0[S_2 > M_5$.

S_1 et S_2 sont deux séquences possibles à partir du marquage initial M_0 ; chacun des marquages est atteint après le tir d'une transition permet le tir de la transition suivante dans la séquence.

Toute séquence de franchissement considérée à partir de M_0 doit débiter par T_1 . La séquence $S = T_2 T_4 T_3$, n'est pas réalisable à partir de M_0 . De même, toute séquence de franchissement considérée à partir de M_2 doit débiter par T_2 . La séquence $S = T_3 T_4 T_2$, n'est pas réalisable à partir de M_2 .

On définit S , vecteur caractéristique de la séquence S en précisant pour chaque transition le nombre de fois où la transition est franchie dans la séquence. L'équation fondamentale d'un marquage M_k à partir d'un marquage M_i est : $M_k = M_i + W.S^T$.

Cette équation permet de calculer en une seule opération, sans parcourir le réseau, le marquage obtenu après le franchissement de la séquence complète. Les résultats de l'équation fondamentale, même s'ils sont toujours calculables, n'ont de sens que si la séquence représentée par S est effectivement réalisable.

A titre d'exemple, on considère la séquence de franchissement $S = T_1 T_2 T_4 T_2$ du RdP précédent qui conduit au marquage M_1 . Le vecteur caractéristique $S = [1 \ 2 \ 0 \ 1 \ 0]^T$.

On commence par établir la matrice d'incidence W comme suit :

$$W = S - E = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix}$$

Le marquage obtenu à partir du marquage initial M_0 est alors :

$$M = M_0 + W.S^T = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Il correspond bien au marquage M_1 .

L'obtention d'un marquage à partir d'un autre ne signifie pas forcément l'unicité du vecteur caractéristique. En effet, si on reprend l'exemple ci-dessous, on aboutit à partir du marquage initial M_0 au marquage M_3 par franchissement de la séquence $T_1 T_3$ ($S = [1 \ 0 \ 1 \ 0 \ 0]^T$) ou par franchissement de la séquence $T_1 T_3 T_2 T_4$ ($S = [1 \ 1 \ 1 \ 1 \ 0]^T$).

Une séquence est dite répétitive si à partir d'un marquage initial, on aboutit à un marquage final identique au marquage initial.

II.6. PROPRIÉTÉS DES RÉSEAUX DE PÉTRI

Les réseaux de Pétri permettent de modéliser un certain nombre de comportements importants dans les systèmes tel que : le parallélisme, la synchronisation, le partage de ressources, la mémorisation, la lecture d'information et la limitation de la capacité de stockage [16,18].

II.6.1. Réseau de Pétri avec et sans Conflit

Un réseau de Pétri est dit sans conflit si et seulement si toute place a au plus une transition de sortie. Un conflit (structurel) correspond à l'existence d'une place P_i qui a au moins deux transitions de sortie T_i et T_j .

Un RdP avec conflit est un réseau qui possède donc une place avec au moins deux transitions de sorties. Cette situation du conflit correspond à la concurrence à la consommation des jetons à une place.

Dans le réseau de Pétri de la Figure II-25 (a), T_2 et T_3 sont en conflit structurel potentiel pour le partage des jetons de la place P_1 . Quand la place P_1 contient un jeton, les transitions T_2 et T_3 sont franchissables. Seule une des deux transitions peut être franchie : il est nécessaire de prendre une décision pour savoir laquelle des deux le sera effectivement.

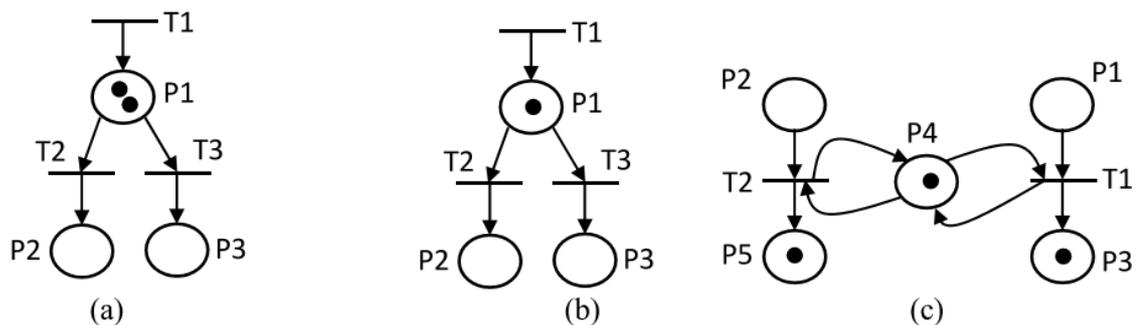


Figure II-25. Réseau de Pétri : (a) sans conflit effectif, (b) avec conflit effectif, (c) persistant

Ce conflit structurel doit être arbitré par une règle de priorité quelconque lorsque le conflit est effectif, c'est-à-dire lorsque les transitions aval en compétition pourraient être activées.

Ne pas arbitrer un conflit structurel fait que le comportement n'est pas entièrement spécifié. Algébriquement, on peut détecter la présence d'un conflit structurel. Deux transitions T_i et T_j sont en conflit structurel si :

$$\exists P_k \text{ tel que } \text{Pré}(P_k, T_i) \times \text{Pré}(P_k, T_j) \neq 0$$

La multiplication est effectuée terme à terme.

Le parallélisme représente la possibilité que plusieurs processus évoluent simultanément au sein du même système. On peut provoquer le départ simultané de l'évolution de deux processus à l'aide d'une transition ayant plusieurs places de sortie. Il est ensuite possible de synchroniser l'achèvement des deux processus.

Dans le réseau de Pétri de la Figure II-26, Lorsque la place P_3 est marquée, les transitions T_2 et T_4 sont en conflit structurel.

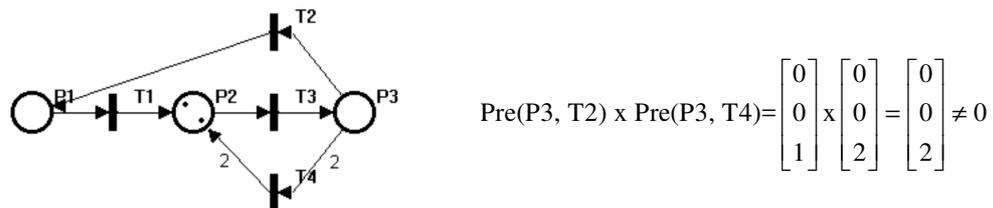


Figure II-26. RdP avec conflit structurel

II.6.2. Réseau de Pétri à Choix Libre

Un RdP est à choix libre ou simple est un réseau dans lequel pour tout conflit $[P_i, \{T_1, T_2, \dots, T_n\}]$ aucune des transitions T_1, T_2, \dots, T_n ne possède aucune autre place d'entrée que P_i (Fig. II-27).

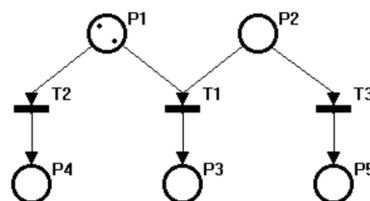


Figure II-27. RdP avec conflit et à choix non libre

II.6.3. RdP en Conflit Relatif au Marquage ou Transitions en Conflit Effectif

Dans un RdP, deux transitions T_i et T_j sont en conflit effectif pour un marquage M si et seulement si :

$$M \geq \text{Pré}(\cdot; T_i), M \geq \text{Pré}(\cdot; T_j)$$

$$\text{et } M \text{ n'est pas supérieur ou égal } \text{Pré}(\cdot; T_i) + \text{Pré}(\cdot; T_j)$$

Cela veut dire qu'il y a assez de jetons pour que l'une des deux transitions T_i ou T_j (exclusivement) soit franchie mais pas les deux à la fois. Dans le réseau de Pétri de la Figure II-28, les transitions T_1 et T_3 sont en conflit effectif pour le marquage initial $M_0 = [0 \ 3 \ 0]^T$.

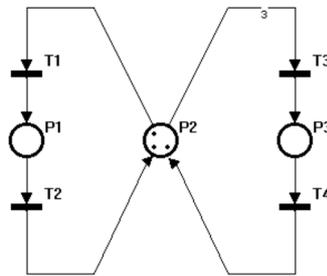


Figure II-28. RdP en conflit effectif

Il y a assez de jetons dans la place P_2 pour le franchissement de T_1 ou bien pour le franchissement de T_3 mais pas pour les deux à la fois. Par contre pour le marquage : $M_0 = [0 \ 4 \ 0]^T$, ces transitions sont toujours en conflit structurel mais elles ne sont plus en conflit effectif.

II.6.4. Réseau de Pétri Pur

Un RdP pur est un réseau dans lequel il n'existe pas de transition ayant une place d'entrée qui soit à la fois place de sortie de cette transition (boucle élémentaire). Dans le cas contraire, on parle de RdP impur comme montre la Figure II-29. La transition T_2 admet à la fois la place P_3 comme place d'entrée et place de sortie.

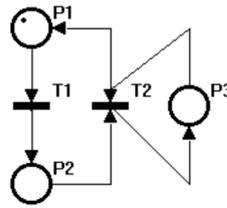


Figure II-29. RdP impur

La matrice d'incidence W est liée à la structure du RdP pur (absence d'une boucle élémentaire) alors il est possible de reconstruire ce RdP à partir de sa matrice d'incidence. En effet le RdP de la Figure II-30 est un RdP impur à cause de la présence d'une boucle élémentaire (P_3, T_5) dont la matrice d'incidence W est donnée ci-contre (donc, la matrice d'incidence W ne contient donc pas toujours toute l'information sur la structure du RdP).

$$W = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 1 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 \end{bmatrix} \rightarrow \text{Mauvaise interprétation de ce zéro}$$

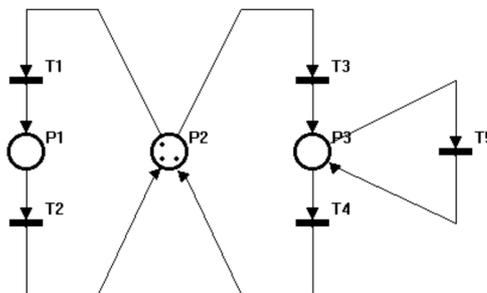
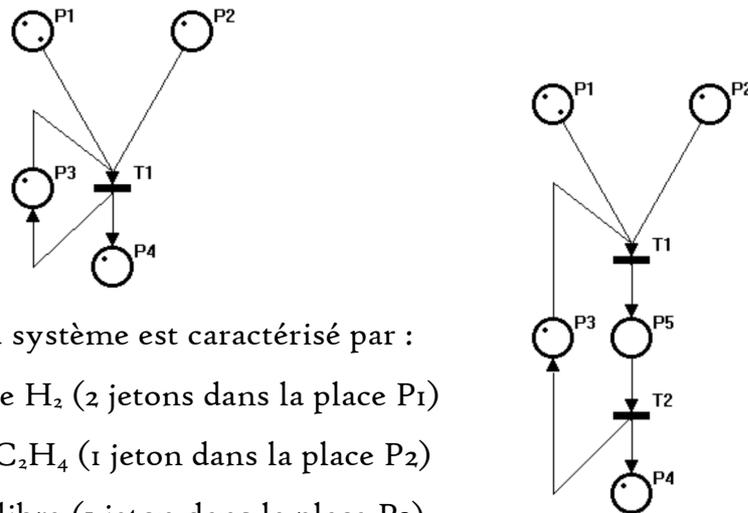


Figure II-30. RdP impur

Par ailleurs, la notion d'impureté n'est pas forcément une propriété contraignante. Elle permet de simplifier la structure d'un RdP.

L'exemple de la Figure II-31 constitue une bonne illustration de ce propos. En effet, on considère la réaction chimique suivante qui doit se dérouler en présence d'un catalyseur (platine) : $H_2 + C_2H_4 \rightarrow C_2H_6$.



- L'état initial du système est caractérisé par :
- deux unités de H_2 (2 jetons dans la place P_1)
 - une unité de C_2H_4 (1 jeton dans la place P_2)
 - le platine est libre (1 jeton dans la place P_3)
 - une unité de C_2H_6 (1 jeton dans la place P_4)

Figure II-30. RdP de la réaction chimique du platine

Le RdP représentant cette réaction chimique est donné par la Figure II-30 (gauche). L'état du système va évoluer quand la réaction chimique se produit (franchissement de la transition T_1). La place P_3 doit être impérativement en entrée de la transition T_1 . Cependant, quand la réaction chimique se produit, le platine étant un catalyseur, il n'est pas consommé. Par suite, la place P_3 doit être en sortie de la transition T_1 de façon à restituer la ressource après que la réaction se soit produite.

Ce RdP n'est pas pur puisque la place P_3 est à la fois place en entrée et en sortie de la transition T_1 . Par contre, une description plus fine de la réaction chimique donne lieu au RdP pur donné par la figure à droite.

Cependant, il est plus complexe. Par description plus fine, on entend que la réaction chimique n'est pas instantanée et les transitions T_2 et T_3 modélisent le début et la fin de cette réaction chimique. Cet exemple montre que des transitions impures peuvent ainsi être introduites pour simplifier le modèle RdP d'un système.

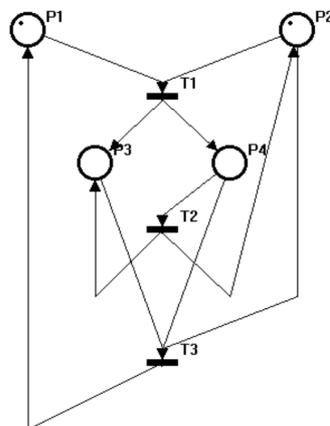
II.6.5. Réseau de Pétri Propre

Un RdP est propre c'est à dire *réinitialisable* si et seulement quel que soit le marquage M_i accessible depuis M_o , il existe une séquence de tirs conduisant à M_o . Cette définition découle en fait de la notion *d'état d'accueil*.

Un RdP possède un état d'accueil M_a pour un marquage initial M_o si pour tout marquage accessible M_i il existe une séquence de tirs S telle que $M_i[S \succ M_a$. Il s'en suit qu'un RdP est réinitialisable (ou réversible) pour un marquage initial M_o si M_o est un état d'accueil.

II.6.6. Réseau de Pétri Vivant

Un RdP est vivant (sans blocage) si quel que soit le marquage M appartenant à l'ensemble de tous les marques accessibles à partir de M_o ($R(M_o)$), il est possible de franchir toute transition du réseau en progressant à travers une séquence de franchissement (Fig. III-31).



Marquage initial : $M_o = [1 \ 1 \ 0 \ 0]$

Tir de $T_1 \rightarrow M_1 = [0 \ 0 \ 1 \ 1]$

Tir de $T_2 \rightarrow M_2 = [0 \ 1 \ 2 \ 0] \rightarrow$ blocage

Figure II-31. RdP non vivant

Donc, un RdP est vivant pour un marquage initial M_o , si et seulement si :

- Toutes les transitions du RdP apparaissent dans le graphe des marquages ;
- Le graphe des marquages est fortement connexe, c'est-à-dire, si et seulement si, en prenant deux nœuds (i, j) quelconque du graphe alors, il existe une séquence d'arcs de i vers j et un autre de j vers i .

$$M_0 \rightarrow M_k ; \forall T_j \exists \text{ séquence de tirs passant par } T_j$$

Un RdP vivant est un RdP sans blocage, un blocage est un état tel qu'aucune transition n'est validée.

II.6.6.1. Réseau de Pétri quasi vivant

Une transition T_j est quasi vivante (Fig. II-32) pour un marquage initial M_0 s'il existe une séquence de franchissements à partir de M_0 contenant T_j . Il s'en suit qu'un RdP est quasi vivant (Fig. II-33) pour un marquage initial M_0 si toutes ses transitions sont quasi vivantes pour ce marquage initial. Conséquemment, une transition qui n'est pas quasi-vivante est inutile.

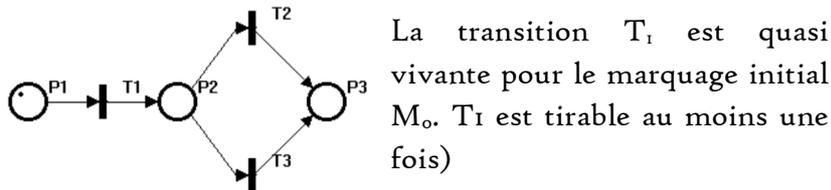


Figure II-32. Transition quasi-vivante

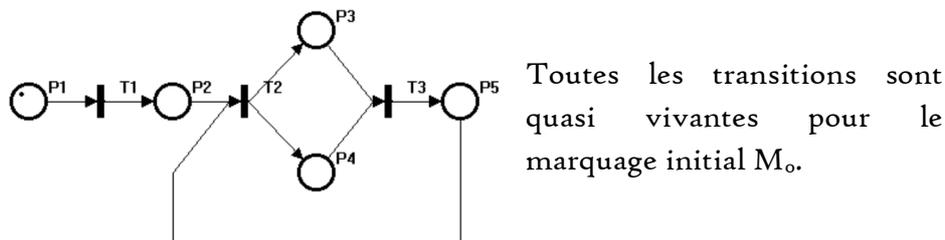


Figure II-33. RdP quasi-vivant

II.6.6.2. Réseau de Pétri pseudo vivant

Un RdP est dit pseudo vivant si depuis le marquage initial, son évolution est telle qu'il existe au moins une transition qui puisse être franchie (Fig. II-34).

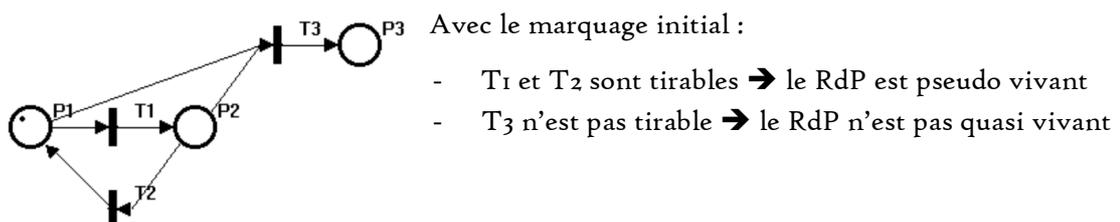


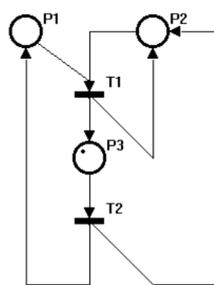
Figure II-34. RdP pseudo vivant

II.6.7. Réseau de Pétri Borné

Soit un réseau R et un marquage M_0 . Une place P_j du réseau marqué (R, M_0) est k -bornée si pour tout marquage M_i accessible depuis M_0 , $M_i(P_j) \leq k$.

$$P_j \text{ est } k\text{-borné} \Leftrightarrow \forall M_i \text{ accessible depuis } M_0, \text{ et } P_j \in P, M_i(P_j) \leq k$$

Un RdP marqué est borné si toutes ses places sont bornées. (Si $k = 1 \rightarrow$ le RdP est dit sauf ou binaire). Dans le cas contraire la place P_j est dite non bornée. Il s'ensuit que le RdP est qualifié de non borné (Fig. II-35).



Marquages successifs :

Marquage initial : $M_0 = [0 \ 0 \ 1]$

Tir de $T_2 \rightarrow M_1 = [1 \ 1 \ 0]$

Tir de $T_1 \rightarrow M_2 = [0 \ 1 \ 1]$

Tir de $T_2 \rightarrow M_3 = [1 \ 2 \ 0]$

Tir de $T_1 \rightarrow M_4 = [0 \ 2 \ 1]$

Tir de $T_2 \rightarrow M_5 = [1 \ 3 \ 0] \dots$

Croissance infinie des jetons dans la place P_2 .

Figure II-35. RdP non borné

II.6.8. Réseau de Pétri Conforme

Un réseau de Pétri vivant et sauf est dit conforme : 1 jeton, sans blocage.

II.6.9. Réseau de Pétri Généralisé

Le franchissement d'une transition validée consiste à enlever un jeton de toutes les places d'entrée de la transition et à en déposer un dans toutes les places de sortie de cette même transition.

Ceci implique que les arcs orientés reliant les places (d'entrée et de sortie) aux transitions ont par défaut une *valuation* ou un *poids* d'une unité. Ce sont les RdP *ordinaires*. De manière plus générale la valuation ou le poids des arcs d'un RdP peut être supérieure à 1.

Dans ce cas les RdP sont dits *généralisés*. Chaque arc porte le nombre de jetons qui le traverse lors d'un tir de transition [16,21].

Tout RdP généralisé peut être transformé en RdP ordinaire. Cette transformation qui ne présente pas d'intérêt particulier pour l'analyse par les RdP est uniquement donnée pour illustrer l'équivalence entre RdP généralisé et RdP ordinaire. Par ailleurs, le RdP présenté à la Figure II-36 est relativement simple mais cette transformation peut conduire à des RdP ordinaires très complexes.

Dans le RdP généralisé l'arc $T_1 \rightarrow P_1$ a un poids de 2. Dans le RdP ordinaire équivalent, la transition T_1 est remplacée par le **sous-réseau** constitué de T'_1 , P'_4 et T''_1 . Pour le marquage indiqué, seule la transition T'_1 est validée. Le franchissement de T'_1 , enlève un jeton de P_2 , de P_3 et de P'_4 et en dépose un dans P_1 et dans P'_4 . Pour ce nouveau marquage seule T''_1 est franchissable. Son franchissement dépose alors un deuxième jeton dans P_1 . Le marquage est alors identique à celui qu'atteindrai le RdP généralisé par le franchissement de T_1 . Le rôle de la place P_4 est de permettre le franchissement en séquence des transitions T'_1 et T''_1 avant le franchissement de T_2 .

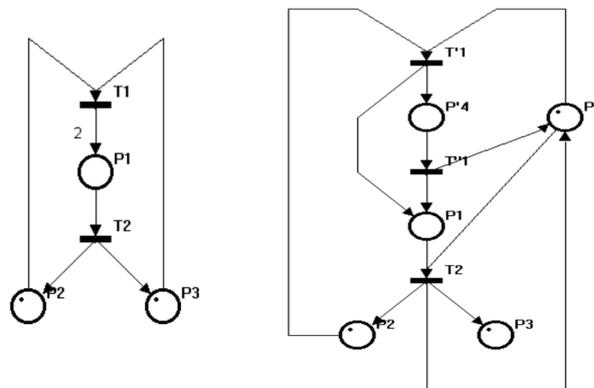


Figure II-36. Transformation du RdP

II.6.10. Réseau de Pétri Autonome et non Autonome

Un RdP autonome décrit le fonctionnement d'un système dont les instants de franchissement ne sont pas connus ou indiqués [22].

Un RdP non autonome décrit le fonctionnement d'un système dont l'évolution est conditionnée par des événements externes ou par le temps. Un RdP non autonome est synchronisé et/ou temporisé.

Le RdP de la Figure II-37 est autonome (cycle des quatre saisons), le moment de passage d'une saison à une autre n'est pas indiqué.

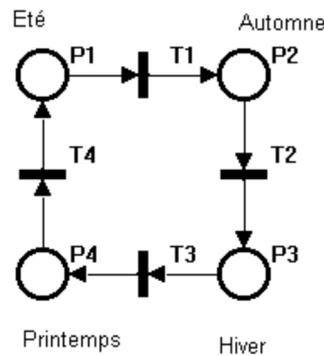


Figure II-37. RdP autonome

II.6.II. Réseau de Pétri à Capacités

Dans un RdP ordinaire (ou généralisé) il n'y a aucune limitation quant au nombre de jetons pouvant être déposé dans une place. Un RdP à capacités est un RdP dans lequel des capacités (nombres entiers strictement positifs) sont associées aux places [21]. Le franchissement d'une transition d'entrée d'une place P_i dont la capacité est $Cap(P_i)$ n'est possible que si le franchissement ne conduit pas à un nombre de jetons dans P_i qui est plus grand que $Cap(P_i)$.

Dans le RdP de la Figure II-38, la transition T_1 ne peut avoir lieu lorsque le nombre de jetons dans P_2 atteint 2.

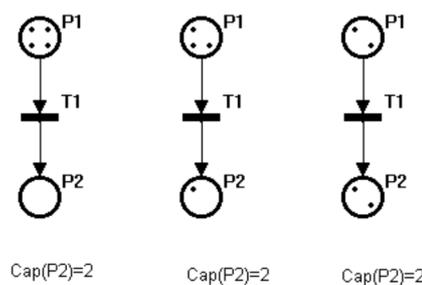


Figure II-38. RdP à capacités

II.6.II.1. Exemple concret

On reprend l'exemple de l'atelier de découpe du bois. Si on considère que la capacité de stockage des pièces traitées est 3 par exemple, le RdP associé est modifié de la manière suivante (Fig. II-39) :

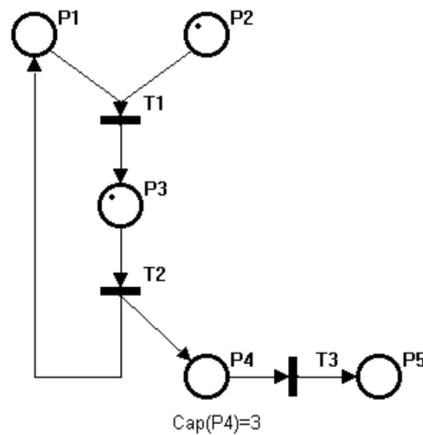


Figure II-39. RdP de l'atelier de découpe du bois

II.6.II.2. Transformation d'un RdP à capacité en RdP ordinaire

Tout RdP à capacité peut être transformé en RdP ordinaire. Pour transformer le RdP à capacité de la Figure II-40 en RdP ordinaire, il suffit de supprimer la capacité de la place P_j est d'ajouter une place P_{ji} qui est place d'entrée de T_i et place de sortie de T_j , la capacité de la place P_j est alors limitée par le fonctionnement du RdP.

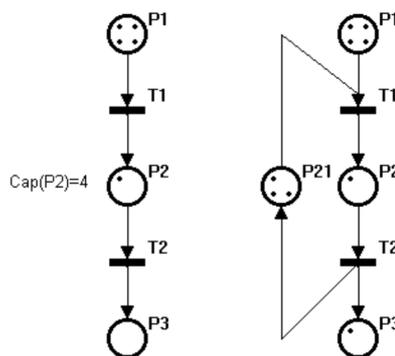


Figure II-40. Transformation d'un RdP à capacité en RdP ordinaire

Le fonctionnement du RdP ordinaire considéré à la Figure II-40 est détaillé ci-dessous (Fig. II-41). La place P_2 ne peut contenir qu'un maximum de 4 jetons.

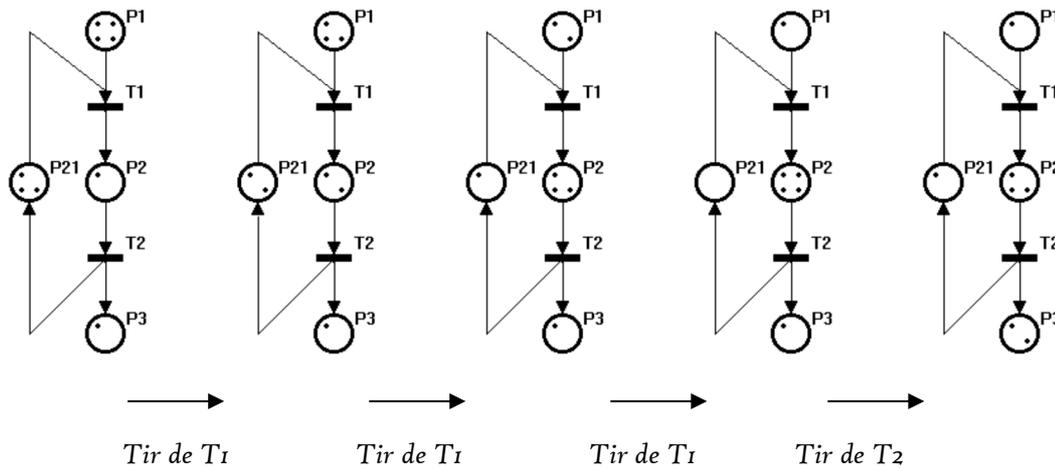


Figure II-41. Passage de transformation d'un RdP à capacité en RdP ordinaire

II.6.12. Réseau de Pétri Interprété

On appelle un RdP interprété un réseau tel que (Fig. II-42) :

- les événements (entrées) sont associés aux transitions T_i ;
- les conditions (actions, sorties) sont associées aux places P_i ;
- les états sont associés aux marquages M_i

Tir de la transition si :

- 1) transition sensibilisée
- 2) prédicat sur des entrées soit vrai (exemple : $ab = 1$)

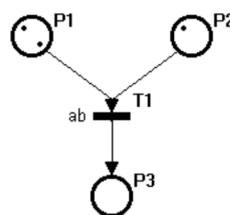


Figure II-42. RdP interprété

II.7. STRUCTURES PARTICULIÈRES DES RÉSEAUX DE PÉTRI

Les RdPs permettent de modéliser un certain nombre de comportements importants dans les systèmes : le parallélisme, la synchronisation, le partage de ressources, la mémorisation et la lecture d'information, la limitation d'une capacité de stockage. Dans cette section, sont présentées les différentes

structures apparaissant dans un réseau de Pétri reproduisant ce type de comportements.

II.7.1. Parallélisme

Le parallélisme représente la possibilité que deux ou plusieurs processus évoluent de manière simultanée au sein du même système à l'aide d'une transition ayant plusieurs places de sortie. Pour cela, le RdP doit contenir la structure présentée dans la Figure II-43.

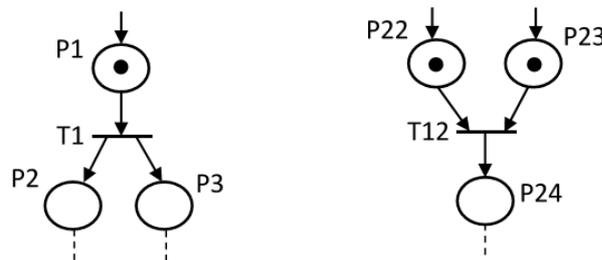


Figure II-43. Parallélisme

Le franchissement de la transition T_1 met un jeton dans la place P_{12} (ce qui marque le déclenchement du processus 1) et un jeton dans la place P_{21} (ce qui marque le déclenchement du processus 2). La place P_{in} correspond à la fin du processus 1 et la place P_{2k} à la fin du processus 2. Le RdP évoluera par franchissement de la transition T_2 . Pour cela, il est nécessaire que les places P_{in} et P_{2k} contiennent chacune au moins un jeton, c'est-à-dire que les processus 1 et 2 soient terminés. Cette remarque traduit la notion de synchronisation.

II.7.2. Synchronisation

II.7.2.1. Synchronisation mutuelle

La synchronisation *mutuelle* ou *rendez-vous* permet de synchroniser les opérations de deux processus (ou plusieurs). Dans l'exemple de la Figure II-44, le franchissement de la transition T_3 ne peut se faire que si la place P_1 du processus 1 et la place P_2 du processus 2 contiennent chacun au moins un jeton. Si ce n'est pas le cas, par exemple la place P_2 ne contient pas de jeton, et la place

P_1 contient un jeton, le processus 1 est “ bloqué ” sur la place P_1 : il attend que l'évolution du processus 2 soit telle qu'au moins un jeton apparaisse dans la place P_2 .

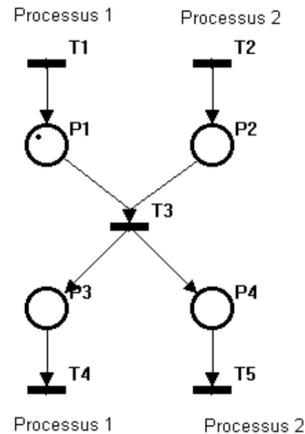


Figure II-44. Passage

II.7.2.2. Synchronisation sémaphore

Les opérations d'un processus ne peuvent se poursuivre que si un processus a atteint un certain niveau dans la suite de ses opérations. Par contre, l'avancement des opérations du premier processus ne dépend pas de l'avancement des opérations du second processus. Pour l'exemple ci-contre, le processus 2 ne peut franchir la transition T_4 que si la place P_0 contient au moins un jeton. Un jeton est ajouté dans la place P_0 lorsque l'évolution du processus 1 amène le franchissement de la transition T_1 . L'évolution du processus 2 va donc dépendre de l'évolution du processus 1.

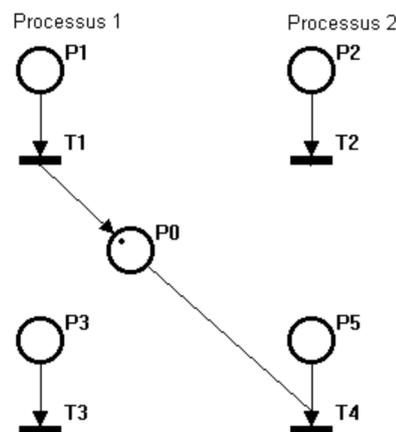


Figure II-45. Synchronisation sémaphore

II.7.3. Partage de Ressources

Cette structure va modéliser le fait qu'au sein du même système plusieurs processus partagent une même ressource. Dans la Figure II-46, le jeton dans la place P_0 représente une ressource mise en commun entre le processus 1 et le processus 2. Le franchissement de la transition T_1 lors de l'évolution du processus 1 entraîne la "consommation" du jeton présent dans la place P_0 . La ressource que constitue ce jeton n'est alors plus disponible pour l'évolution du processus 2 puisque le franchissement de la transition T_2 n'est plus possible.

Lors de l'évolution du processus 1, lorsque la transition T_2 est franchie, un jeton est alors "redonné" à la place P_0 : la ressource redevient alors disponible pour l'évolution des deux processus.

On retiendra donc que la propriété d'exclusion mutuelle est garantie par l'utilisation exclusive du jeton de la place P_0 par les processus P_1 et P_2 .

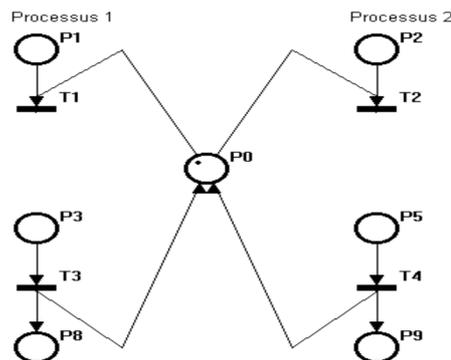


Figure II-46. Partage de ressources

II.8. RÉSEAUX DE PÉTRI TEMPORISÉS

Les réseaux de Pétri abordés jusqu'à présent modélisent des systèmes dont l'évolution ne dépend pas du temps. En effet, dans le modèle classique, les transitions sont activées sans tenir compte des contraintes temporelles comme les temporisations par exemple. Il existe aussi des réseaux de Pétri temporels qui permettent d'étudier des problèmes plus spécifiques aux systèmes en temps réel, comme l'ordonnancement dynamique. Ces réseaux ont montré leur intérêt en particulier dans l'évaluation des performances d'un système [15,16].

Deux façons de modéliser la temporisation : par une temporisation associée aux places (RdP P-temporisé) ou associée aux transitions (RdP T-temporisé). On montre par modification de la structure, il est possible de transformer un RdP P-temporisé en RdP T-temporisé et inversement. Ainsi et sans perte de généralité, on présentera uniquement les RdP P-temporisés.

A titre d'illustration, on considère l'exemple suivant dans lequel, on a associé une temporisation d_1 pour la place P_1 et une temporisation d_2 pour la place P_2 .

II.8.1. RdP P-Temporisé

Une durée minimale de séjour dans les places : Durée pendant laquelle tout jeton qui vient d'être produit dans une place ne peut pas encore servir à l'activation de transitions aval (Fig. II-47).

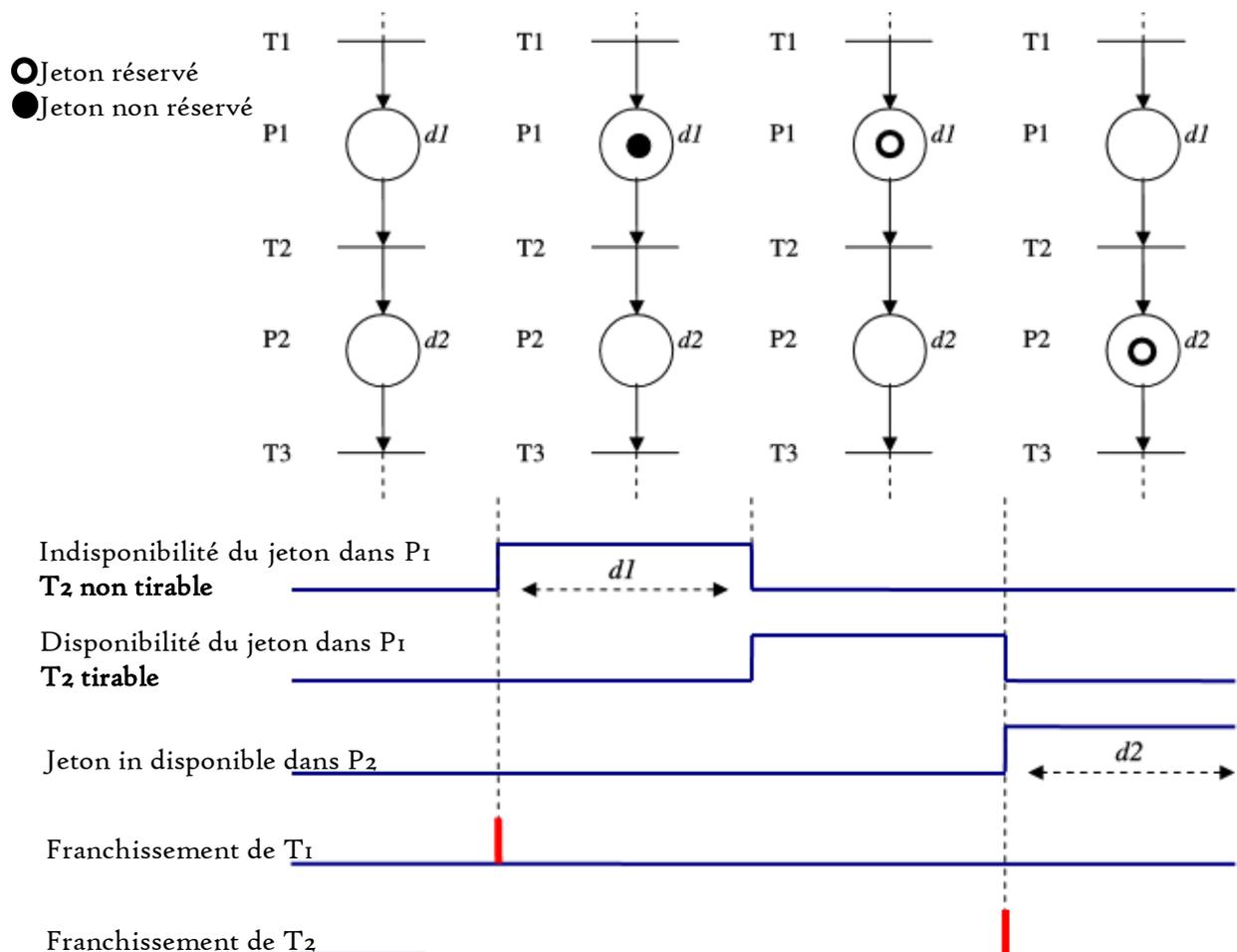


Figure II-47. RdP P-Temporisé

Pour les RdP P-temporisés, on associe à chaque place P une temporisation T, éventuellement nulle. On considère le cas le plus simple où T est fixe mais dans le cas général elle peut être variable.

II.8.2. RdP T-Temporisé

Une durée d'activation pour les transitions : Durée pendant laquelle un jeton situé dans chaque place amont de la transition activée est « réservé » pour cette transition (avant de disparaître), et au-delà de laquelle un jeton apparaît dans chacune des places aval (Fig. II-48).

Lorsqu'un jeton est déposé dans une place P, ce jeton doit rester au moins pendant le temps T. On dit que le jeton est *indisponible* pendant cette durée. Quand la temporisation T s'est écoulée, alors le jeton devient *disponible*.

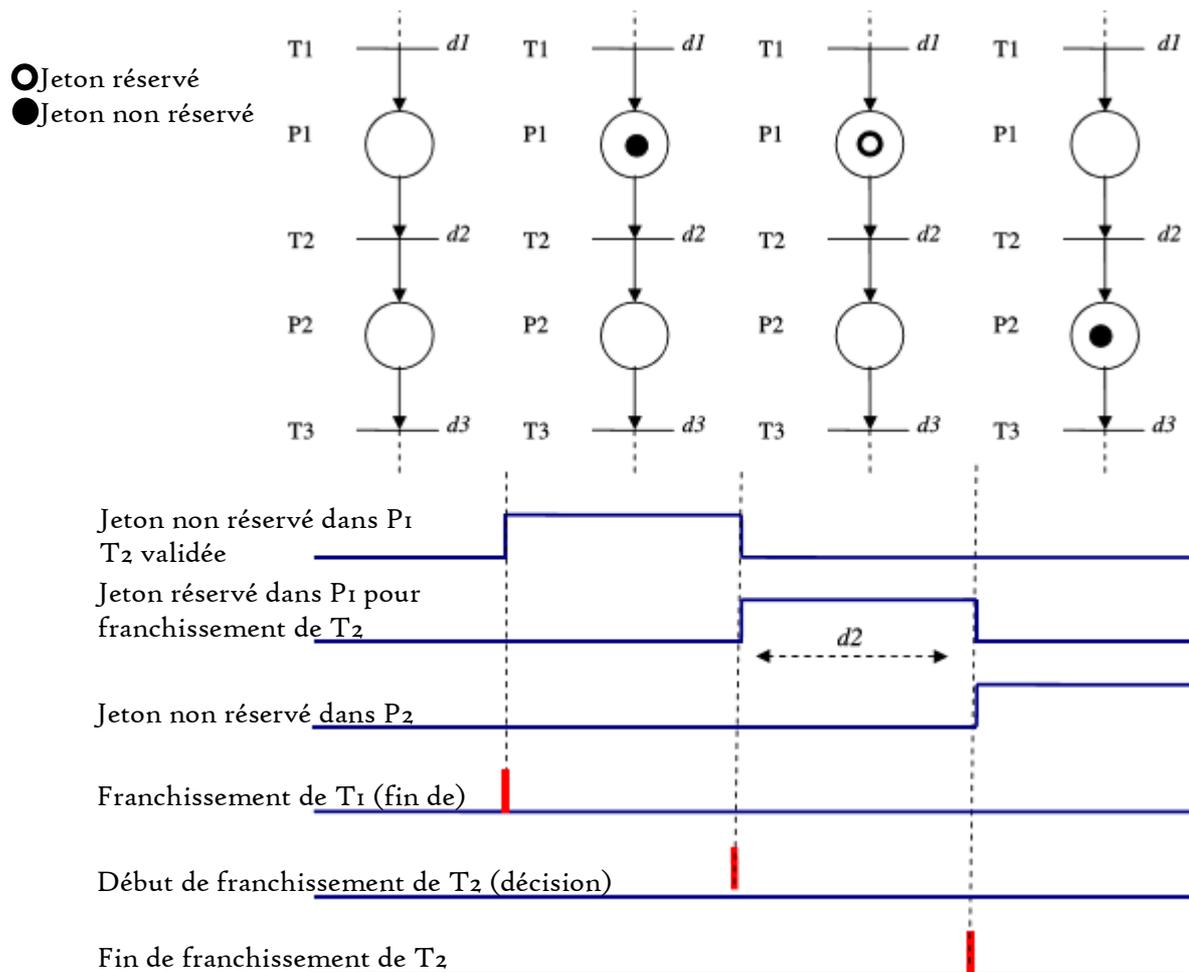


Figure II-48. RdP T-Temporisé

II.9. CONCLUSION

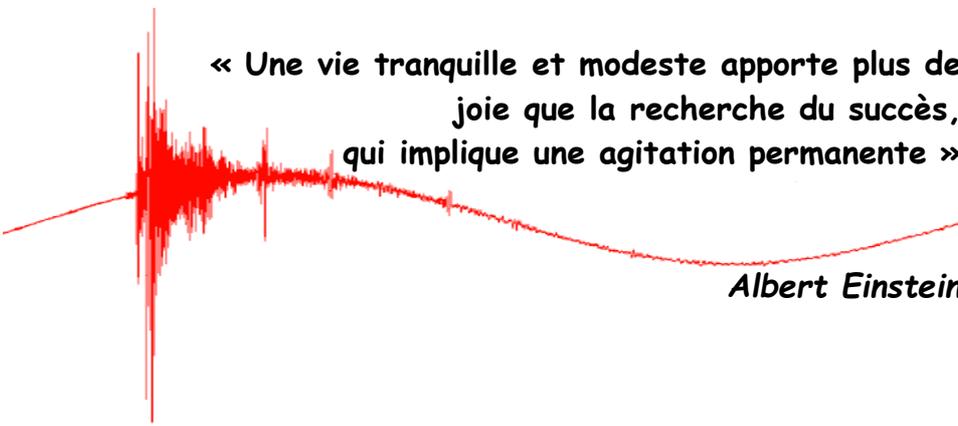
J'ai présenté dans ce chapitre le formalisme de RdP. Les RdPs sont particulièrement bien adaptés à la description des différents aspects dynamiques ou comportementaux d'un système. Des concepts tels que les notions de bases, l'évolution, le marquage et les séquences s'expriment aisément dans le cadre de ce formalisme. Ainsi que les différentes topologies et propriétés ont été présentées.

L'avantage des RdP par rapport à la grande majorité des autres outils, est leur aptitude à donner des informations concernant l'exécution du système. De la théorie mathématique, on peut déduire un certain nombre de propriétés qui sont autant d'informations à la disposition du concepteur pour se faire une idée du comportement de son système.

Après avoir présenté un outil de modélisation des structures des systèmes automatisés, on peut passer à l'étude des bases et les règles d'établissement du GRAFCET qui sera le titre du chapitre suivant.

CHAPITRE III

GRAFCET OU SFC



« Une vie tranquille et modeste apporte plus de joie que la recherche du succès, qui implique une agitation permanente »

Albert Einstein

CHAPITRE III

GRAFCET OU SFC

III.1. INTRODUCTION

Au départ, le **GRAFCET** (**GRA**phe **F**onctionnel de **C**ommande par **E**tapes et **T**ransitions) a été conçu comme un outil méthodologique de description du cahier des charges de la partie commande d'un système automatisé. Il résulte de travaux entrepris dès 1975 sous l'égide de l'AFCECT (Association Française pour la Cybernétique Economique et Technique) par un groupe d'industriels et de chercheurs. L'ADEPA (Agence nationale pour le DEveloppement de la Production Automatisée) s'est ensuite attachée à sa promotion tout en lui conférant une forme susceptible d'aboutir à des normes nationales et internationales.

De fait, le GRAFCET a été normalisé en France en 1982, mais il a fallu attendre 1988 pour le voir reconnaître sur le plan international par la norme 848 édictée par la CEI (Commission Electrotechnique Internationale). Pour l'occasion, il a d'ailleurs été rebaptisé **SFC** (**S**equential **F**unction **C**hart).

La présentation du GRAFCET en tant qu'outil de description des automatismes industriels, permet de traiter la majorité des problèmes rencontrés lors de la spécification, la conception et la réalisation d'une partie commande d'un automate industriel.

Dans ce chapitre, je commence par donner les bases nécessaires de la modélisation des systèmes automatisés en langage GRAFCET. Ensuite, j'illustrerai ces différentes règles et on présente les différentes structures du GRAFCET. Les différentes actions associées aux étapes sont définies par la suite. En fin, on montre la transformation du modèle GRAFCET en des équations.

III.2. DÉFINITION DU GRAFCET

Le **GRAFCET** (**GRA**phe **F**onctionnel de **C**ommande par **E**tapes et **T**ransitions) ou **SFC** (**S**equential **F**onction **C**hart) est un outil graphique qui décrit les différents comportements de l'évolution d'un automatisme et établit une correspondance à caractère séquentiel et combinatoire entre [9-13] :

- ◆ Les *entrées*, c'est-à-dire les transferts d'informations de la Partie Opérative vers la Partie Commande ;
- ◆ Les *sorties*, transferts d'informations de la Partie Commande vers la Partie Opérative.

C'est un outil graphique puissant, directement exploitable, car c'est aussi un langage pour la plupart des API existants sur le marché.

Le GRAFCET comprend :

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

Un GRAFCET est un graphe orienté biparti défini par $G=(E,T,A,M_0)$:

- ◆ des étapes E
- ◆ des transitions T
- ◆ des arcs A
- ◆ un marquage initial M_0

III.3. 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 [12,17,23].

III.4. DOMAINE D'APPLICATION

Le diagramme fonctionnel est indépendant des techniques séquentielles “tout ou rien”, pneumatique, électrique ou électronique, câblées ou programmées, pouvant être utilisées pour réaliser l'automatisme de commande. Mais l'utilisation de séquenceurs, d'une part, et d'automates à instructions d'étapes d'autre part, permet une transcription directe du diagramme fonctionnel [15].

Cette représentation graphique concise et facile à lire est aisément compréhensible par toute personne en relation avec le système automatisé, du concepteur à l'utilisateur sans oublier l'agent de maintenance. Utilisé industriellement, le GRAFCET est aussi enseigné dans les options techniques et l'enseignement supérieur.

Depuis les premières publications le concernant et surtout depuis la Norme Française NF Co3-190 de 1982, cet outil a été travaillé et enrichi par le groupe systèmes logiques de l'AF CET (Association Française pour la Cybernétique Economique et Technique) Il existe une documentation et symboles graphiques, diagramme fonctionnel "Grafcet" éditée par l'Union Technique de l'Electricité. UTE Co3-190 Novembre 1990.

III.5. NOTIONS DU POINT DE VUE SAP/PO/PC

Le diagramme La représentation d'un système automatisé par un grafcet prend en compte le *point de vue* selon lequel l'observateur s'implique au fonctionnement de ce système (Fig. III-1).

On distingue trois points de vue [13-17,24,25] :

- GRAFCET du point de vue Système Automatisé de Production (SAP) ;
- GRAFCET du point de vue Partie Opérative (PO) ;
- GRAFCET du point de vue Partie Commande (PC).

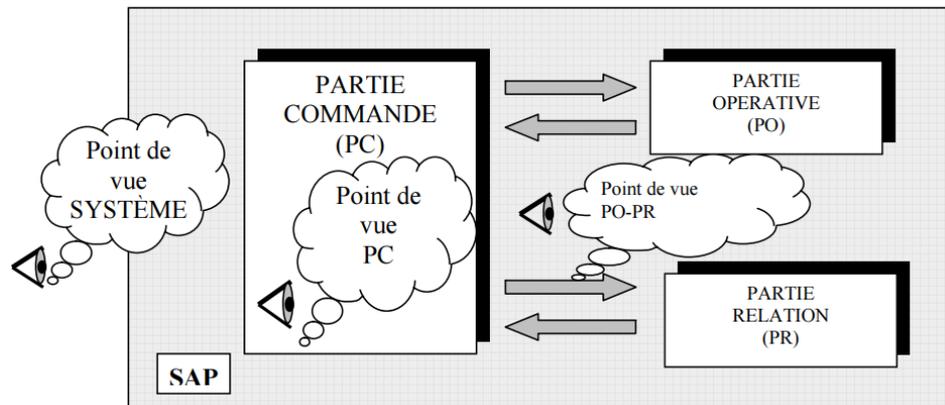


Figure III-1. Notion du point de vue SAP/PO/PC

III.5.1. GRAFCET Point de Vue Système Automatisé de Production (SAP)

Description faite par un observateur se situant d'un point de vue externe au SAP. Le point de vue système décrit le comportement du système vis à vis du produit. Le procédé est l'ensemble des fonctions successives exécutées sur un même produit au cours de sa fabrication [20].

Le processus est l'organisation du procédé. C'est la succession des fonctions simultanées réalisées sur tous les produits présents dans le système automatisé.

Le GRAFCET du point de vue système permet le dialogue entre le client et le concepteur pour la spécification du système automatisé.

III.5.2. GRAFCET Point de Vue Partie Opérative (PO)

Description du comportement du système faite par un observateur se situant d'un point de vue interne au SAP et externe à la PC. Les choix technologiques de la PO sont effectués. Le point de vue partie opérative décrit les actions produites par les actionneurs à partir des informations acquises par les capteurs [20,24].

Le GRAFCET du point de vue partie opérative permet le dialogue entre le concepteur de la partie opérative et le concepteur de la partie commande.

La notation, à ce niveau peut être littérale ou symbolique en utilisant les repères du dossier technique.

III.5.3. GRAFCET Point de Vue Partie Commande (PC)

Description du comportement du système par un observateur se situant d'un point de vue interne à la PC. Ce GRAFCET prend en compte les choix technologiques et l'ensemble des échanges $PC \leftrightarrow PO$ et $PC \leftrightarrow \text{Opérateur}$. Il décrit dans un premier temps la marche normale et peut évoluer en fonction des modes de marches et d'arrêts imposés par le cahier des charges du système automatisé. C'est le GRAFCET du point de vue du réalisateur de la Partie Commande. La notation retenue à ce niveau est la notation symbolique utilisant les repères du dossier technique [24,25].

III.6. STRUCTURE GRAPHIQUE DU GRAFCET

Un GRAFCET est un graphe composé de *séquences*, c'est-à-dire une *succession d'étapes* et de *transitions* reliées par des *arcs orientés*.

 **Exemple :** Pour percer une pièce, la séquence suivante s'exécute après une action de l'opérateur de mise en marche :

- serrer la pièce ;
- Percer la pièce ;
- desserrer la pièce.

Il s'agit d'un fonctionnement séquentiel qui peut être représenté par un GRAFCET dans la Figure III-2.

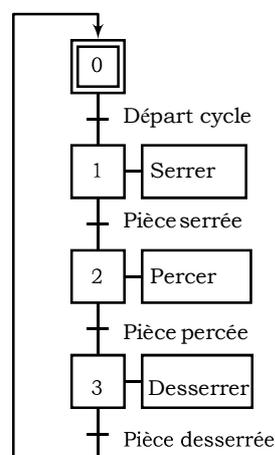


Figure III-2. Grafcet point de vue système de perçage d'une pièce

III.7. ÉLÉMENTS DE BASE DU GRAFCET

La définition simplifiée précédente a permis de présenter de manière intuitive les trois concepts fondamentaux du GRAFCET : *étape - transition - liaisons orientées*. Je vais maintenant en donner des définitions plus précises.

III.7.1. Étape

- ▶ une *étape* caractérise un comportement invariant d'une partie ou de la totalité de la partie commande ; à un instant donné et suivant l'évolution du système ;
- ▶ une *étape* est soit *active* ou *inactive* ;
- ▶ l'ensemble des étapes actives définit la *situation* de la partie commande ;
- ▶ les étapes qui sont actives au début du processus sont appelées *étapes initiales* qui seront représentées par un carré double ;
- ▶ une étape est repérée numériquement par un numéro, i , et possède ainsi une variable d'état notée X_i ;
 - ⊕ $X_i = 1$ quand l'étape est *active* ;
 - ⊕ $X_i = 0$ quand l'étape est *inactive*.

La situation initiale d'un système automatisé est indiquée par une étape dite *étape initiale* et représentée par un carré double (Fig. III-3) [20].

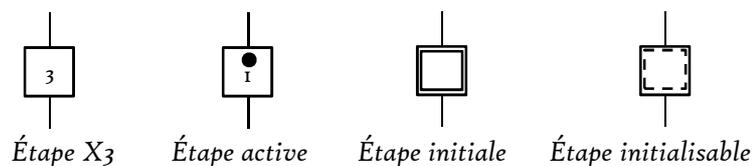


Figure III-3. Définitions étapes

III.7.2. Action

- ▶ une ou plusieurs actions peuvent être associées à une étape ;
- ▶ une *action* traduit *ce qui doit être fait* chaque fois que l'étape à laquelle elle est associée est active (Fig. III-4).

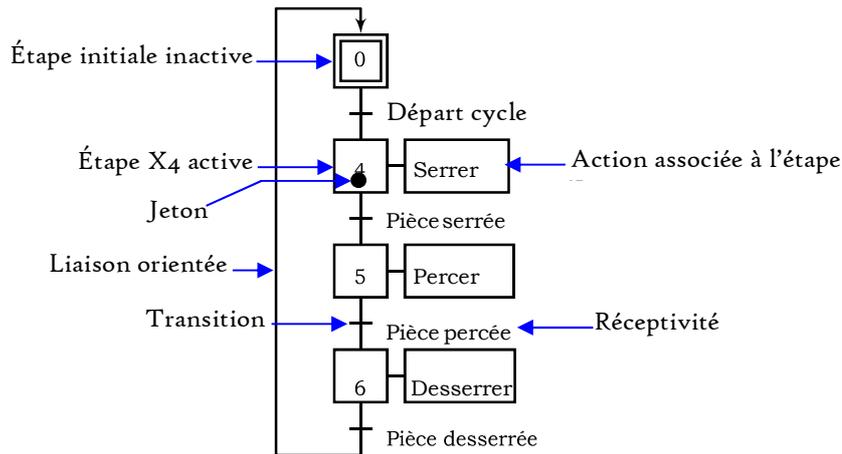


Figure III-4. Éléments de base du GRAFCET

III.7.3. Transition

- ▶ une **transition** est représentée par un petit trait perpendiculaire à la liaison orientée ;
- ▶ une **transition** indique la possibilité d'évolution entre plusieurs étapes ;
- ▶ une **transition** peut être **franchie** provoquant le changement d'activité des étapes.

III.7.4. Réceptivité

- ▶ une **réceptivité** est une **condition logique** associée à une **transition** ;
- ▶ une **réceptivité** peut être **vraie** ou **fausse**.

III.7.5. Liaisons Orientée

- ▶ une **liaison orientée** relie les étapes aux transitions et les transitions aux étapes ;
- ▶ une liaison orientée indique les voies d'évolution.

III.8. CLASSIFICATION DES ACTIONS ASSOCIÉES AUX ÉTAPES

Les actions associées à une étape indiquent ce qui doit être fait chaque fois que l'étape à laquelle elles sont associées est active. Elles sont choisies parmi les sorties du système. Leur nature diffère donc légèrement en fonction du point de vue adopté. Le critère retenu pour la classification des actions les plus usitées est la durée de l'action comparativement à la durée d'activité d'une étape [25,26].

III.8.1. Action Continue

L'ordre est émis, de façon continue, tant que l'étape, à laquelle il est associé, est active (Fig. III-5) [20,28].

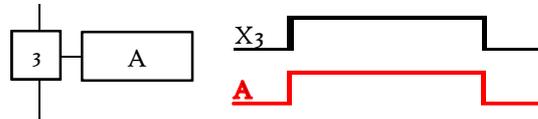


Figure III-5. Action continue

III.8.2. Action Conditionnelle

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 [20].

III.8.2.1. Action conditionnelle simple (Type C)

C'est une action continue dont l'exécution est soumise à une condition logique. La condition d'assignation peut être une expression booléenne (Fig. III-6).

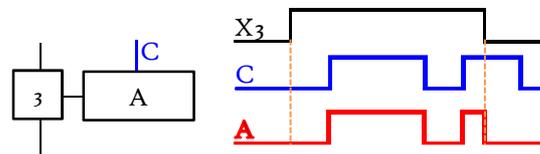


Figure III-6. Action conditionnelle simple

III.8.2.2. Action temporisée retardée (Type D)

C'est un cas particulier d'action conditionnelle dans laquelle le temps intervient comme condition associée à l'action. L'*action retardée* : la condition d'assignation n'est vraie qu'après une durée t_r de puis l'activation de l'étape (Fig. III-7).

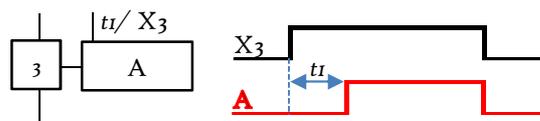


Figure III-7. Action temporisée retardée

III.8.2.3. Action temporisée limitée (Type L)

La condition d'assignation n'est vraie que pendant une durée t_l depuis l'activation de l'étape (Fig. III-8).

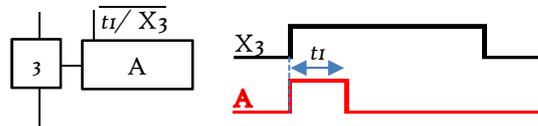


Figure III-8. Action temporisée limitée

III.8.3. Action Maintenu

L'action doit se poursuivre pendant la durée d'un certain nombre d'étapes consécutives (Fig. III-9). Effet maintenu par des actions continues non mémorisées :

- ▶ soit répéter l'action dans toutes les étapes ;
- ▶ soit utiliser les structures de séquences simultanées.

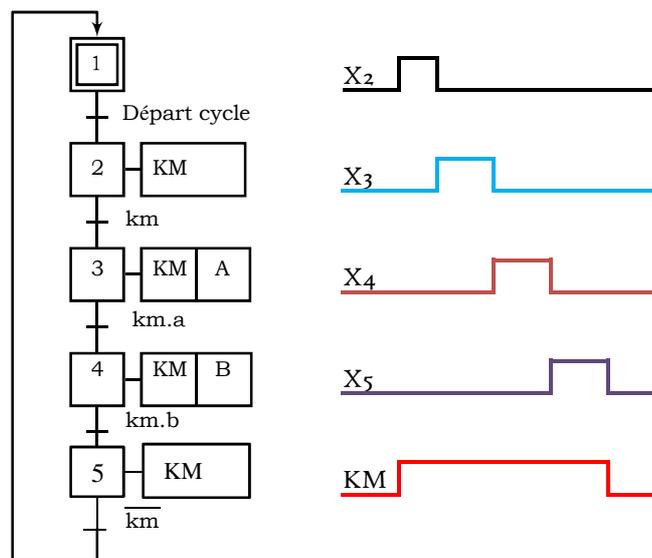


Figure III-9. Action maintenue

III.8.4. Action Mémorisée

Une action mémorisée décrit comment affecter une valeur à une sortie qui la conserve. L'action mémorisée doit obligatoirement être associée à un événement

interne (activation, désactivation, événement ou franchissement). A l'initialisation, la valeur de cette sortie est nulle.

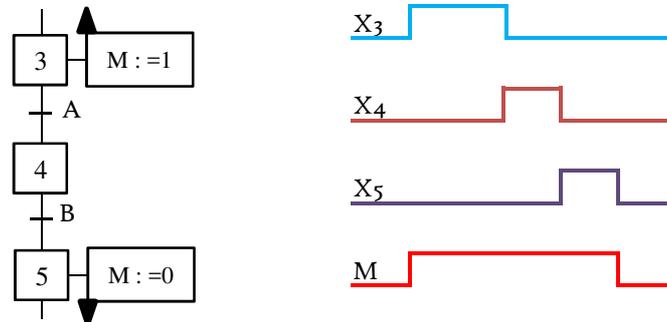


Figure III-10. Action mémorisée

III.8.5. Action Manuelle

Une action manuelle est une action particulière qui correspond à une intervention de l'opérateur (chargement, mise en position de la pièce, etc.) ne peut être considérée comme une action associée à une étape car elle ne correspond pas à un ordre issu de la partie commande vers la partie opérative. Dans ce cas, l'étape doit être considérée comme une étape sans action associée (étape vide). Si l'action est spécifiée à côté de l'étape, elle ne doit pas être encadrée mais signalée comme un commentaire entre guillemets [20].

L'activation de l'étape correspond alors à l'attente d'un événement extérieur et seul le compte-rendu de cet événement reçu par la partie commande permet de faire évoluer le système.

III.9. CLASSIFICATION DES RÉCEPTIVITÉS

III.9.1. Variables

III.9.1.1. Variable externe

- Une variable binaire délivrée par la partie opérative à commander (état des capteurs) ou par son environnement (état d'un bouton manipulé par l'opérateur) ;

📌 **Exemple :** Réceptivité inscrite de façon littérale (Fig. III-11).

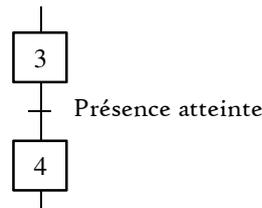


Figure III-11. Réceptivité littérale

➤ **Exemple :** Réceptivité inscrite de façon symbolique à la droite de la transition (Fig. III-12).

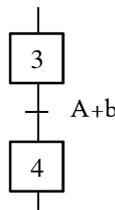


Figure III-12. Réceptivité symbolique

➤ Une variable binaire relative au temps.

➤ **Exemple :** Réceptivité faisant intervenir le temps : l'étape 3 sera active pendant 20 secondes (Fig. III-13).

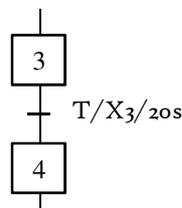


Figure III-13. Réceptivité binaire relative au temps

III.9.1.2. Variable interne

- Une variable binaire relative à la partie commande, c'est à dire à la situation dans laquelle se trouve le GRAFCET ;
- Une variable générée par le modèle GRAFCET (compteur, variable de calcul ...) ;
- Une variable binaire relative à un prédicat (ex : $[T > 8^{\circ}\text{C}]$ signifie que lorsque la proposition logique « $T > 8^{\circ}\text{C}$ » sera vérifiée alors la variable binaire $[T > 8^{\circ}\text{C}]$ sera égale à 1).

- 📌 **Exemple :** Réceptivité faisant intervenir un prédicat. La réceptivité est vraie lorsque l'assertion $T \leq 20^{\circ}\text{C}$ est vérifiée (Fig. III-14).

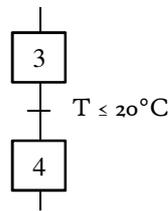


Figure III-14. Réceptivité binaire relative à un prédicat

III.9.2. Événements

Un événement est un changement d'état d'une variable binaire.

- On notera " $\uparrow a$ " l'événement d'entrée front montant de " a " associé au passage de la valeur 0 à la valeur 1 de la variable a .
- 📌 **Exemple :** Réceptivité faisant intervenir le " front montant de a " qui vaut 1 uniquement quand on passe de 0 à 1. (Fig. III-15).

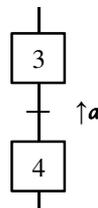


Figure III-15. Réceptivité binaire relative à un événement

- On notera " $\downarrow a$ " l'événement d'entrée front descendant de " a " associé au passage de la valeur 1 à la valeur 0 de la variable a .

III.10. STRUCTURE DE BASE DU GRAFCET

Pour un GRAFCET, on distingue trois structures :

III.10.1. GRAFCET à Séquence Unique

- une séquence est tout ensemble d'étapes successives où chaque étape est suivie d'une seule transition et chaque transition n'est validée que par une seule étape ;

- le GRAFCET le plus simple est le GRAFCET linéaire qui ne comporte qu'une seule séquence.

🔪 **Exemple :** la Figure III-16 présente un chariot pouvant se déplacer sur deux rails de guidage, vers la gauche ou vers la droite entre deux positions *a* et *b*. Initialement, le chariot est au repos en *a*. Lorsqu'on appuie sur bouton *dcy*, il part vers la droite puis il retourne à gauche après une attente de 20 secondes.

- ▣ le GRAFCET de la Figure III-17 correspondant au fonctionnement du système. C'est une suite d'étapes pouvant être activées les unes après les autres.

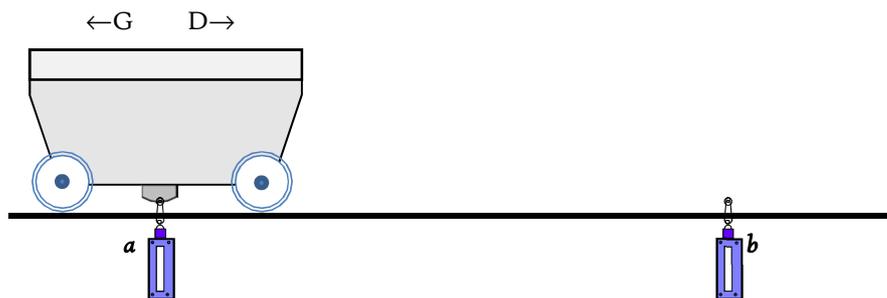


Figure III-16. Système d'aller-retour d'un chariot

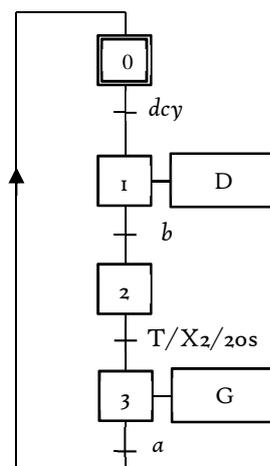


Figure III-17. GRAFCET à séquence unique

III.10.2. GRAFCET à Sélection de Séquence

- la sélection de séquences permet de faire un choix d'une séquence à partir d'une étape ;
- la sélection de séquences est composée d'une *divergence en OU*, symbolisée par *un trait de liaison simple*, qui relie plusieurs transitions comportant les conditions de choix ;
- chaque transition est suivie d'une séquence et une *convergence en OU* qui termine la sélection ;
- le franchissement de chaque transition conduira à la réalisation de la séquence ou de l'action qui suit cette transition ;
- le choix entre les séquences doit être *exclusif*, c'est-à-dire que seulement une des transitions peut être franchie à la fois.

✂ **Exemple :** la Figure III-18 présente un chariot pouvant se déplacer sur deux rails de guidage, vers la gauche ou vers la droite. Initialement, le chariot est au repos en *a*. Si l'on appuie sur bouton *S1*, il part jusqu'à *b1* puis en appuyant sur *R1* il revient en *a*. Si l'on appuie sur bouton *S2*, il part jusqu'à *b2* puis il revient après une attente de 20 secondes.

- le GRAFCET de la Figure III-19 correspondant au fonctionnement du système.

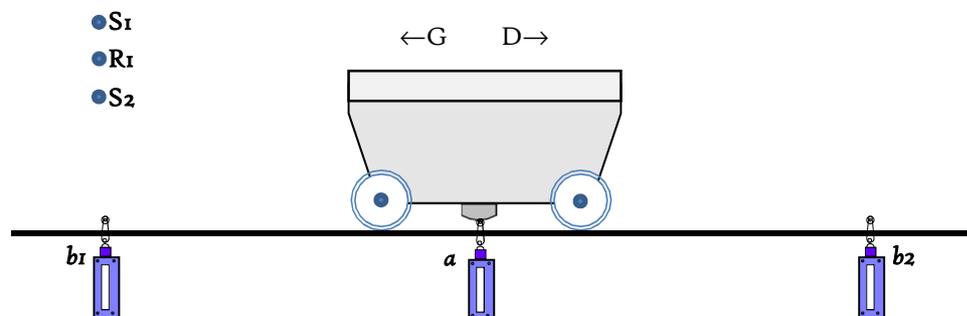


Figure III-18. Système d'aller-retour d'un chariot

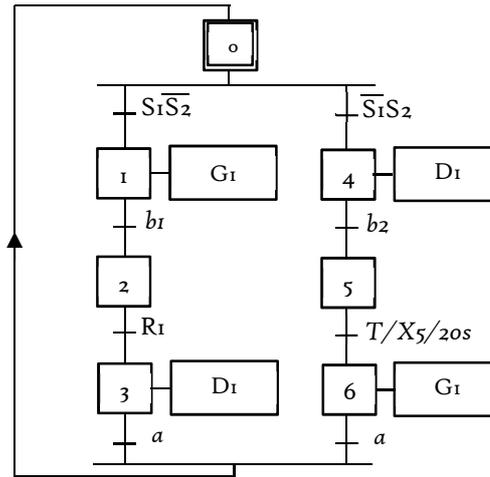


Figure III-19. GRAFCET à sélection de séquence

III.10.3. GRAFCET à Saut d'Étapes

- Le saut d'étapes est un cas particulier de la sélection de séquences ;
- Le saut d'étapes permet soit d'exécuter une séquence, soit de sauter des étapes. Une des branches de la sélection ne contient donc aucune étape.
- ✍ **Exemple :** la Figure III-20 représente un chariot pouvant se déplacer sur deux rails de guidage, vers la gauche ou vers la droite. Le chariot peut faire deux cycles différents à partir de la position de repos en a :
 - En sélectionnant le commutateur C_1 , il part vers b_1 , puis vers b_2 après 20 secondes d'attente et enfin il revient immédiatement à a .
 - En sélectionnant le commutateur C_2 , il part directement vers b_2 enfin il revient immédiatement à a .
- le GRAFCET de la Figure III-21 correspondant au fonctionnement du système.

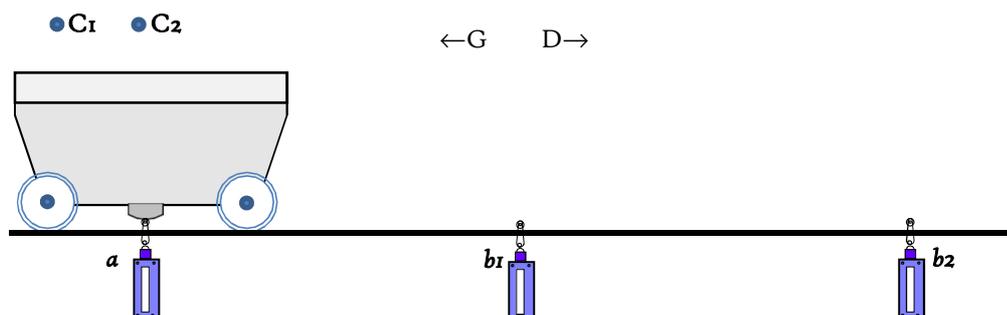


Figure III-20. Système d'aller-retour d'un chariot

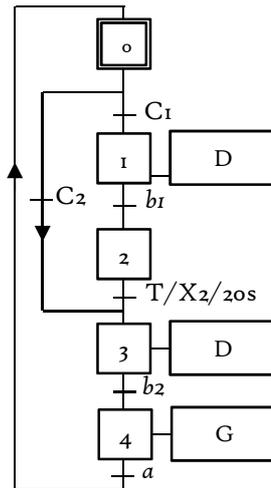


Figure III-21. GRAFCET à saut d'étapes

III.10.4. GRAFCET à Reprise d'Étapes

La *reprise d'étapes* est utilisée lorsqu'il faut répéter une même séquence jusqu'à ce qu'une condition soit satisfaite [I4,20].

📌 **Exemple :** la Figure III-20 présente un chariot pouvant se déplacer sur deux rails de guidage, vers la gauche ou vers la droite. En appuyant sur le bouton départ cycle *dcy*, le chariot se déplace vers *b1* où il s'arrête pendant 20 secondes puis vers *b2* où il attend pendant 30 secondes avant de retourner gauche vers *a*.

- le GRAFCET de la Figure III-22 correspondant au fonctionnement du système.

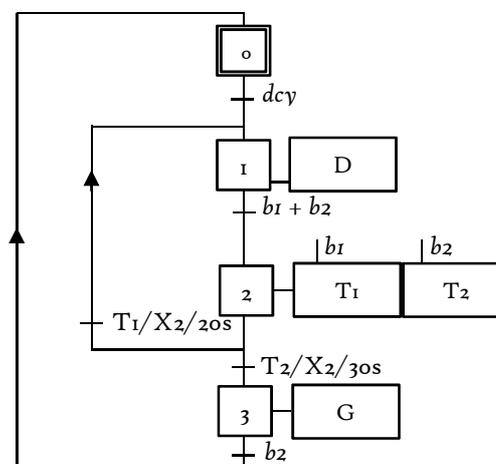


Figure III-22. GRAFCET à reprise d'étapes

III.10.5. GRAFCET à Exclusivité

On distingue trois (3) types d'exclusivité [13,20,24,29] :

III.10.5.1. Exclusivité logique

Les réceptivités $\bar{a}b$ et $a\bar{b}$ sont logiquement exclusives (Fig. III-23).

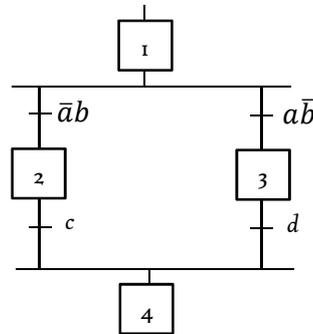


Figure III-23. GRAFCET à exclusivité logique

III.10.5.2. Exclusivité technologique

Les réceptivités C_1 et C_2 sont technologiquement exclusives (Fig. III-24).

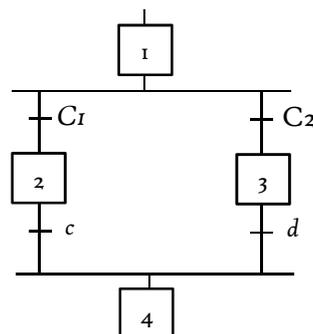


Figure III-24. GRAFCET à exclusivité technologique

III.10.5.3. Exclusivité par priorité

Les réceptivités a et $\bar{a}b$ sont exclusives par priorité. Si $a = b = 1$, l'évolution est de 1 vers 2 (Fig. III-25).

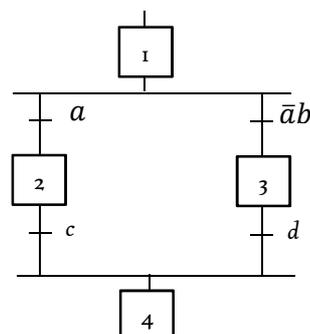


Figure III-25. GRAFCET à exclusivité par priorité

III.10.6. GRAFCET à Séquences Simultanées

- le *parallélisme* est l'exécution simultanée de plusieurs séquences, il est réalisé par une transition qui possède plusieurs étapes de sortie ;
- l'exécution en parallèle de plusieurs séquences débute par une *divergence en ET*, représentée par un trait de liaison double ;
- les séquences parallèles débutent simultanément mais l'évolution dans chaque branche est indépendante. Les séquences parallèles se terminent par une *synchronisation* ;
- la synchronisation consiste en une *convergence en ET*, symbolisée par un trait de liaison double, qui est une transition possédant plusieurs étapes d'entrée ;
- la transition qui suit la synchronisation n'est validée que lorsque toutes les étapes d'entrée sont actives.

 **Exemple :** On considère dans la Figure III-26 les deux chariots CH_1 et CH_2 pouvant se déplacer sur deux rails de guidage, vers la gauche ou vers la droite entre deux positions a_1 et b_1 pour CH_1 et a_2 et b_2 pour CH_2 . Les chariots sont tous deux au repos (en a_1 et a_2), et lorsqu'on appuie sur le bouton dcy , ils partent simultanément vers la droite. Le premier chariot qui arrive à l'extrémité droite repart à gauche et provoque le retour de l'autre chariot. Un nouveau départ ne peut être donné que s'ils sont tous deux en position gauche.

- le GRAFCET de la Figure III-27 correspondant au fonctionnement du système.

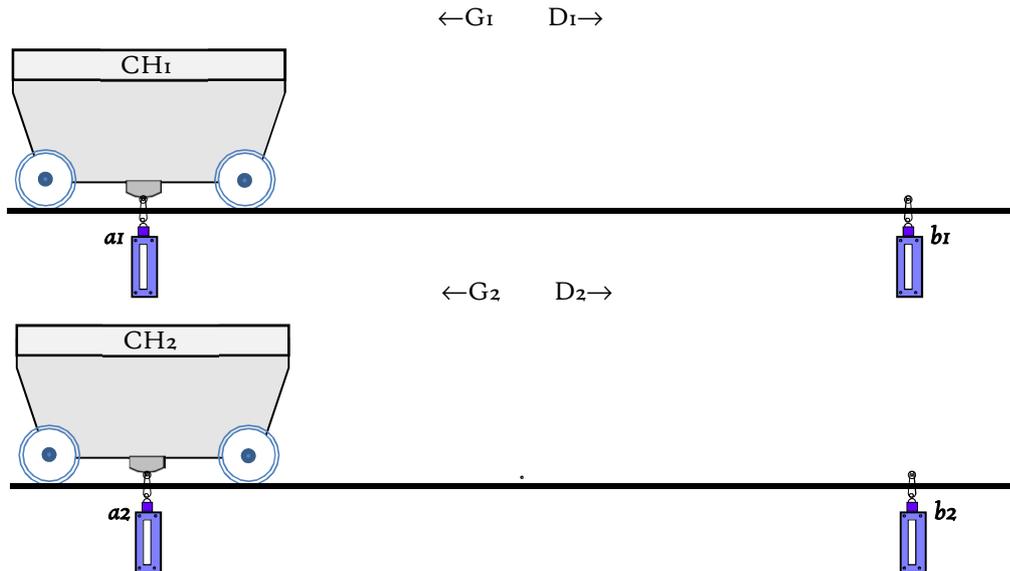


Figure III-26. Système d'aller-retour des deux chariots

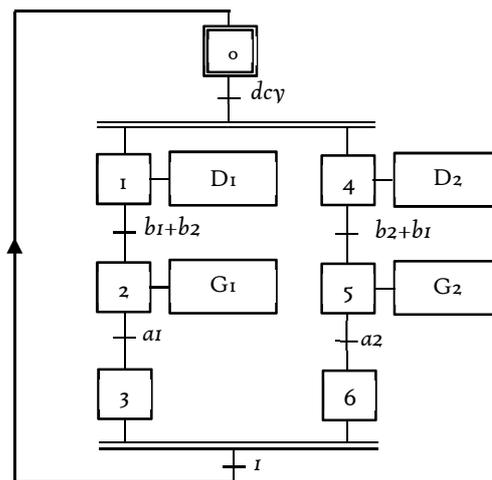


Figure III-27. GRAFCET à séquences simultanées

III.II. RÈGLES DE SYNTAXE DANS UN GRAFCET

- ▶ l'alternance étape-transition et transition-étape doit toujours être respectée quelle que soit la séquence parcourue ;
- ▶ deux étapes ou deux transitions ne doivent jamais être reliées par une liaison orientée ;
- ▶ La liaison orientée relie obligatoirement une étape à une transition ou une transition à une étape.

- ▶ La **divergence en OU ne doit pas être** précédée par une transition ou suivie par des étapes (Fig. III-28-a).
- ▶ La **convergence en OU ne doit pas être** précédée par des étapes ou suivie par une transition (Fig. III-28-b).

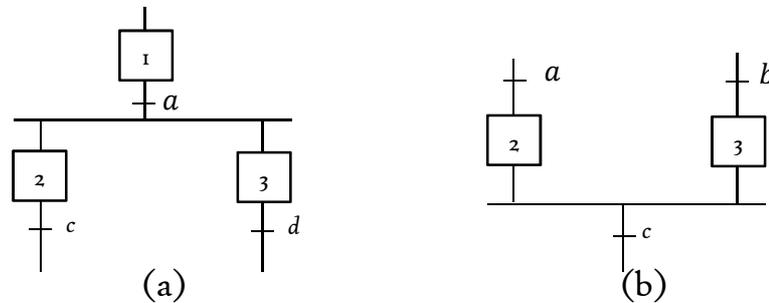


Figure III-28. Syntaxe **interdite** en (a) divergence en OU (b) convergence en OU

- ▶ La **divergence en ET ne doit pas être** précédée par une étape ou suivie par des transitions (Fig. III-29-a).
- ▶ La **convergence en ET ne doit pas être** précédée par des transitions ou suivie par une étape (Fig. III-29-b).

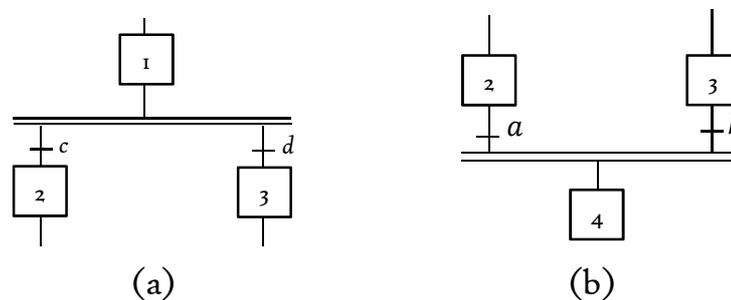


Figure III-29. Syntaxe **interdite** en (a) divergence en ET (b) convergence en ET

III.12. RÈGLES D'ÉVOLUTION D'UN GRAFCET

La modification de l'état de l'automatisme est appelée évolution, et est régie par cinq (5) règles [9-13,20-27] :

III.12.1. Règle I : Situation initiale du GRAFCET

La situation initiale précise quelles étapes doivent être activées lors de la mise en service du système de commande de la machine. Cette situation est choisie par le concepteur. Elle est caractérisée par l'ensemble des **étapes initiales** (Fig. III-30).

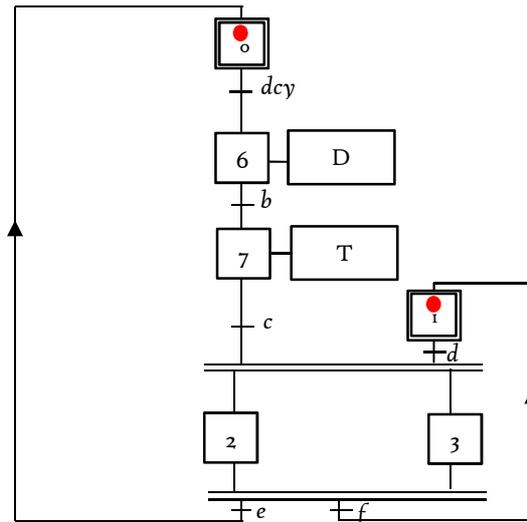


Figure III-30. Situation initiale du GRAFCET

III.12.2. Règle 2 : Transition franchissable

Une transition est franchissable et obligatoirement franchie si les deux conditions suivantes sont remplies :

- ▶ La transition est validée (toutes les étapes qui précèdent immédiatement la transition sont actives) ;
- ▶ La réceptivité associée à cette transition est vraie.

Dans la Figure III-31 (a) : malgré $a = 1$, la transition n'est pas validée, sur la Figure III-31 (b) : l'étape 3 n'est pas encore active, la transition n'est pas validée et dans la Figure III-31 (c) : toutes les étapes sont active et $a = 0$ la transition est validée, c'est à dire prête à être franchie.

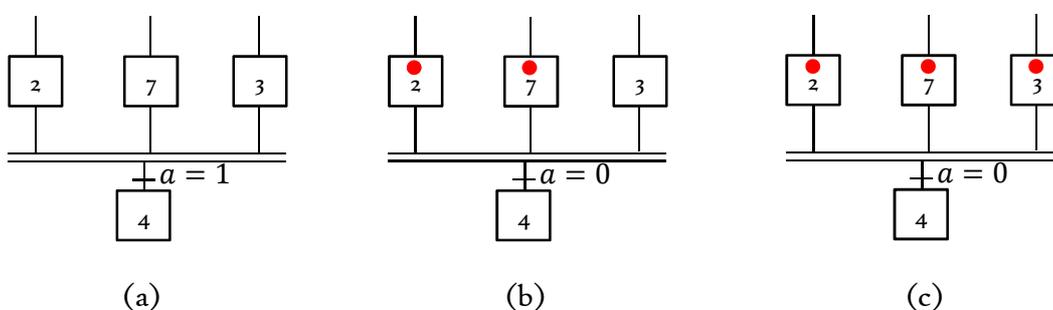


Figure III-31. Transition franchissable

III.12.3. Règle 3 : Franchissement d'une transition

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

Dans la Figure III-32, les étapes sont actives et $a = 1$, la transition est obligatoirement franchie, c'est à dire désactivation des étapes 2, 7 et 3 et l'activation de l'étape 4.

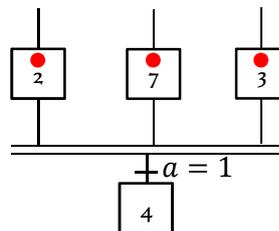


Figure III-32. Franchissement d'une transition

III.12.4. Règle 4 : Franchissements simultanés

Plusieurs transitions simultanément franchissables sont simultanément franchies (Fig. III-33).

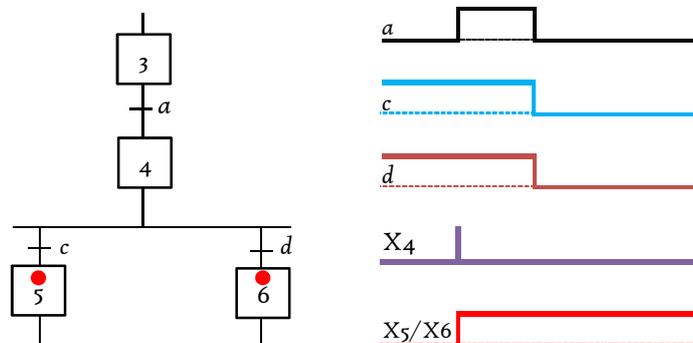


Figure III-33. Franchissements simultanés

III.12.5. Règle 5 : Activation et désactivation simultanées

Si au cours du fonctionnement, la même étape est simultanément activée et désactivée, elle reste active (Fig. III-34).

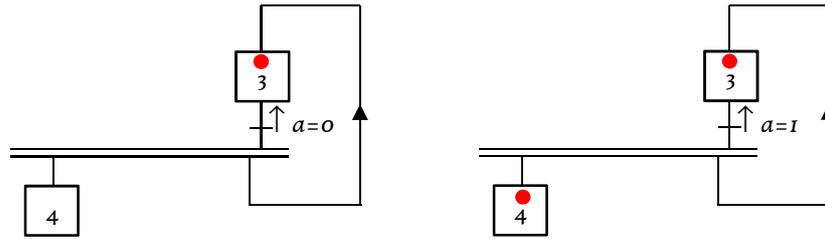


Figure III-34. Activation et désactivation simultanées

III.13. MISE EN ÉQUATION D'UN GRAFCET

On peut matérialiser un GRAFCET en utilisant un automate programmable industriel (API), à l'aide de séquenceurs de circuits logiques, circuits électriques etc. Par conséquent, le GRAFCET doit être traduit par des équations logiques décrivant les conditions d'activation et de désactivation de chaque étape.

Il existe plusieurs méthodes qui permettent d'obtenir les équations du GRAFCET. Les plus connues sont :

- ▶ la méthode des conditions du franchissement des transitions à *évolution synchrone* : méthode structurée et normalisée qui traduit directement les règles du GRAFCET
- ▶ la méthode "industrielle" à *évolution asynchrone* : cette méthode, bien que plus rapide à programmer, peut poser de problème pour quelques configuration du GRAFCET.

Ces méthodes passent par trois étapes :

- ❶ la recherche des *fonctions transitions* : des équations des conditions de franchissement des transitions du GRAFCET ;
- ❷ l'écriture des *équations des étapes* du GRAFCET : l'étape sera caractérisée par une mémoire (bascule RS), dont il faut préciser l'entrée (S) pour l'activation et l'entrée (R) pour la désactivation ;
- ❸ l'établissement des *équations des actions*.

III.13.1. Fonctions Transitions

Considérons la séquence de la Figure III-35 d'un GRAFCET. D'après la 2^{ème} règle d'évolution, on peut affirmer que :

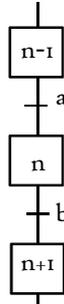


Figure III-34. Séquence d'un GRAFCET

- La transition $T_{n-1,n}$ peut être *franchie*, si elle est *validée* et la réceptivité associée est *vraie* (équation (III-1)) ;
- La transition $T_{n,n+1}$ peut être *franchie*, si elle est aussi *validée* et la réceptivité associée est *vraie* (équation (III-2)).

On définit alors deux conditions logiques dites "*fonctions transitions*" traduisant les affirmations précédentes :

$$X_{n-1,n} = X_{n-1} \cdot a \quad (\text{III-1})$$

$$X_{n,n+1} = X_n \cdot b \quad (\text{III-2})$$

III.13.2. Équations des Étapes

Une mémoire (SR) est caractérisée par l'équation (III-3) :

$$X = S + \bar{R}X \quad (\text{III-3})$$

L'équation de l'étape n peut être établie selon deux évolutions possibles :

III.13.2.1. Évolution synchrone

D'après la 3^{ème} règle d'évolution, on a :

- l'étape n s'*active* si la transition $T_{n-1,n}$ est *franchie* ;
- l'étape n se *désactive* si la transition $T_{n,n+1}$ est *franchie*.

Par conséquent, les entrées S et R de la mémoire seront :

$$S = Y_{n-1,n} \quad (\text{III-4})$$

$$R = Y_{n,n+1} \quad (\text{III-5})$$

L'équation X_n s'écrit alors :

$$X_n = Y_{n-1,n} + \overline{Y_{n,n+1}}X_n \quad (\text{III-6})$$

L'évolution est dite **synchrone** car le franchissement de la transition $Y_{n,n+1}$ entraîne simultanément l'activation de l'étape X_{n+1} et la désactivation de l'étape X_n .

III.13.2.2. Évolution asynchrone

Dans ce cas, la 3^{ème} règle n'est pas vérifiée. Les conditions d'activation et de désactivation sont comme suit :

- l'étape n s'**active** si la transition $T_{n-1,n}$ est **franchie** ;
- l'étape n se **désactive** si l'étape suivante X_{n+1} est **active**.

Par conséquent, les entrées S et R de la mémoire seront :

$$S = Y_{n-1,n} \quad (\text{III-7})$$

$$R = X_{n+1} \quad (\text{III-8})$$

L'équation X_n s'écrit alors :

$$X_n = Y_{n-1,n} + \overline{X_{n+1}}X_n \quad (\text{III-9})$$

L'évolution est dite **asynchrone** car l'activation de l'étape X_{n+1} entraîne la désactivation de l'étape X_n .

III.13.3. Équations des Actions

L'équation d'une action est la somme logique des variables X_j des étapes qui sont lui associés.

 **Exemple** : On considère le GRAFCET de la Figure III-35 :

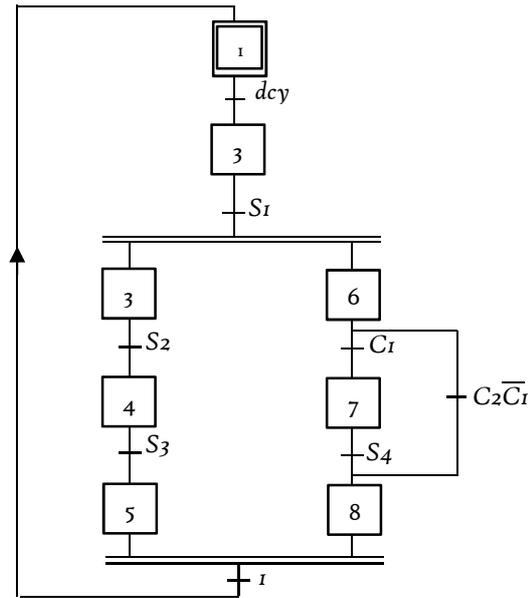


Figure III-35. GRAFCET

III.13.3.1. Fonctions transitions

$$Y_{12} = dcy \cdot X_1 \quad (\text{III-10})$$

$$Y_{236} = S_1 \cdot X_2 \quad (\text{III-11})$$

$$Y_{34} = S_2 \cdot X_3 \quad (\text{III-12})$$

$$Y_{34} = S_3 \cdot X_4 \quad (\text{III-13})$$

$$Y_{67} = C_1 \cdot X_6 \quad (\text{III-14})$$

$$Y_{68} = C_2 \bar{C}_1 \cdot X_6 \quad (\text{III-15})$$

$$Y_{78} = S_4 \cdot X_7 \quad (\text{III-16})$$

$$Y_{581} = X_5 \cdot X_8 \quad (\text{III-17})$$

III.13.3.2. Équations des étapes en évolution synchrone

$$\begin{cases} S = Y_{581} \\ R = Y_{12} \end{cases}; X_1 = S + X_1 \bar{R} + I_n \quad (\text{III-18})$$

$$\begin{cases} S = Y_{12} \\ R = Y_{236} \end{cases}; X_2 = S + X_2 \bar{R} \quad (\text{III-19})$$

$$\begin{cases} S = Y_{236} \\ R = Y_{34} \end{cases}; X_3 = S + X_3\bar{R} \quad (\text{III-20})$$

$$\begin{cases} S = Y_{34} \\ R = Y_{45} \end{cases}; X_4 = S + X_4\bar{R} \quad (\text{III-21})$$

$$\begin{cases} S = Y_{45} \\ R = Y_{581} \end{cases}; X_5 = S + X_5\bar{R} \quad (\text{III-22})$$

$$\begin{cases} S = Y_{236} \\ R = Y_{67} + Y_{68} \end{cases}; X_6 = S + X_6\overline{Y_{67} \cdot Y_{68}} \quad (\text{III-23})$$

$$\begin{cases} S = Y_{67} \\ R = Y_{78} \end{cases}; X_7 = S + X_7\bar{R} \quad (\text{III-24})$$

$$\begin{cases} S = Y_{68} + Y_{78} \\ R = Y_{581} \end{cases}; X_8 = S + X_8\bar{R} \quad (\text{III-25})$$

III.13.3.3. Équations des étapes en évolution asynchrone

$$\begin{cases} S = Y_{581} \\ R = X_2 \end{cases}; X_1 = S + X_1\bar{R} + I_n \quad (\text{III-26})$$

$$\begin{cases} S = Y_{12} \\ R = X_3 \cdot X_6 \end{cases}; X_2 = S + X_2(\bar{X}_3 + \bar{X}_6) \quad (\text{III-27})$$

$$\begin{cases} S = Y_{236} \\ R = X_4 \end{cases}; X_3 = S + X_3\bar{R} \quad (\text{III-28})$$

$$\begin{cases} S = Y_{34} \\ R = X_5 \end{cases}; X_4 = S + X_4\bar{R} \quad (\text{III-29})$$

$$\begin{cases} S = Y_{45} \\ R = X_1 \end{cases}; X_5 = S + X_5\bar{R} \quad (\text{III-30})$$

$$\begin{cases} S = Y_{236} \\ R = X_7 + X_8 \end{cases}; X_6 = S + X_6\bar{X}_7\bar{X}_8 \quad (\text{III-31})$$

$$\begin{cases} S = Y_{67} \\ R = X_8 \end{cases}; X_7 = S + X_7\bar{R} \quad (\text{III-32})$$

$$\begin{cases} S = Y_{68} + Y_{78} \\ R = X_1 \end{cases}; X_8 = S + X_8\bar{R} \quad (\text{III-33})$$

III.14. EXTENSION DU MODÈLE GRAFCET

III.14.1. Étapes Sources-Étapes Puits

III.14.1.1. Étapes sources

On appelle étape source une étape non reliée à une transition amont (l'étape X_5 dans la Figure III-36). Cette étape ne peut être activée que par un ordre de forçage. Une étape initiale sans transition amont est une étape source activée inconditionnellement à la mise sous tension [24].

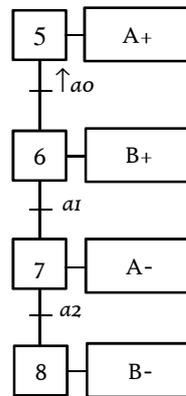


Figure III-36. Étapes sources-Étapes puits

Pour montrer la mise en équation d'une étape source, supposons que l'étape X_5 est forcée par l'étape X_{10} d'un GRAFCET maître, alors les équations seront :

$$Y_{56} = X_5 \cdot a_0 \quad (\text{III-34})$$

$$X_5 = X_{10} + X_5 \cdot \overline{Y_{56}} \quad (\text{III-35})$$

III.14.1.2. Étapes puits

On appelle étape puits une étape non suivie d'une transition (l'étape X_8 dans la Figure III-36). Cette étape ne peut être désactivée que par un ordre de forçage.

Supposons que l'étape X_8 soit désactivée par l'étape X_{20} d'un GRAFCET maître, alors les équations seront :

$$Y_{78} = X_7 \cdot a_2 \quad (\text{III-36})$$

$$X_8 = Y_{78} + X_8 \cdot \overline{X_{20}} \quad (\text{III-37})$$

III.14.2. Transitions Sources-Transitions Puits

III.14.2.1. Transitions sources

Une transition source est une transition non précédée d'une étape (transition $\uparrow a_0$ dans la Figure III-37). Par convention cette transition est toujours validée (malgré l'absence d'étape précédente), pour la franchir il suffit que la réceptivité soit vraie. La réceptivité associée à une transition source est en générale une réceptivité sur front (sinon l'étape suivante est activée en permanence).

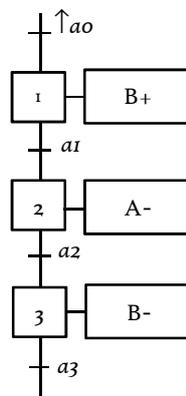


Figure III-37. Transitions sources-Transitions puits

Ainsi, les équations seront :

$$Y_{12} = X_1 \cdot a_1 \quad (\text{III-38})$$

$$X_1 = \uparrow a_0 + X_1 \cdot \overline{Y_{12}} \quad (\text{III-39})$$

III.14.2.2. Transitions puits

Une transition puits est une transition non suivie d'une étape (transition a_3 dans la Figure III-37). Les règles de franchissement s'appliquent à ce type de transition, lors du franchissement l'étape précédente est désactivée, aucune autre étape n'est activée.

Ainsi, les équations seront :

$$Y_{23} = X_2 \cdot a_2 \quad (\text{III-40})$$

$$Y_4 = X_4 \cdot a_3 \quad (\text{III-41})$$

$$X_4 = Y_{23} + X_4 \cdot \overline{Y_4} \quad (\text{III-42})$$

III.15. MACRO-ÉTAPE

III.15.1. Principe

Le concept de *macro-étape* (*ME*) est à différencier de celui de sous-programme, une *macro-étape* est plutôt une extension du concept d'étape. Les *macro-étapes* sont à utiliser pour faciliter la représentation de systèmes complexes dans un but de structuration. Le concept de *macro-étape* se prête aisément à l'analyse descendante du système. A chaque *macro-étape* correspond une *expansion* et réciproquement, si plusieurs cycles identiques sont à représenter, autant de macro et d'extension seront nécessaires [28].

III.15.2. Symbole

Une *macro-étape* (*ME*) est la représentation unique d'un ensemble d'étapes et de transition nommé *expansion d'étapes*, la macro-étape se substitue à une étape du GRAFCET (Fig. III-38).

- ❶ L'expansion de la ME comporte une étape d'entrée et une étape de sortie repérées par E et S.
- ❷ Tout franchissement de la transition amont de la macro-étape active l'étape E d'entrée de son expansion.
- ❸ L'étape de sortie participe à la validation des transitions aval de la macro-étape.
- ❹ La transition suivant la macro-étape n'est validée que lorsque la dernière étape de l'expansion de macro-étape est active.

Lorsque la transition $T_{I-M_{10}}$ est franchie (Fig. III-38), alors la macro-étape M_{10} est activée, ce qui entraîne l'activation de l'étape E_{10} de l'entrée de l'expansion et le cycle décrit dans l'expansion se déroule jusqu'à l'étape de sortie S_{10} . Dès que l'étape S_{10} est active, la transition $T_{M_{10}-2}$ sera franchie. Il est préférable de ne pas associer d'actions aux étapes d'entrée et de sortie de la macro-étape.

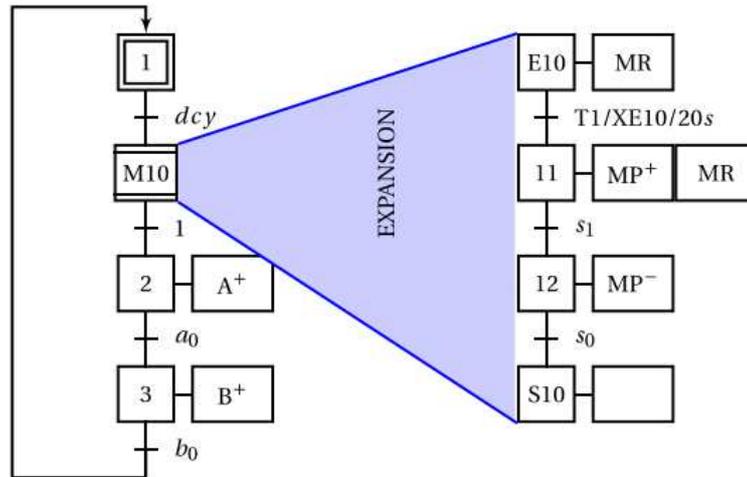


Figure III-38. Macro-étape et expansion

L'équation de M10 en évolution synchrone est :

$$X_{M10} = Y_{1-M10} + X_{M10} \cdot \overline{Y_{M10-2}} \quad (III-43)$$

$$\begin{cases} Y_{1-M10} = dcy \cdot X_1 \\ Y_{M10-2} = X_{M10} \cdot X_{S10} \end{cases} \quad (III-44)$$

Exemple : On considère le GRAFCET d'un doseur malaxeur de la Figure III-39 :

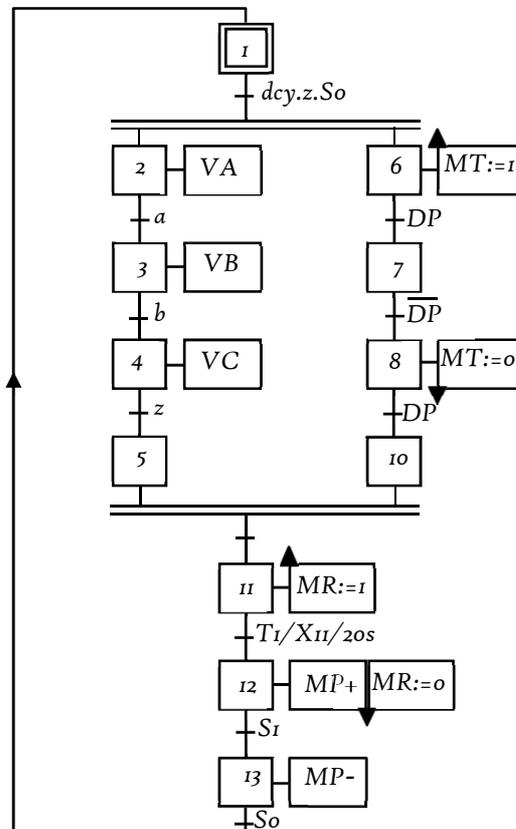


Figure III-39. GRAFCET d'un doseur malaxeur

la Figure III-40 représente un GRAFCET équivalent au GRAFCET de la Figure III-39 à base de macro-étapes :

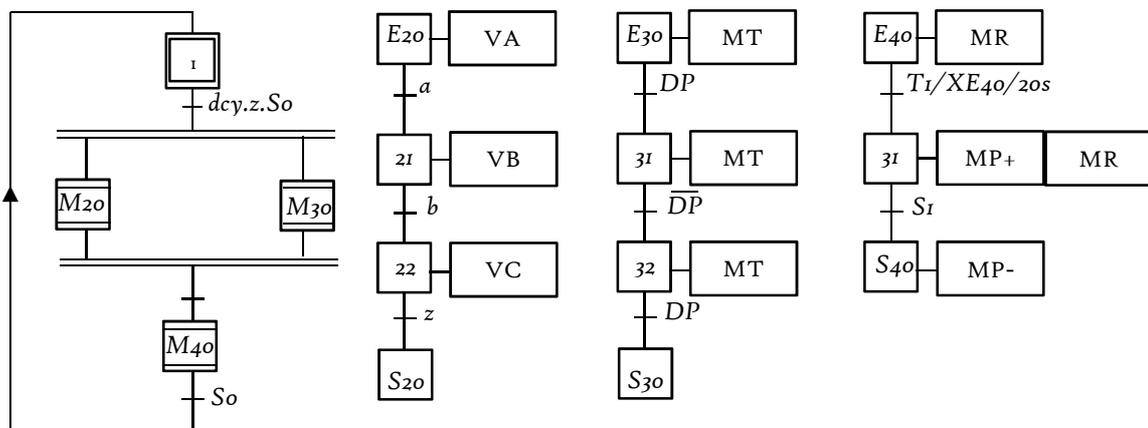


Figure III-40. GRAFCET d'un doseur malaxeur à base de macro-étapes

III.16. TACHES SOUS-PROGRAMMES

Le concept de tâche permet lui aussi de décrire les systèmes de façon structurée, il correspond à la notion de sous-programme (Fig. III-41).

Lorsqu'une tâche doit être réalisée plusieurs fois dans un cycle, la programmation sous forme de tâche permet de simplifier l'analyse et la programmation.

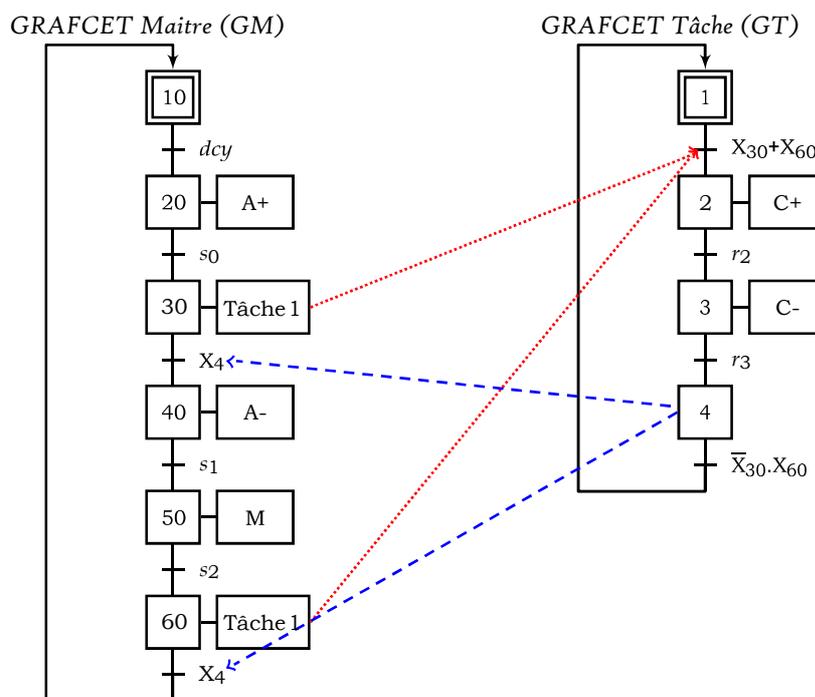


Figure III-41. Tâche sous-programme

Supposons que les GRAFCETs (GM) et (GT) sont initialisés.

- Si l'étape X_{30} est active, la transition T_{12} sera franchie et le GRAFCET (GT) s'exécute ;
- Si l'étape X_4 (étape vide) est active, la transition T_{3040} sera franchie et le GRAFCET(GM) continue son cycle ;
- Si l'étape X_{40} est active ($\overline{X_{30}} \cdot X_{60} = 1$), la transition T_{41} sera franchie et la GRAFCET(GT) s'initialise pour un nouveau cycle.

📌 **Exemple :** On reprend le GRAFCET de la Figure III-39 où la dernière séquence sera remplacée par une tâche (Fig. III-42).

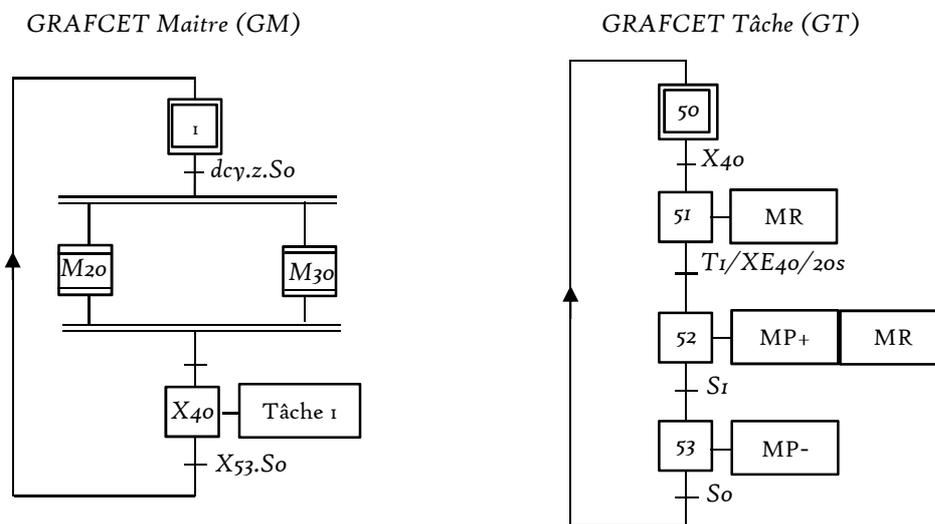


Figure III-42. GRAFCET avec des macro-étapes et tâche

III.17. FORÇAGE ET FIGEAGE

Le forçage n'est pas à proprement parlé une méthode de structuration des GRAFCETs, mais c'est un outil nécessaire à la hiérarchisation des GRAFCETs. La description du fonctionnement normal est facilitée par les outils d'analyse descendante, par contre les fonctionnements anormaux (sécurité, arrêt d'urgence) qui nécessite une prise de décision et une action rapide nécessite un outil capable d'inhiber le déroulement normal du GRAFCET, pour autoriser la réalisation de cycle de sécurité.

III.17.1. Forçage

- Par forçage de situation, on entend le passage imposé de la situation courante du GRAFCET désigné à une situation déterminée différente de celle qu'aurait atteint celui-ci s'il avait évolué normalement.
- Le GRAFCET forcé évolue alors sans franchissement de transition.
- L'ordre de forçage ne peut être émis que par un GRAFCET de niveau hiérarchiquement supérieur vers un GRAFCET de niveau inférieur. Ces ordres de forçage sont utiles principalement pour prendre en compte les sécurités.

III.17.2. Règles de Forçage

Une Le forçage est un ordre interne au GRAFCET consécutif à une évolution. Pour une situation comportant plusieurs ordres de forçage, les GRAFCET forcés prendront immédiatement et directement la ou les situations imposées.

- ❶ Toute apparition d'une situation de forçage est prioritaire par rapport à toute autre activité du GRAFCET ;
- ❷ Les GRAFCET forcés sont maintenus dans la situation de forçage tant que l'ordre de forçage est valide ;
- ❸ Un GRAFCET ne peut être forcé que par un GRAFCET hiérarchiquement supérieur ;
- ❹ A tout instant un GRAFCET ne peut être forcé que dans une situation et une seule à partir d'un et un seul GRAFCET supérieur.

III.17.3. Représentation du Forçage

De manière générale le forçage est représenté dans le GRAFCET par la séquence suivante : *F/nom du GRAFCET : (Situation)*.

Le forçage est inscrit dans un cadre d'action.

III.17.4. Types de Forçage

Le forçage décrit comment doit se retrouver le(s) GRAFCET(s) forcé(s) après le forçage, on distingue plusieurs situation types :

III.17.4.1. Forçage dans une situation non vide

- ⊙ Le *forçage* dans une situation non vide correspond à l'*activation* d'une ou plusieurs étapes dès que l'étape de forçage du GRAFCET supérieur est activée (Fig. III-43) ;
- ⊙ Il est possible de forcer plusieurs étapes. Le forçage est effectué en dehors des règles du GRAFCET ;
- ⊙ Ce forçage est principalement utilisé dans des procédures d'initialisation mais aussi pour activer un cycle particulier après un arrêt d'urgence ;
- ⊙ Pour que la situation évolue, il faut que le forçage cesse, sinon le GRAFCET forcé reste dans la situation de forçage.

III.17.4.2. Forçage dans une situation vide

- ⊙ Le *forçage* dans une situation vide correspond à la *désactivation* de toutes les étapes d'un GRAFCET dès que l'étape de forçage du GRAFCET supérieur est activée (Fig. III-43) ;
- ⊙ Le forçage dans une situation vide nécessite par la suite le forçage dans une *situation non vide* pour pouvoir *redémarrer* ;
- ⊙ Attention de ne pas forcer dans une situation vide tous les GRAFCETs du système (impossible de redémarrer).

Supposons que les étapes X_0 , X_{10} et X_{20} sont actives.

- ❖ Si on appuie sur le bouton d'Arrêt d'Urgence AU ($AU = 1$), il y aura :
 - ① l'étape X_1 s'active et un ordre de forçage apparaît ;
 - ② il s'agit d'un forçage des GRAFCETs (GC) et(GPN) dans une situation vide. Par conséquent toutes les étapes des GRAFCETs (GC) et (GPN) se désactivent.

- ❖ Si on relâche le bouton d'Arrêt d'Urgence AU (AU = o), il y aura :
 - ❶ l'étape X2 s'active et un ordre de forçage apparaît ;
 - ❷ il s'agit d'un forçage des GRAFCETs (GC) et (GPN) dans les étapes X10 et X20, respectivement. Par conséquent ces étapes s'activent, la transition T20 sera franchie et on reprend la situation de fonction normal quand l'étape X0 sera active.

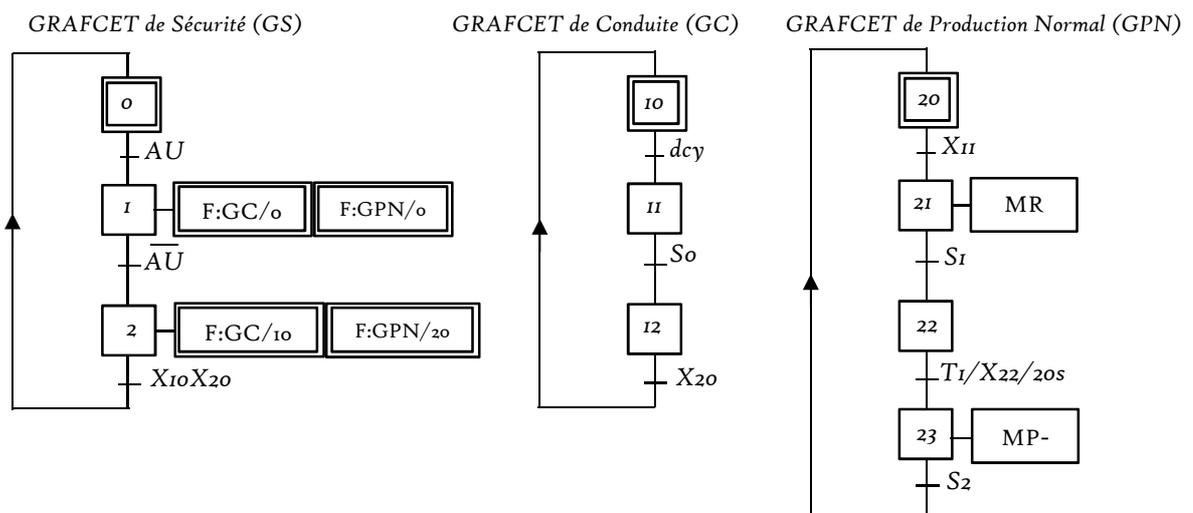


Figure III-43. Forçage de situation

III.17.5. Figeage

- Le figeage est un cas particulier du forçage dans lequel l'évolution normale du GRAFCET est figée, Le GRAFCET figée voit son évolution stoppée dès l'apparition de l'ordre de figeage ;
- Le figeage est représenté dans le GRAFCET par la séquence suivante : **F/nom du GRAFCET : (* situation)** ;
- Le figeage est à utiliser avec précaution, en effet les ordres émis par les étapes restent vrais indépendamment de l'évolution des réceptivités ;
- Il est souvent nécessaire de bloquer simultanément les sorties.

III.18. CONCLUSION

Le GRAFCET est un diagramme fonctionnel dont le but est de décrire graphiquement, suivant un cahier des charges, les différents comportements de

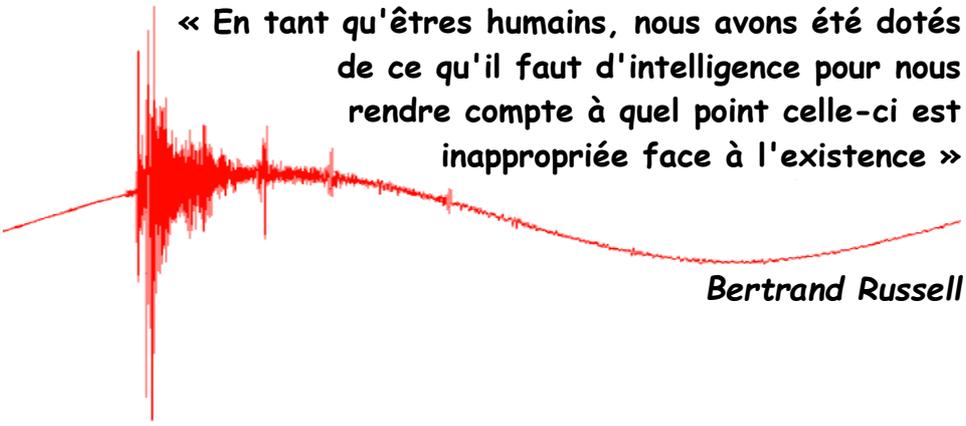
l'évolution d'un automatisme séquentiel. Il est à la fois simple à utiliser et rigoureux sur le plan formel et constitue un unique outil de dialogue entre toutes les personnes collaborant à la conception, à l'utilisation ou à la maintenance de la machine à automatiser.

Un des points forts du GRAFCET est la facilité de passer du modèle à l'implantation technologique de celui-ci dans un automate programmable industriel. Le GRAFCET passe alors du langage de spécification au langage d'implémentation utilisé pour la réalisation de l'automatisme.

Le chapitre suivant sera donc consacré au GEMMA (Guide d'Etude des Modes de Marches et d'Arrêts) d'un système automatisé de production.

CHAPITRE IV

GEMMA



« En tant qu'êtres humains, nous avons été dotés de ce qu'il faut d'intelligence pour nous rendre compte à quel point celle-ci est inappropriée face à l'existence »

Bertrand Russell

CHAPITRE IV

GEMMA

IV.1. INTRODUCTION

Les études faites précédemment dans le troisième chapitre sur le GRAFCET, le cahier des charges était volontairement réduit à sa plus simple expression, c'est-à-dire, à ce que l'on appelle les *spécifications fonctionnelles*. Celles-ci permettent de décrire le fonctionnement attendu du système à une situation de *production normale*, pendant laquelle il n'y a pas d'incident, nous appellerons GRAFCET de *fonctionnement normale (GFN)*.

Cependant, les systèmes automatisés sont pratiquement sujets de *pannes*, de *réparations* et d'*arrêts*. Ces situations inévitables dans la vie d'une usine et conduisant aux *modes de marches particuliers*, doivent être considérées lors de la conception et au cours de l'exploitation de chaque automatisme. Les exemples typiques de modes de marches particuliers sont : vérification manuelle étape par étape, préparation pour mise en route, et *arrêt d'urgence* à la suite d'une *défaillance*.

Le **GEMMA** (**G**uide d'**E**tude des **M**odes de **M**arches et d'**A**rrêts) est un "*outil-méthode*" permettant de mieux définir les Modes de Marches et d'Arrêts d'un système industriel automatisé. Il est constitué d'un guide graphique qui sera rempli progressivement lors de la conception du système ce qui permet de sélectionner et de décrire les différents états de marches et d'arrêts ainsi que les possibilités d'évoluer d'un état à un autre.

Le GEMMA n'intervient donc que si la partie commande est en énergie et active. Le GEMMA définit donc, à tout instant, l'état dans lequel se trouve la partie commande du système automatisé de production.

Dans ce chapitre je présente le concept et structuration du GEMMA et son mode d'emploi, procédures de fonctionnement, d'arrêt et les procédures en

défaillances et l'utilisation pratique du GEMMA et applications. Les différents boucles opérationnelles associées aux GEMMA sont définis, ainsi que les GRAFCETs associés aux Guide d'Etude des Modes de Marches et d'Arrêts.

IV.2. BESOIN D'OUTILS-MÉTHODES

L'automatisation de la production exige la réalisation de machines automatiques de plus en plus complexes mais en même temps offrant plus de sécurité et une plus grande souplesse d'emploi.

Par ailleurs, un système automatisé est un carrefour : pour le concevoir, le réaliser, le mettre au point, le réparer, le faire évoluer, des hommes ayant des compétences et des motivations diverses doivent se comprendre [22,29].

Par conséquent, le développement de la production automatisée passe par la création, la promotion et la pratique d'outils-méthodes utilisant des concepts clairement définis et facilitant la conception, la réalisation et l'exploitation des machines automatiques [30].

Ressenti d'abord au niveau de la description du fonctionnement, ce besoin d'outils-méthodes a donné naissance au GRAFCET. Des outils-méthodes complémentaires se révèlent maintenant nécessaires [22].

IV.3. BESOIN D'UN VOCABULAIRE PRÉCIS

Dans le domaine des Modes de Marches et d'Arrêts, le vocabulaire pratiqué est imprécis et parfois même contradictoire : il conduit à des incompréhensions graves.

Par exemple, très utilisés des praticiens les termes "*Marche Automatique, Semi-Automatique, Manuelle*" recouvrent des notions toutes relatives : selon son expérience et son environnement, chaque technique, chaque société et chaque individu leurs donnent des significations différentes. Ainsi, une *marche semi-automatique* sera souvent considérée comme une *marche automatique* pour des systèmes moins complexes.

Par ailleurs, la liste des termes utilisés pour dénommer les *Modes de Marches* s'allonge sans pour autant apporter d'idée unificatrice. Quelques termes utilisés, à titre d'exemple : Marche "*pas à pas*", Marche "*coup par coup*", Marche "*d'intervention*", Marche "*réglage*", Marche "*cycle par cycle*", ... Arrêt "*d'urgence*", Arrêt "*figeage*", Arrêt "*fin de mouvement*", Arrêt "*à l'étape*" ... etc [22].

Par conséquent, pour que toutes les personnes concernées se comprennent, il est indispensable de définir un vocabulaire précis en le rattachant à des critères fondamentaux, indépendants du genre de l'équipement et de la technologie de réalisation.

IV.4. BESOIN D'UNE APPROCHE GUIDÉE

Le plus souvent lors de l'étude d'un système automatisé, les besoins en Modes de Marches et d'Arrêts sont peu ou mal exprimés. Après réalisation du système, c'est alors au prix de modifications et tâtonnements coûteux qu'il faut répondre à ces besoins essentiels.

Le concepteur a donc besoin d'une approche guidée et systématique, du genre "check-list", pour tout prévoir dès l'étude et envisager les conséquences, tant pour la partie opérative que pour la partie commande du système à réaliser [22].

Le GRAFCET permet une expression précise de certains *Modes de Marches et d'Arrêts*. Seul, il ne permet cependant pas l'approche systématique et globale nécessaire. Pour les Modes de Marches et d'Arrêts, le GEMMA répond à ces besoins : c'est un outil-méthode qui définit un vocabulaire précis, en proposant une approche guidée systématique pour le concepteur : le guide graphique GEMMA.

IV.5. QU'EST QUE LE GEMMA ?

Le GEMMA (Guide d'Etude des Modes de Marche et d'Arrêt) est un guide graphique permettant de sélectionner et de décrire simplement les différents états de *marches* et d'*arrêts*, ainsi que les possibilités d'*évoluer* d'un *état* à un *autre*. Il a

été créé parce qu'il y avait un grand besoin d'avoir un vocabulaire commun et précis. Le GEMMA permet d'avoir une approche guidée de l'analyse des modes de marches et d'arrêts.

Le GEMMA permet le recensement et la *description des différents états* du système automatisé, de la mise en route à la *production normale*. Il précise les *procédures* à mettre en œuvre après analyse d'une *anomalie* ou un *défaut* de fonctionnement. Pour une machine donnée, les modes de marches et d'arrêts doivent être choisis et compris de toutes les personnes chargées d'intervenir.

Un GEMMA est établi pour chaque machine lors de sa conception, puis utilisé tout au long de sa vie : réalisation, mise au point, maintenance, modifications, réglages ... Dans ses principes et dans sa mise en œuvre, le GEMMA doit donc être connu de toutes les personnes concernées par les automatismes, depuis leur conception jusqu'à leur exploitation.

IV.6. CONCEPTS DE BASE DU GEMMA

IV.6.1. Système en Ordre de Marche

Le guide graphique GEMMA, est constitué de deux zones :

IV.6.1.1. Partie commande hors énergie

Dans cet état la partie opérative n'est pas sous le contrôle de la partie commande. La partie opérative peut être en énergie ou hors énergie. La sécurité est garantie par les choix technologiques et la procédure de mise en énergie de la partie opérative. Cette zone du GEMMA, située à l'extrême gauche, correspond à l'état inopérant de la partie commande. Elle ne figure que pour la forme.

IV.6.1.2. Partie commande sous énergie

C'est la partie qui va nous permettre de décrire ce qui se passe lorsque la partie commande et de définir les différents modes de marche et d'arrêt de notre machine ainsi que les conditions de passage d'un mode à l'autre.

Cette partie est subdivisée en trois zones ou en trois familles de procédures.

IV.6.2. Marche en Production

Un système industriel automatisé est conçu fondamentalement pour produire une certaine *valeur ajoutée*. C'est la justification principale de la construction du système. On dira que le système est en *production* si la valeur ajoutée pour laquelle le système a été conçu est obtenue. On dira que le système est *hors production* dans le cas contraire. La zone de production se situe dans la zone partie commande (PC) sous énergie.

IV.6.3. Familles de Modes de Marches et d'Arrêts

Le GEMMA se présente sous la forme d'un guide graphique (Fig. IV-1). Il contient deux zones : Partie Commande hors énergie qui ne figure que par principe et Partie Commande sous énergie qui couvre la quasi-totalité du guide et qui est également subdivisée en deux zones : Machine en Production si la valeur ajoutée pour laquelle la machine a été conçue est obtenue (zone délimitée par un rectangle double en traits discontinus) et hors production dans le cas contraire [11,20].

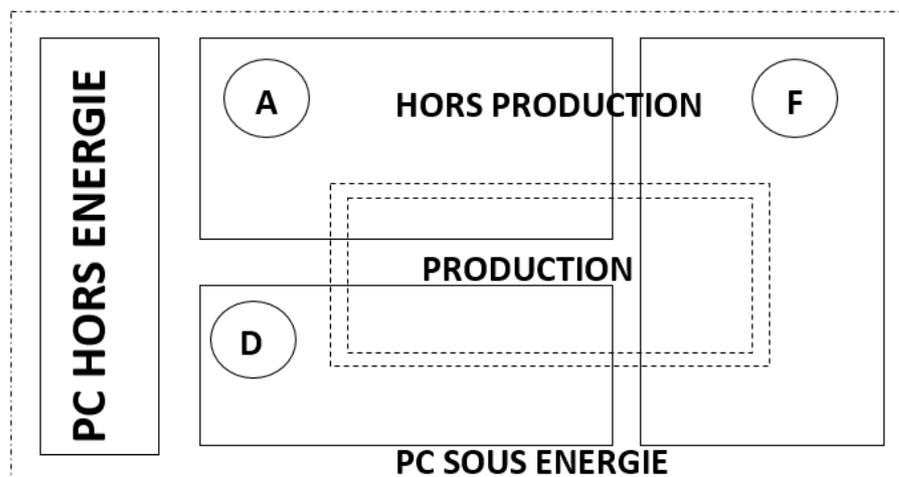


Figure IV-1. Description générale du guide GEMMA

Il propose différents modes de marches et d'arrêts. A chacun de ces modes de marches et d'arrêts correspond un rectangle état (Fig. IV-2).

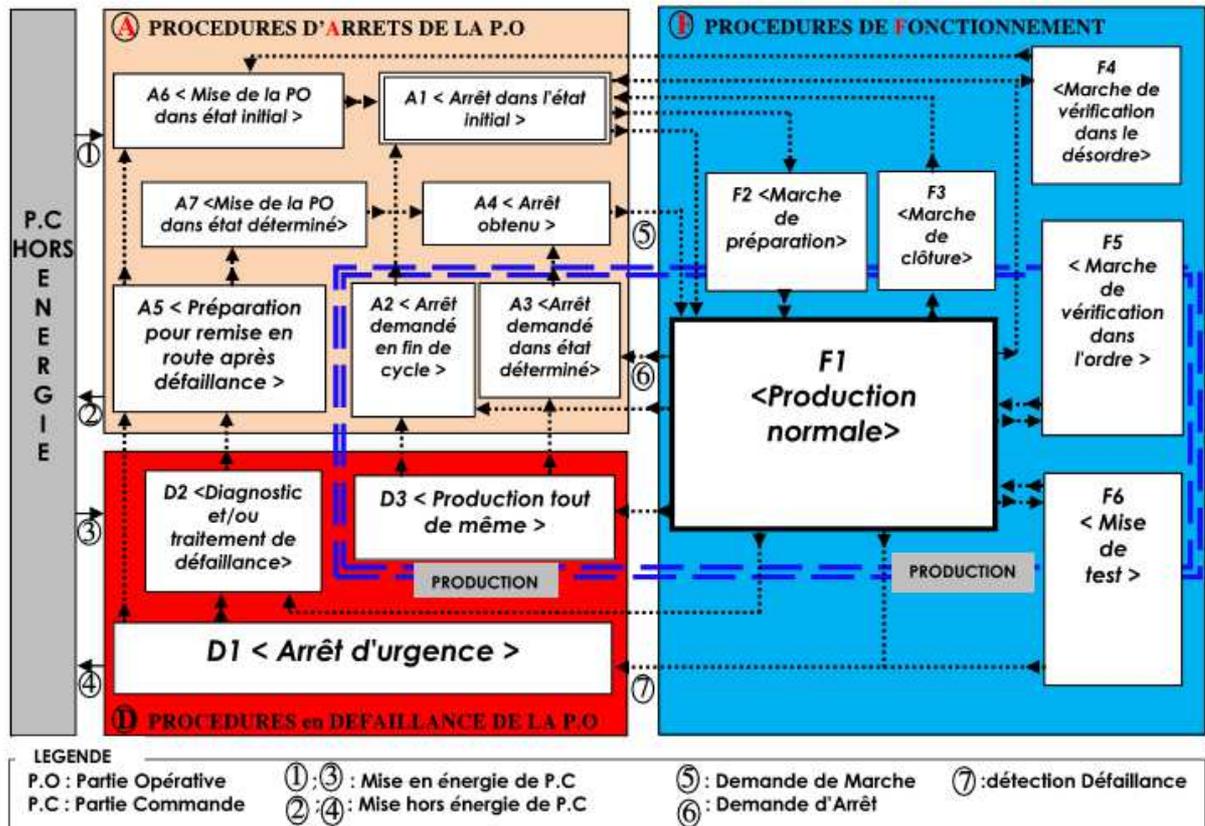


Figure IV-2. Guide d'Etude des Modes de Marches et d'Arrêts

On peut classer en trois grandes familles ou états (A, F et D) les modes de marches et d'arrêts d'un système automatisé (Fig. IV-2).

IV.6.3.1. Rectangle-état

Sur le guide graphique GEMMA, chaque mode de marche ou de procédure en défaillance peut être décrit à l'intérieur d'une case du GEMMA appelée rectangle-état prévus à cette fin. Ce dernier traduit un état du système à un moment donné de son évolution (Fig. IV-3).

La position d'un rectangle-état sur le guide graphique définit :

- Son appartenance à l'une des trois familles (A, F et D), procédure de fonctionnement, d'arrêt ou de défaillance.
- Le fait qu'il soit en ou hors production.

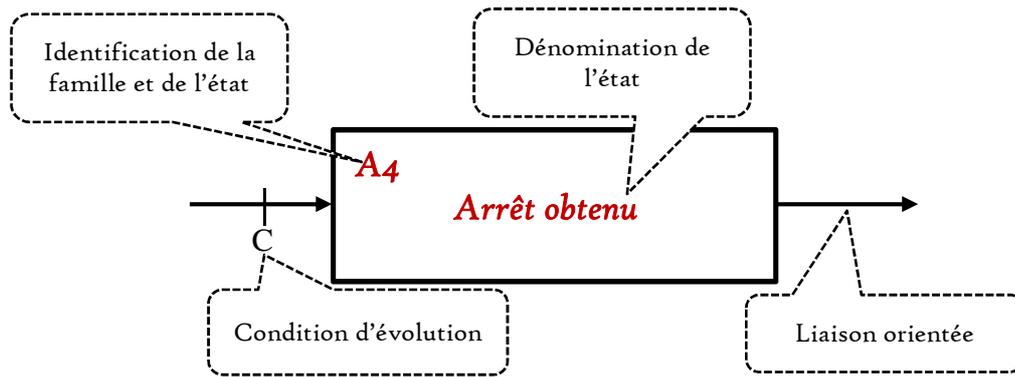


Figure IV-3. Rectangle-état

IV.6.3.2. Procédures d'Arrêt ou les états A_i de la zone A

Les états A_i de la zone A sont les *procédures d'Arrêt* et de remise en route de la partie opérative. Une machine fonctionne rarement en continu, c'est pour cela qu'il faut prévoir des arrêts en fin de journée ou arrêts pour réapprovisionnement [20].

Une marche automatique fonctionne rarement 24 heures sur 24. Il est nécessaire de l'arrêter de temps à autre, pour des raisons extérieures au système (la journée est finie) ou bien par manque d'approvisionnement. On classera dans cette famille tous les modes conduisant à (ou traduisant) un état d'arrêt du système pour des raisons extérieures. Ils sont regroupés dans une *zone A*, "*procédures d'Arrêt*", du guide graphique.

On regroupe tous les états du système automatisé qui traduisent un arrêt pour des raisons extérieures au système. Ce sont les arrêts normaux.

► *État A_1 : Arrêt dans l'état initial*

C'est l'*état Repos* de la machine, il correspond en général à la *situation initiale* du GRAFCET : c'est pour cela que ce rectangle est repéré par un double encadrement comme une étape initiale. Cet état est utile dans la description de tout système.

Pour une étude plus facile de l'automatisme, il est recommandé de représenter la machine dans son état initial [20,30].

➤ **État A2 : Arrêt demandé en fin de cycle**

Lorsque l'*arrêt est demandé*, la machine continue à produire jusqu'à la *fin du cycle*. C'est un état transitoire vers l'état A1.

➤ **État A3 : Arrêt demandé dans état déterminé**

La machine continue de produire jusqu'à un *arrêt* en une position *autre* que la *fin du cycle*. C'est un état transitoire vers l'état A4.

➤ **État A4 : Arrêt obtenu**

La machine est *arrêtée* en une autre position que la *fin du cycle*.

➤ **État A5 : Préparation pour remise en route après défaillance**

Dans cet état, on procède à toutes les opérations nécessaires (dégagements, nettoyages, ... etc) à une remise en route *après défaillance*.

➤ **État A6 : Mise de la PO dans état initial**

Dans cet état, la machine étant en A6, on remet manuellement et/ou automatiquement la PO en position pour un redémarrage dans l'*état initial*.

➤ **État A7 : Mise de la PO dans état déterminé**

Dans cet état, on remet la PO en position pour un redémarrage dans une position *déterminé autre que l'état initial*.

IV.6.3.3. Procédures de fonctionnement ou les états Fi de la zone F

Les *états Fi* de la zone F sont les *procédures de marches* ou de *fonctionnement* [20].

Les procédures de fonctionnement définissent les états de fonctionnement du système. Ils sont au nombre de **6 états**. On regroupe dans cette famille tous les modes ou états qui sont indispensable à l'obtention de la valeur ajoutée. Dans ces rectangles d'état, le système peut produire, mais on peut aussi le régler, le tester. C'est dans ces procédures que l'on trouve le rectangle d'état caractérisant la production normale de tout système.

Donc il peut s'agir :

- ☞ Procédures préparatoires à la production ;
- ☞ Réglages, tests ... qui sont néanmoins indispensables à la production.

La sous états F_1 , F_2 et F_3 assure la mise en/ou hors service de la production normale. Les autres états F_4 , F_5 et F_6 permet les essais et les vérifications avant ou après le passage en production normale.

➤ **État F_1 : Production normale**

Dans cet état, la machine produit normalement : c'est l'état pour lequel elle a été conçue. C'est pour cela que ce *rectangle* a un cadre particulièrement *renforcé*. On peut souvent faire correspondre à cet état un GRAFCET que l'on appelle *GRAFCET de Production Normale (GPN)*. On parle souvent de *mode automatique*.

➤ **État F_2 : Marche de préparation**

Cet état est utilisé pour les machines nécessitant une *préparation* préalable à la production normale : préchauffage d'un four jusqu'à atteindre la température désirée, remplissage de bacs, ...etc. C'est un état transitoire vers l'état F_1 .

➤ **État F_3 : Marche de clôture**

C'est l'état nécessaire pour certaines machines devant être vidées, nettoyées, etc. en fin de journée ou en fin d'une série.

➤ **État F_4 : Marche de vérification dans le désordre**

Cet état permet de vérifier certaines fonctions ou certains mouvements sur la machine, *sans respecter* l'ordre du cycle. On peut associer à cet état un *GRAFCET de Marche Manuelle (GMM)*.

➤ **État F_5 : Marche de vérification dans l'ordre**

Dans cet état, le cycle de production peut être exploré au *rythme voulu* par la personne effectuant la vérification, selon les cas, la machine peut produire ou non.

➤ *État F6 : Marche de test*

Cet état permet des opérations de *réglage*. Plusieurs machines comportent des capteurs qui doivent être réglés ou étalonnés périodiquement, cet état permet ces opérations de réglage ou d'étalonnage.

IV.6.3.4. Procédures de défaillances ou les états Di de la zone D

Les *états Di de la zone D* sont les *procédures de défaillances* de la partie opérative. Il est rare qu'un système fonctionne sans incident durant toute sa vie. Il faut envisager qu'il aura des défaillances prévisibles ou imprévisibles. Un système ne pourra pas fonctionner sans incidents, il faut prévoir des états pour *traiter les défaillances* [20].

On regroupera dans cette famille tous les modes conduisant à un état d'arrêt du système pour des raisons intérieures au système, autrement dit, à cause de défaillances de la partie opérative.

Les procédures de la *famille D* ont pour objectif de limiter au maximum les conséquences ou risques pour le personnel ou matériel.

Lors du fonctionnement d'un système, il peut se produire des incidents : on est donc conduit à prévoir les *défaillances* inhérentes ou internes au système. Tous les modes conduisant (ou traduisant) un état d'arrêt du système pour des raisons internes sont consignés dans la *zone D* : "*Procédures de défaillance*" du guide. Cette zone répond donc à la *sécurité* du matériel et du personnel qui constitue un souci lors de la conception d'un automatisme. Dans le cas d'un fonctionnement anormal ou dangereux, l'opérateur doit disposer du pouvoir d'arrêter l'évolution du cycle par une simple action et reprendre le contrôle en mode manuel.

Les procédures en défaillance définissent les états que devra avoir la partie opérative en cas de défaillance.

Les *procédures de défaillance* sont en nombre de *trois états*. La description de cette famille de procédure est donnée ci-dessous :

➤ *État D₁ : Arrêt d'urgence*

C'est l'état pris lors d'un *arrêt d'urgence*. On y prévoit non seulement les arrêts, mais aussi les cycles de dégagements, les procédures et précautions nécessaires pour limiter le danger et les conséquences *dues à la défaillance*.

L'*arrêt d'urgence* peut être déclenché de deux manières :

- ☞ Soit par l'opérateur (action manuelle) ;
- ☞ Soit par l'apparition d'un signal de sécurité.

➤ *État D₂ : Diagnostic et/ou traitement de défaillance*

C'est dans cet état que la machine peut être *examinée après défaillance* et qu'il peut être apporté un *traitement* permettant le redémarrage.

➤ *État D₃ : Production tout de même*

Il est parfois nécessaire de *continuer* la production *même après défaillance* de la machine : on aura alors une *production dégradée* ou une *production forcée* ou aidée par des opérateurs non prévus en production normale.

IV.7. MODE D'EMPLOI DU GEMMA

L'étude des modes de marches et d'arrêts est prévue dès la conception de la machine et intégrée dans sa réalisation. Sa mise en œuvre aura lieu après que les spécifications fonctionnelles ont été analysées par les GRAFCETs. Ces spécifications fonctionnelles ne concernent que le fonctionnement normal.

Le GEMMA s'attarde sur les spécifications opérationnelles qui en visage les différents *modes de marches et d'arrêts*, ainsi que les divers cas de *défaillance*. Chaque rectangle d'état est défini par son appartenance à l'une des trois familles de procédures (*A*, *F* et *D*), son appartenance ou non à la zone de production, son repère et sa désignation. L'automaticien peut compléter la désignation de chaque rectangle d'état.

IV.7.1. Sélection des Modes de Marches et d'Arrêts

Dans un *premier temps* l'automaticien va choisir les rectangles d'état nécessaires à la description du système automatisé de production. Pour chaque rectangle d'état choisi il donnera une description précise soit de l'état de la partie opérative soit des procédures correspondantes au fonctionnement attendu.

Après avoir lu le cahier des charges :

- si le mode proposé *est retenu*, il sera *précisé* en langage littéral de fonctionnement propre à la machine, dans le *rectangle d'état*. Retenir les états qui sont nécessaires pour le système. Par exemple, le système aurait-il besoin d'une marche de préparation, est ce qu'il y a un intérêt de l'arrêter dans un état autre que l'état initial, est ce qu'une marche manuelle est prévue ... etc.
- si le mode proposé *n'est pas nécessaire* pour la machine, une *croix* est portée dans le *rectangle d'état*, pour bien signifier qu'il n'est pas retenu.

IV.7.2. Conditions d'Évolution entre les Modes de Marches et d'Arrêts

Dans un *deuxième temps*, l'automaticien précisera les *conditions de passage* d'un rectangle d'état à l'autre. Il établira ainsi plusieurs boucles opérationnelles. Une boucle opérationnelle est une succession d'états caractérisant le fonctionnement du système automatisé de production.

L'élaboration de ces conditions de passage rend possible la conception du pupitre de commande et entraîne éventuellement l'adjonction de capteurs supplémentaires. Le passage d'un état vers l'autre s'effectue de deux façons :

- soit avec une condition d'évolution qui est portée sur la liaison orientée entre les deux états et concrétisée par un capteur sur machine ou par un auxiliaire de commande sur pupitre ;
- soit sans condition d'évolution (si cela n'apporte aucune information complémentaire).

IV.8. SÉLECTION DES MODES ET DES ÉVOLUTIONS

Pour utiliser le GEMMA, il faut commencer dans un premier temps de regarder chacun des rectangles-états et se demander quels sont ceux qui s'appliquent à l'automatisme qui est analysé. Il faut donc envisager tous les états possibles. Si le mode proposé est retenu, il sera précisé en langage littéral de fonctionnement propre à la machine (langage machine) dans le rectangle-état. Si au contraire le mode proposé n'est pas nécessaire pour la machine, une croix est portée dans le rectangle-état, pour bien signifier qu'il n'est pas retenu.

Deux états essentiels, définis dès le début de l'étude, se retrouvent sur toutes les machines :

- ⊙ L'état A_I , dit "*arrêt dans état initial*", ou "*état repos*" de la machine ;
- ⊙ L'état F_I , mode de "*production normale*" pour lequel la machine a été conçue.

En partant de chacun des deux états essentiels, A_I et F_I , on recherche les évolutions vers les autres états. On se demande d'abord quelle évolution suivre lors du démarrage. Le choix est :

- ⊙ $A_I \rightarrow F_I$: démarrage sans marche de préparation ;
- ⊙ $A_I \rightarrow F_2 \rightarrow F_I$: démarrage avec marche de préparation.

Ce qui permet de répondre à la question suivante : *une marche de préparation est-elle nécessaire ?* Ensuite, on se demande quelle évolution suivre lors de l'arrêt normal de production. Le choix est alors :

- ⊙ $F_I \rightarrow A_2 \rightarrow A_I$: arrêt en fin de cycle sans marche de clôture ;
- ⊙ $F_I \rightarrow F_3 \rightarrow A_I$: arrêt avec une marche de clôture ;
- ⊙ $F_I \rightarrow A_3 \rightarrow A_4$: arrêt dans un état autre que la condition initiale.

Ce qui permet de répondre aux questions suivantes :

- ☞ Une marche de clôture est-elle nécessaire ?
- ☞ Un arrêt dans un état autre que la condition initial est-il nécessaire ?

Puis, on se demande quelle évolution suivre lors d'une défaillance de l'automatisme. Un grand nombre de choix est disponible :

- ⊙ $F_1 \rightarrow D_3$: défaillance légère permettant une marche de production tout de même ;
- ⊙ $D_1 \rightarrow D_2 \rightarrow A_5 \rightarrow A_7 \rightarrow A_4 \rightarrow D_3$: arrêt d'urgence puis évolution pour production tout de même (défaillance légère impliquant l'arrêt d'un poste);
- ⊙ $D_1 \rightarrow A_5 \rightarrow A_6 \rightarrow A_1$: arrêt d'urgence puis évolution pour un arrêt en condition initiale (défaillance majeure) ;
- ⊙ $D_1 \rightarrow A_5 \rightarrow A_7 \rightarrow A_4$: arrêt d'urgence puis évolution pour un arrêt dans le même état que lors de l'apparition de l'arrêt d'urgence (défaillance mineure);
- ⊙ $D_1 \rightarrow D_2 \rightarrow A_5 \rightarrow A_6 \rightarrow A_1$: arrêt d'urgence avec diagnostic et traitement, puis évolution pour un arrêt en condition initiale (défaillance majeure) ;
- ⊙ $D_1 \rightarrow D_2 \rightarrow A_5 \rightarrow A_7 \rightarrow A_4$: arrêt d'urgence avec diagnostic et traitement, puis évolution pour un arrêt dans le même état que lors de l'arrêt d'urgence (défaillance mineure).

IV.9. EFFET DU GEMMA SUR LE GRAFCET

Le GEMMA affecte directement le GRAFCET de niveau 2, car il envisage tous les cas de fonctionnement autres que le fonctionnement normal qui est couvert par le GRAFCET.

Cette combinaison du GRAFCET et du GEMMA mène à deux approches possibles :

- ☞ Enrichissement du GRAFCET ;
- ☞ Découpages en tâches.

IV.10. FINALITÉ DU GEMMA

IV.10.1. GRAFCET de Conduite

Le **GRAFCET de Conduite** (GC) doit permettre d'assurer la bonne marche du système automatisé en intégrant les dispositions précisées par le GEMMA lors de l'étude des modes de marches et d'arrêts.

IV.10.2. Pupitre de Commande

La conception et l'organisation du pupitre de commande découle directement des conditions d'évolution des modes de marches et d'arrêts formalisés par le GEMMA. Le choix et la localisation des organes de dialogue (boutons, voyants, etc.) tiennent compte des spécifications de conduite et de sûreté, des règles de l'art (normes, réglementations) et des objectifs ergonomiques.

IV.11. BOUCLES OPÉRATIONNELLES

Sur le document GEMMA, plusieurs boucles peuvent être caractérisées. Une boucle est une succession d'états caractérisant le fonctionnement d'un système automatisé. Le passage d'un état à un autre doit se faire par respect des conditions d'évolution, mais il arrive parfois qu'il soit impossible de passer d'un état à un autre sans utiliser un état intermédiaire.

Le passage d'un état à l'autre s'effectue de deux façons :

- Soit **avec une condition d'évolution** : la condition peut être liée à l'action sur un bouton du pupitre de commande, ou à l'actionnement d'un capteur situé sur la machine ;
- Soit **sans condition d'évolution** : dans certains évolutions entre états, l'écriture d'une condition n'apporterait aucune information utile : c'est le cas lorsque celle-ci est évidente, ou parce que l'état atteint dépend de l'intervenant.

IV.II.1. Marche de Production Automatique

Ce GEMMA correspond au cas où la machine produit en *mode automatique*. La condition de mise en marche est que le sélecteur de mode soit à automatique, que le bouton de départ de cycle soit appuyé et que la machine soit en condition initiale. Lorsque la machine produit, elle peut s'arrêter en fin de cycle suite à l'appui du bouton d'arrêt de cycle (*acy*) (Fig. IV-4).

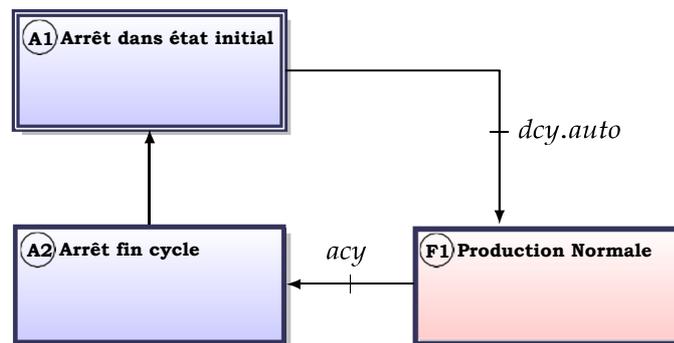


Figure IV-4. Marche de production automatique

IV.II.2. Marche Automatique avec Arrêt d'Urgence

Dans le cas où un *arrêt d'urgence* causé par une *défaillance grave* est envisagé, il faut pouvoir aller au rectangle-état *DI* lorsque cet arrêt d'urgence se produit. Et cela, quelque soit le rectangle-état ou la machine se situe.

Pour éviter d'encombrer le GEMMA, on met simplement en évidence l'évolution de *FI* vers *DI* et on ajoute un symbole de regroupement avec la mention *depuis tous les états*. Après l'arrêt d'urgence, il faut préparer la machine à sa remise en route en la nettoyant ou en dégageant les pièces coincées (*F5*), puis remettre la partie opérative en condition initiale de façon manuelle ou par une initialisation automatisée (*F6*). La machine sera alors prête à être redémarrée (Fig. IV-5).

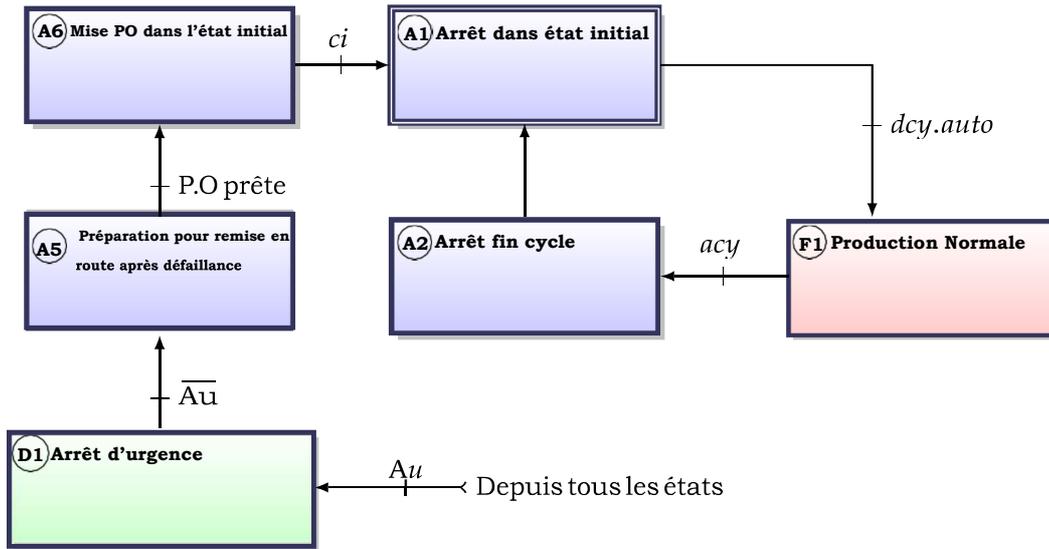


Figure IV-5. Marche automatique avec arrêt d'urgence

IV.11.3. Marche de Production Cycle par Cycle

Ce GEMMA correspond au cas où la machine produit uniquement à la pièce. Ce mode de fonctionnement est appelé mode semi-automatique. Le GRAFCET résultant est directement le GRAFCET de base. La *condition de mise en marche*, c'est que la machine soit en condition initiale et que l'opérateur appui sur le bouton de départ de cycle (Fig. IV-6).

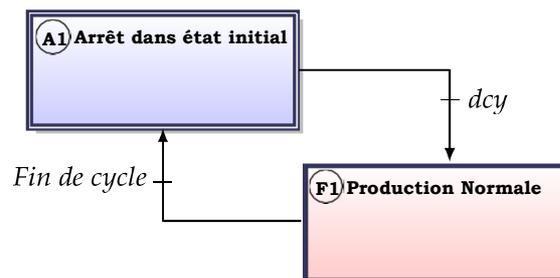


Figure IV-6. Condition dcy de mise en marche de production cycle par cycle

IV.11.4. Marche de Production à Cycle Répété

Ce GEMMA correspond au cas où la machine produit uniquement après la consigne de départ donnée par l'opérateur, les cycles se succèdent sans intervention de celui-ci. La *condition de mise en marche*, c'est que la machine soit

en condition initiale et que l'opérateur appui sur le bouton de départ de cycle (Fig. IV-7).

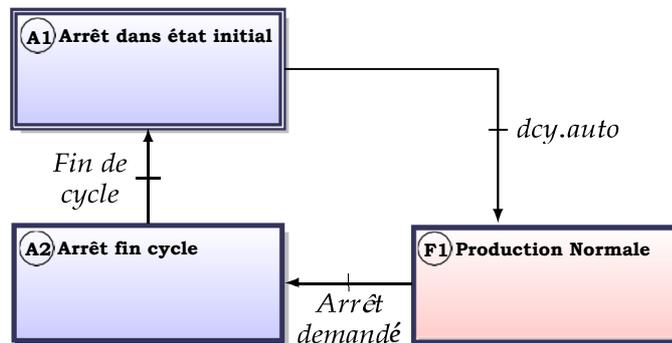


Figure IV-7. Condition dcy de mise en marche de production à cycle répété

IV.11.5. Marche de Vérification Manuelle dans l'Ordre

Cette marche pas à pas a pour but de vérifier la conformité du déroulement du cycle, le déroulement s'effectue sous le contrôle permanent de l'opérateur. C'est la boucle qui permet d'atteindre le *mode manuel*, ce mode permet à l'opérateur de pouvoir tester les actionneurs, pré-actionneurs, capteurs, etc., du système, dans l'ordre (Fig. IV-8).

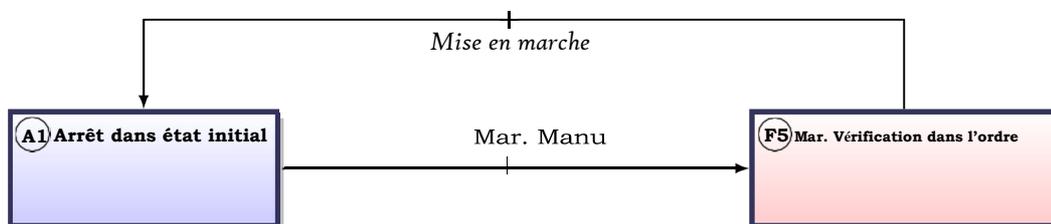


Figure IV-8. Marche de vérification manuelle dans l'ordre

IV.11.6. Marche de Vérification Manuelle dans le Désordre

C'est la boucle qui permet d'atteindre le *mode manuel*, ce mode permet à l'opérateur de pouvoir tester les actionneurs, pré-actionneurs, capteurs, etc., du système, dans le désordre. A la fin d'une marche de vérification dans le désordre, la machine n'est pas dans son état initial, il y a donc obligatoirement passage par le rectangle d'état A6 (Fig. IV-9).

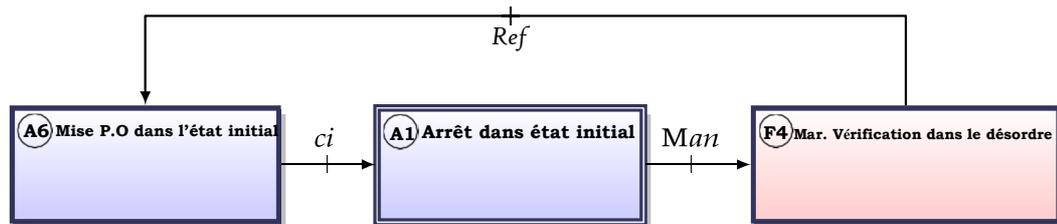


Figure IV-9. Marche de vérification manuelle dans le désordre

IV.ii.7. Marche Automatique avec Marche de Préparation et de Clôture

Ce GEMMA correspond au cas où la machine produit en *mode automatique* avec une marche de *préparation* et de *clôture*. Le rectangle d'état *F2* est utilisé lorsque l'automatisme exige certaines préparations avant de passer en production normale. Ces préparations peuvent être faites automatiquement ou manuellement. Pendant cette phase de préparation, il peut y avoir production ou non, ce qui explique que ce rectangle-état chevauche la frontière entre la zone *en production* et la zone *hors production*.

Par exemple, si l'automatisme est une unité de perçage, il faut actionner le moteur de perçage avant d'entrer en production.

Lorsque la machine exige de faire certaines opérations (en plus de la fin de cycle normal) pour retourner en condition initiale, il faut prévoir une marche dite de clôture. Cette marche peut être faite automatiquement ou manuellement. Selon les actions à faire, il y a production ou non, ce qui explique que ce rectangle-état chevauche la frontière entre la zone en production et la zone hors production.

Par exemple, on peut prévoir une marche de clôture pour indiquer à l'opérateur de nettoyer la machine. On peut aussi y prévoir la vidange d'un convoyeur et l'arrêt progressif des postes d'une machine (Fig. IV-10).

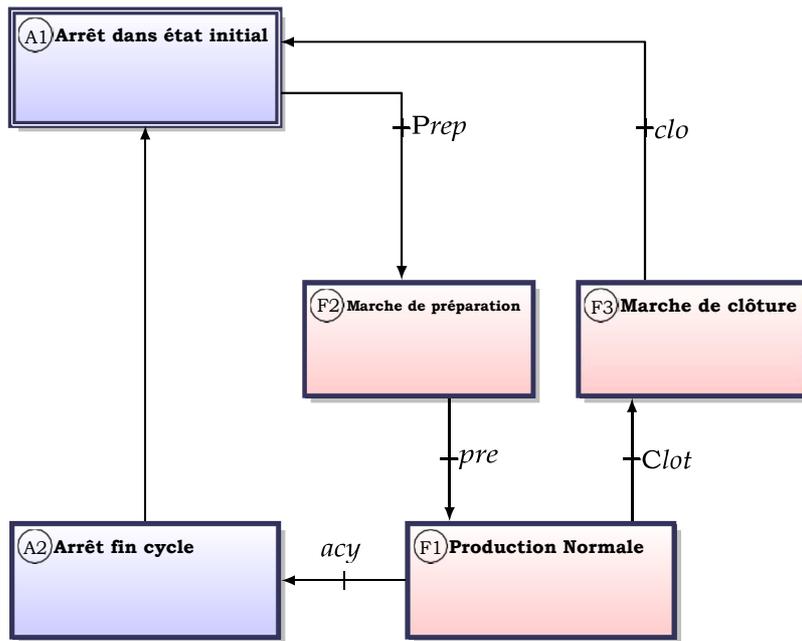


Figure IV-10. Marche de production automatique avec marche de préparation et de clôture

IV.11.8. Marche de Redémarrage à Partir de d'État Initiale

Le redémarrage (*initialisation*) peut être réalisé. C'est la boucle qui permet d'atteindre le *mode manuel*, ce mode permet à l'opérateur de pouvoir remanger le système. (Fig. IV-11).

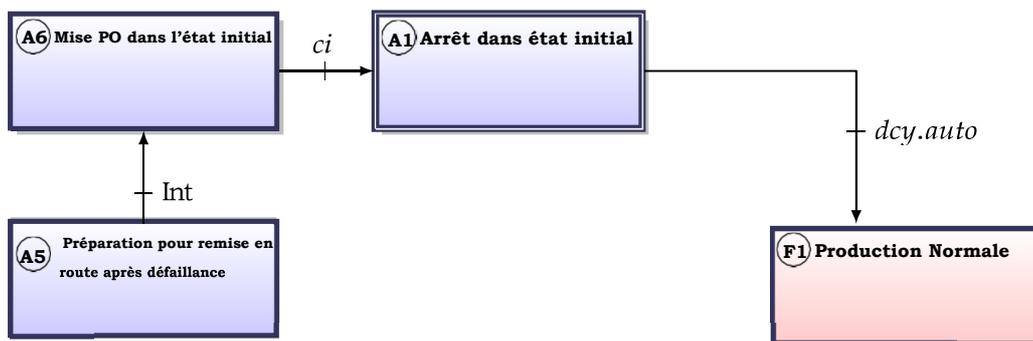


Figure IV-11. Marche de redémarrage à partir de d'état initiale

IV.11.9. Marche de Redémarrage à Partir de d'État Quelconque

En fonction de la cinématique de la machine et des conséquences physiques des choix faits précédemment, le redémarrage peut être réalisé (Fig. IV-12).

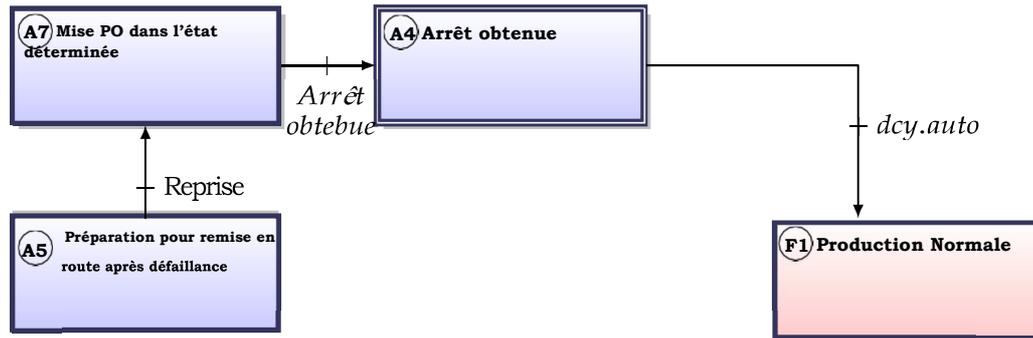


Figure IV-12. Marche de redémarrage à partir de d'état quelconque

IV.12. GRAFCETS ASSOCIÉS AUX GEMMA

IV.12.1. GRAFCET de Conduite G.C.

Il prend en compte tous les états et les conditions d'évolution retenus sur le GEMMA. C'est le GRAFCET qui gère tous les modes de marches sauf le GRAFCET de Traitement de l'Arrêt d'Urgence (G.T.A.U.). Son évolution est liée à une autorisation du GRAFCET de Sûreté (G.S.).

IV.12.2. GRAFCET de Sûreté G.S.

Il prend en compte toutes les procédures d'*arrêt d'urgence* énoncées dans le cahier des charges. C'est le *GRAFCET maître* de plus *haut niveau*. C'est ce GRAFCET qui assure le *forçage* des autres GRAFCET "*esclaves*" lors des procédures d'*arrêt d'urgence*.

IV.12.3. GRAFCET de Fonctionnement Normal G.F.N.

Le *GRAFCET de Fonctionnement Normal* (G.F.N.) ou bien le *GRAFCET de Production Normal* (G.P.N.) son évolution est liée à une autorisation du GRAFCET de Conduite (G.C.). Il correspond à l'état *F1* du GEMMA.

IV.12.4. GRAFCET de Traitement de l'Arrêt d'Urgence G.T.A.U.

Ce GRAFCET est géré directement par le *GRAFCET de sécurité*. Il contient toutes les actions de blocage des mouvements et de dégageement prévus par le cahier des charges et indiquées dans l'état *DI* du GEMMA.

IV.12.5. GRAFCET d'INITialisation G.I.N.I.T.

C'est lui qui assure la mise en *état initiale* de tous les organes de la partie opérative. Son évolution est liée à une autorisation du GRAFCET de Conduite (G.C.). Il correspond à l'état **A6** du GEMMA.

IV.12.6. GRAFCET de REGlage G.R.E.G.

C'est le GRAFCET qui permet à l'opérateur de *piloter* les différents organes du système séparément et dans l'ordre qui lui convient dans le cas d'un mode de marche manuelle programmé. Son évolution est liée à une autorisation du GRAFCET de Conduite (G.C.). Il correspond à l'état **F4** du GEMMA.

IV.12.7. GRAFCET de Coordination de Tâche G.C.T.

Ce sont les GRAFCETs qui correspondent aux différentes boîtes de l'analyse SADT (nœuds **A1**, **A2**, ...). Lorsqu'ils existent, ils sont gérés par le GRAFCET de Fonctionnement Normal (G.F.N.) que l'on nomme alors G.C.T. (GRAFCET de Coordination de Tâches).

Remarque : Au minimum le guide GEMMA utilise *trois GRAFCETs* qui travaillent ensemble : *GRAFCET de Conduite (G.C.)*, le *GRAFCET de Production Normal (G.P.N.)* et le *GRAFCET de Sûreté (G.S.)*.

IV.13. MISE EN ŒUVRE DU GEMMA

Il est impératif lors de l'opération de conditionnement des feuilles d'aluminium, pour les protéger des éraflures, d'intercaler une feuille de cuir entre chacune des feuilles d'aluminium (Fig. IV-13).

La consigne de marche (*dcy*) entraîne l'empilage des feuilles d'aluminium en intercalant des feuilles de cuir. Le cycle s'arrête dans le cas où l'un des stocks est à zéro. Le cycle d'empilage débute toujours par la prise d'une feuille de cuir.

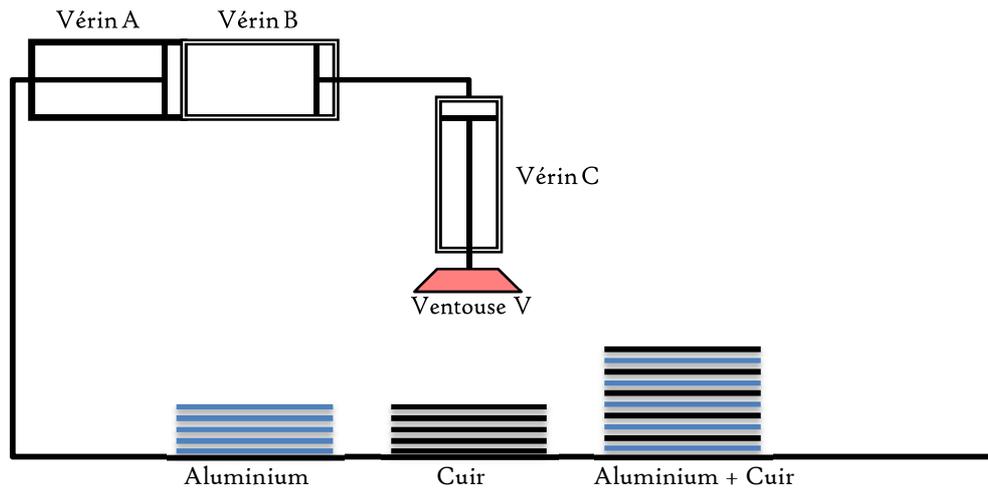


Figure IV-13. Système de conditionnement des feuilles d'aluminium

Les différents actionneurs du système ainsi que les capteurs utilisés sont indiqués dans les tableaux suivants :

Tableau IV-1. Actionneurs du système

DESIGNATIONS	COMMANDES
Vérin double effet A associé à un distributeur 5/2 bistable (24V)	A+ (Sortir le vérin) ; A- (Rentrer le vérin)
Vérin double effet B associé à un distributeur 5/2 bistable (24V)	B+ (Sortir le vérin) ; B- (Rentrer le vérin)
Vérin double effet C associé à un distributeur 4/2 monostable (24V)	C (Sortir le vérin)
Générateur de vide venturi V associé à un distributeur 3/2 monostable	V (Créer une aspiration)

Tableau IV-2. Capteurs du système

DESIGNATION	FONCTION	TYPE
a_0	Vérin A rentré	Détecteur magnétique
a_1	Vérin A sorti	Détecteur magnétique
b_0	Vérin B rentré	Détecteur magnétique
b_1	Vérin B sorti	Détecteur magnétique
c_0	Vérin C rentré	Détecteur magnétique
c_1	Vérin C sorti	
j	Présence de vide dans la ventouse	Vacuostat
cu	Présence d'une feuille de cuir	Détecteur de proximité
al	Présence d'une feuille d'aluminium	Détecteur de proximité

➤ Les modes de marche à considérer sont les suivants :

- ☞ marche automatique ;
- ☞ initialisation automatique de la partie opérative ;
- ☞ marche manuelle : des boutons de commande manuelle doivent permettre de démarrer le cycle aller/retour de chaque vérin et d'assurer le vide ;
- ☞ arrêt d'urgence.

➤ Travail demandé :

- 1- Compléter le guide GEMMA relatif au système.
- 2- Tracer les GRAFCETs de conduite GC, de production normale GPN, de marche manuelle GMM et de sûreté GS.

Les GRAFCETs de tâche (GT) et de production normal (GPN), de point de vue partie opérative, sont donnés respectivement par les Figures (IV-14) et (IV-15).

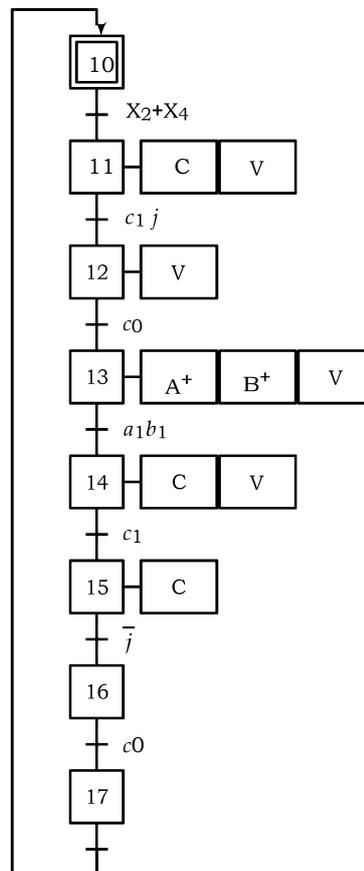


Figure IV-14. GRAFCET de tâche (GT)

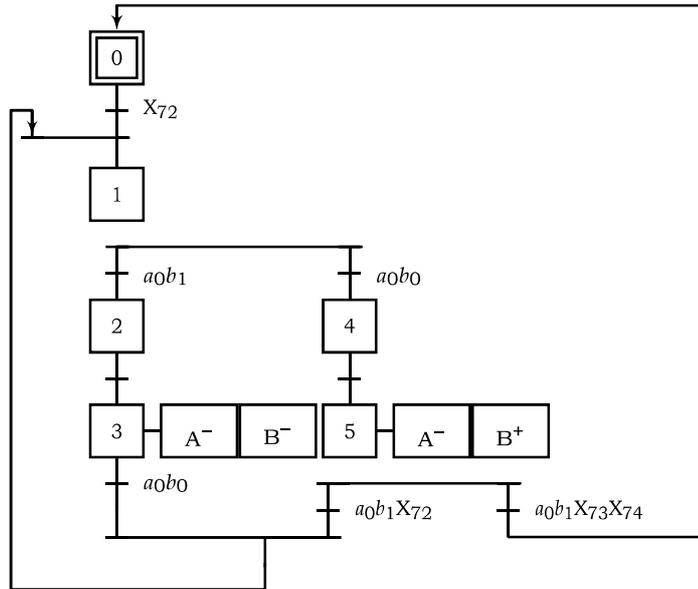


Figure IV-15. GRAFCET de production normal (GPN)

Les GRAFCETS de sécurité (GS) et de conduite (GC), sont donnés respectivement par les Figures (IV-16) et (IV-17). On peut décrire tout d'abord les modes de marche souhaité par le GEMMA de la Figure (IV-18).

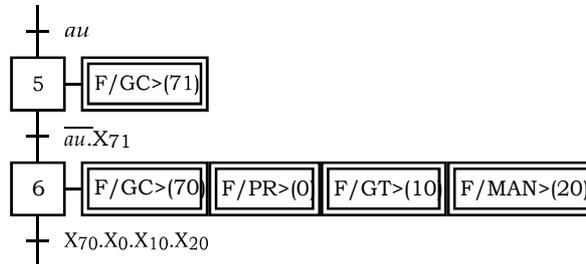


Figure IV-16. GRAFCET de sécurité (GS)

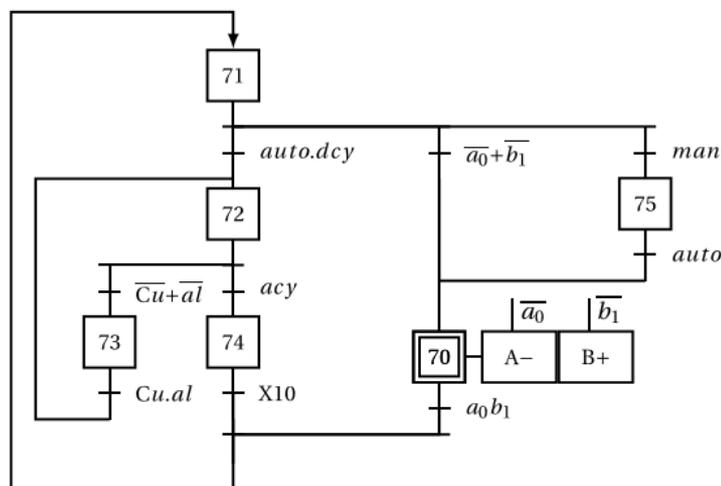


Figure IV-17. GRAFCET de conduite (GC)

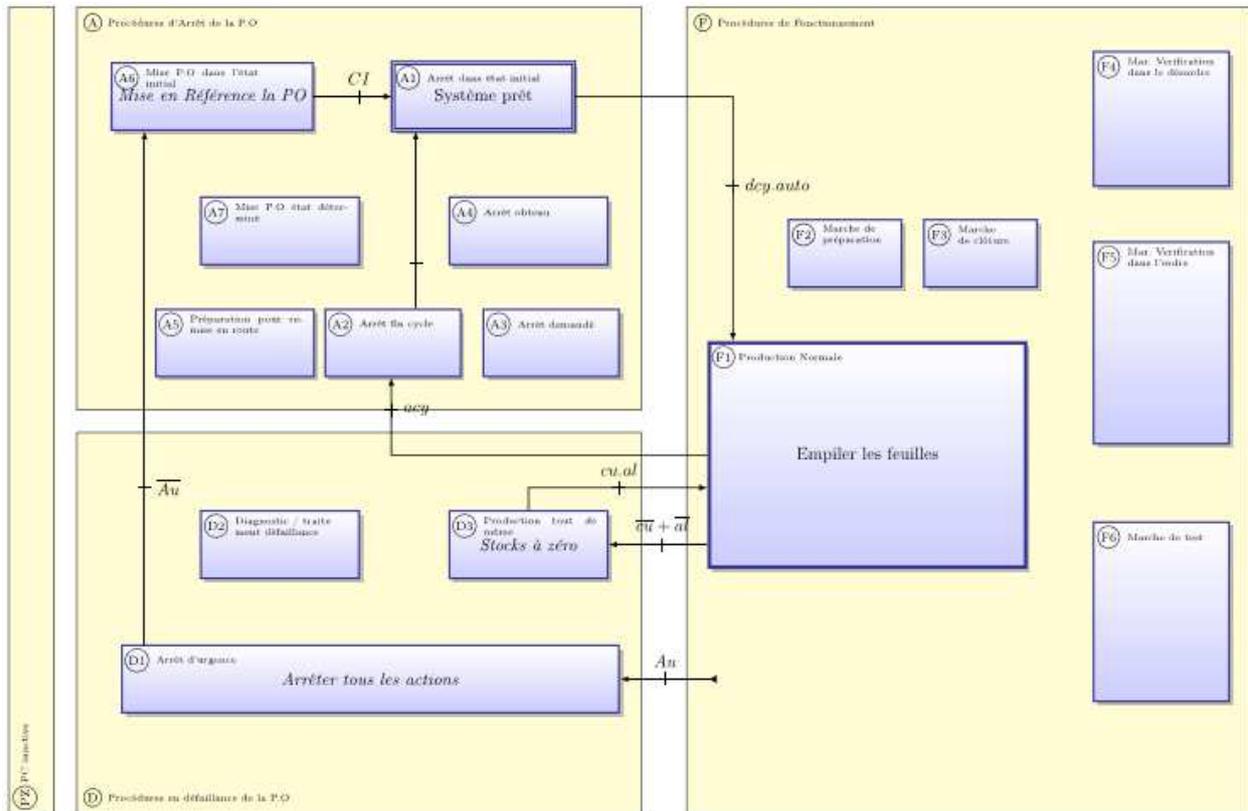


Figure IV-18. GEMMA du système de conditionnement des feuilles d'aluminium

IV.14. CONCLUSION

Il reste alors à en déduire le **GRAFCET complété** afin de terminer la définition des spécifications de la Partie Commande, y compris le pupitre et les capteurs supplémentaires éventuellement nécessaires : le GEMMA est un outil d'aide à la synthèse du cahier des charges. Enfin, ce document accompagne la vie du système : le GEMMA est un outil d'aide à la conduite de la machine, à sa maintenance ainsi qu'à son évolution.

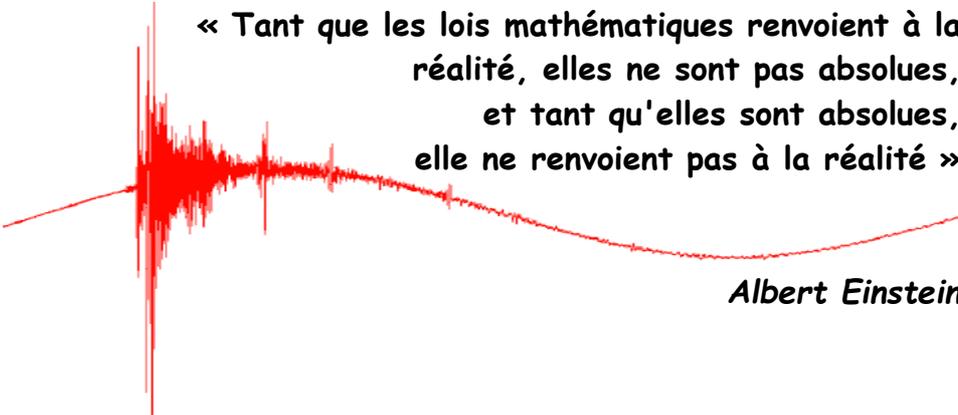
La description faite avec un GEMMA est très importante dans l'élaboration du fonctionnement d'un système automatisé. Elle va permettre de structurer le fonctionnement du système. Si généralement on souhaite que le système automatisé soit en production automatique, il est nécessaire de connaître précisément tous les autres comportements. Ce n'est pas en appuyant sur l'arrêt d'urgence que l'on découvrira le comportement du système dans cet état.

Le GEMMA est conçu pour une partie commande unique pilotant une partie opérative unique. Autrement dit, on considère le système globalement, ce qui implique qu'à tout instant on est dans un mode et un seul ou, ce qui revient au même, dans un "rectangle-état" et un seul. Les conséquences de cette unicité sont expliquées à la fin du chapitre.

Le chapitre suivant sera donc consacré aux architectures des automates programmables industriels des systèmes de production.

CHAPITRE V

ARCHITECTURE DES API



« Tant que les lois mathématiques renvoient à la réalité, elles ne sont pas absolues, et tant qu'elles sont absolues, elle ne renvoient pas à la réalité »

Albert Einstein

CHAPITRE V

ARCHITECTURE DES API

V.i. INTRODUCTION

L'objectif du développement de la recherche scientifique dans tous les domaines est d'aboutir à un progrès technologique voué principalement au bien être de l'humanité. Par sa nature de chercher à avoir la quantité, la qualité, le confort avec le minimum d'effort et du coût, l'homme a commencé à réfléchir, à concevoir et à réaliser des systèmes autonomes qui peuvent lui assurer tout dont il a besoin. Par conséquent, les outils et les appareils à énergies musculaires actionnés par des opérateurs sont remplacés par leurs équivalents à énergies électriques, mécaniques ou hydrauliques. La substitution d'opérateurs par ces systèmes définit l'automatisation.

Les *Automates Programmables Industriels (API)* ou *Programmable Logic Controller (PLC)* sont apparus aux Etats-Unis vers 1969 où ils répondaient aux désirs des industries de l'automobile de développer des chaînes de fabrication automatisées qui pourraient suivre l'évolution des techniques et des modèles fabriqués.

Un automate programmable industriel (API) est une forme particulière de contrôleur à microprocesseur qui utilise une mémoire programmable pour stocker les instructions et qui implémente différentes fonctions, qu'elles soient logiques, de séquençement, de temporisation, de comptage ou arithmétiques, pour commander les machines et les processus.

Ce chapitre constitue une introduction aux API, ainsi qu'à leur fonction générale, leurs formes matérielles et leur architecture interne. Les API sont employés dans de nombreuses tâches d'automatisation, dans différents domaines, comme les processus de fabrication industriels. Cette vue d'ensemble sera complétée par des explications plus détaillées.

L'objectif de ce chapitre est de présenter la structure interne et description des éléments d'un A.P.I, choix de l'unité de traitement, choix d'un automate programmable industriel, les interfaces d'entrées-sorties, la mise en œuvre d'un automate programmable industriel et les principes des réseaux d'automates.

V.2. CONTRAINTES DU MONDE INDUSTRIEL

Une contrainte technique est une donnée d'entrée dans la recherche d'une solution technique. Elle va compliquer le développement ou la mise en œuvre d'une solution. La contrainte peut être assimilée à un obstacle, à de la censure.

☞ *Influences externes :*

- poussières ;
- température ;
- humidité ;
- vibrations ;
- parasites électromagnétiques, ...

☞ *Personnel :*

- mise en œuvre du matériel aisée (pas de langage de programmation complexe) ;
- dépannage possible par des techniciens de formation électromécanique ;
- possibilité de modifier le système en cours de fonctionnement.

☞ *Matériel :*

- évolutif ;
- modulaire ;
- implantation aisée.

V.3. POURQUOI L'AUTOMATISATION ?

L'automatisation permet d'apporter des éléments supplémentaires à la valeur ajoutée par le système. Ces éléments sont exprimables en termes d'objectifs par [20] :

- Accroître la productivité (rentabilité, compétitivité) du système ;
- Améliorer la flexibilité de production ;
- Améliorer la qualité du produit ;
- Adaptation à des contextes particuliers tel que les environnements hostiles pour l'homme (milieu toxique, dangereux ... nucléaire ...), adaptation à des tâches physiques ou intellectuelles pénibles pour l'homme (manipulation de lourdes charges, tâches répétitives parallélisées ...) ;
- Augmenter la sécurité, etc...

V.4. DOMAINES D'EMPLOI DES AUTOMATES

On utilise les API dans tous les secteurs industriels pour la commande des machines (convoyage, emballage ...) ou des chaînes de production (automobile, agroalimentaire ...) ou il peut également assurer des fonctions de régulation de processus (métallurgie, chimie ...) [20].

Il est de plus en plus utilisé dans le domaine du bâtiment (tertiaire et industriel) pour le contrôle du chauffage, de l'éclairage, de la sécurité ou des alarmes.

V.5. AUTOMATES PROGRAMMABLES INDUSTRIELS

L'automatisation est définie comme étant l'ensemble des procédés qui rendent l'exécution d'une tâche, auparavant manuelle, automatique, sans intervention de l'homme.

L'automatisation des systèmes industriels n'est aujourd'hui pas seulement très répandue dans le domaine industriel, mais aussi indispensable à l'industrie, vu la complexité des systèmes de production de cette dernière, et les contraintes économiques dans un environnement très concurrentiel [24].

L'outil d'automatisation par excellence est l'automate programmable industriel appelé le plus souvent *API* ou *PLC* (*Programmable Logic controller*), qui offre des solutions simples à mettre en œuvre une souplesse d'adaptation à l'évolution des processus de production et une grande flexibilité.

V.5.1. Définition

Un *Automate Programmable Industriel* (API) (Fig. V-1) est une machine électronique programmable par un personnel non informaticien et destiné à piloter en ambiance industrielle et en temps réel des procédés industriels. Un automate programmable est adaptable à un maximum d'application, d'un point de vue traitement, composants, langage. C'est pour cela qu'il est de construction modulaire [28].



Figure V-1. Automate Programmable Industriel (API)

V.5.2. Rôle des APIs

L'API est un ensemble électronique qui gère et assure la commande d'un système automatisé. Il se compose de plusieurs parties, 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 [31].

V.5.3. Fonctions Principales des APIs

L'automate programmable Industriel effectue les fonctions principales suivantes (Fig. V-2) :

- ▶ La détection, depuis des capteurs de tous types répartis sur la machine ;
- ▶ La commande d'actions, vers les actionneurs et pré-actionneurs ;
- ▶ Le dialogue d'exploitation : des dialogues hommes-machine sont nécessaires pour la conduite de la machine, pour ses réglages et pour ses dépannages ;
- ▶ Le dialogue de supervision de production ; Les automates offrent un écran de visualisation où l'on peut voir l'évolution des entrées / sorties ;
- ▶ Le dialogue de programmation pour la première mise en œuvre.

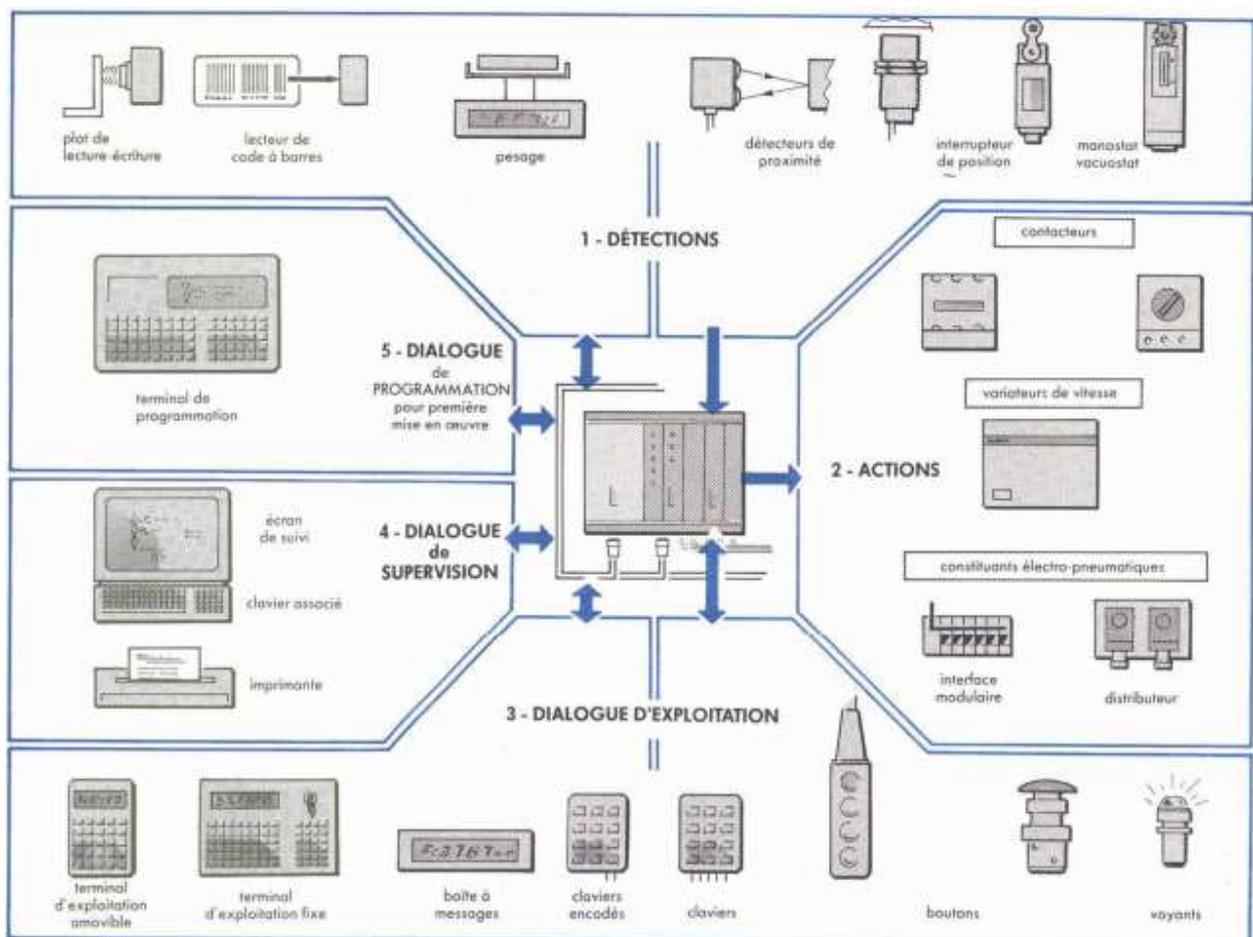


Figure V-2. Fonctions principales des APIs

V.5.4. Environnement des APIs

Les API se fonctionnent dans un environnement industriel qui se trouve en trois types :

- ❖ *Environnement physique et mécanique* : vibrations, chocs, humidité, température ;
- ❖ *Environnement chimique* : gaz corrosifs, vapeurs d'hydrocarbures, poussières métalliques (fonderies, aciéries, ...), poussières minérales (cimenteries, ...) ;
- ❖ *Environnement électrique* : les éléments perturbateurs sont : les parasites d'origine électrostatiques, les interférences électromagnétiques ... etc.

Ces environnements peuvent provoquer des dégâts non négligeables pour les appareils et des dangers pour les utilisateurs. Il faut donc veiller à la sécurité en analysant les risques et les normes en vigueur pour adapter les automates parfaitement à l'environnement industriel : Entrées/Sorties conformes aux standards de signaux industriels, protection contre les parasites électromagnétiques, le tenue aux chocs et aux vibrations, la résistance à la corrosion qui les contacts et provoque des courts-circuits, ... etc.

V.5.5. Avantage des APIs

Les automates programmables industriels, avec leur solution programmée, présentent de nombreux avantages par rapport à la technologie de logique câblée.

Parmi ces intérêts, on peut citer :

- ✓ La réduction de beaucoup d'espace requis pour l'installation ;
- ✓ Les éléments qui les composent sont particulièrement robustes leur permettant de fonctionner dans des environnements particulièrement hostiles. Améliorer les conditions de travail en éliminant les travaux répétitifs ;
- ✓ Améliorer la productivité en augmentant la production ;
- ✓ Améliorant la qualité des produits ou en réduisant les coûts de production ;

- ✓ Automates programmables sont programmés facilement et ont un langage de programmation facile à comprendre (logique programmé) alors la modification du programme facile par rapport à la logique câblée ;
- ✓ Simplification du câblage ;
- ✓ Puissance et rapidité ;
- ✓ Facilité de maintenance (l'API par lui-même est relativement fiable et peut aider l'homme dans sa recherche de défauts) ;
- ✓ Augmenter la sécurité ;
- ✓ Possibilités de communication avec l'extérieur (ordinateur, autre API) énorme possibilité d'exploitation ;
- ✓ Plus économique.

V.5.6. Inconvénient des APIs

En contrepartie, les APIs présentent les inconvénients suivants :

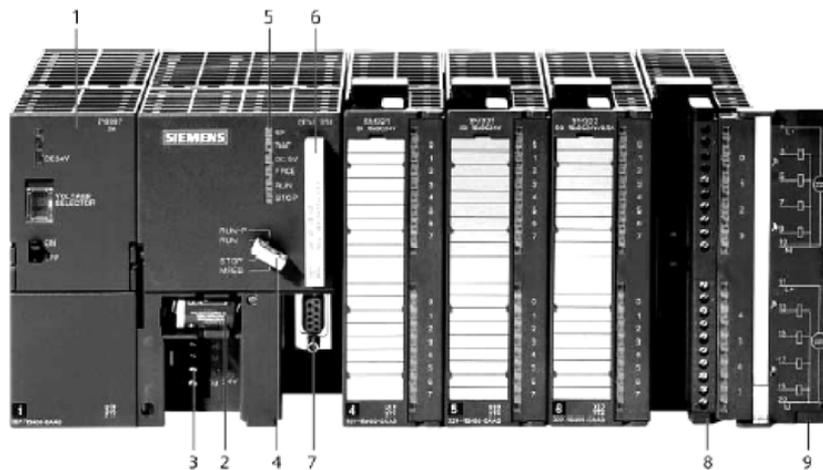
- ✘ Le prix est cher : le prix est notamment dépendant du nombre d'entrées/sorties nécessaires, de la mémoire dont on veut disposer pour réaliser le programme, de la présence ou non de modules métier ;
- ✘ La connaissance des langages de programmation ;
- ✘ Plantage ;
- ✘ Il y a trop de travail requis dans les fils de connexion ;
- ✘ Besoin de formation.

V.6. ARCHITECTURE DES AUTOMATES

Un automate programmable industriel est une machine électronique, programmable par un personnel non informaticien et destiné à commander au moyen de signaux d'entrées et de sorties, et d'un programme informatique, en temps réel, des procédés de processus industriels par un traitement séquentiel.

V.6.1. Aspect Extérieur des APIs

L'automate programmable (Fig. V-3) se présente comme un ensemble de blocs fonctionnels. Généralement, chaque bloc est physiquement réalisé par un module spécifique (coffret, rack, baie ou cartes). Ces différents modules s'articulant autour d'un canal de communication : le bus interne. L'automate programmable est du type modulaire contenant un rack, un module d'alimentation, un processeur, des modules d'E/S, des modules de communication et de comptage. Cette organisation modulaire permet une grande souplesse de configuration pour les besoins de l'utilisateur, ainsi qu'un diagnostic et une maintenance facilités et elle destinée pour les automatismes complexes où puissance, capacité de traitement et flexibilité sont nécessaires.



- | | | | |
|---|-------------------------------------------|---|----------------------------|
| 1 | Module d'alimentation | 6 | Cartes mémoires |
| 2 | Pile de sauvegarde | 7 | Interface multipoint (MPI) |
| 3 | Connexion au 24Vcc | 8 | Connecteur frontal |
| 4 | Commutateur de mode (à clé) | 9 | Volet en face avant |
| 5 | LED de signalisation d'état et de défauts | | |

Figure V-3. Automate modulaire (SIEMENS) [11,20]

Dans ces systèmes industriels (Fig. V-4), les APIs occupent une place de choix. Les équipements notés *commande* sont souvent des automates. Remplaçant initialement des ensembles en technologie câblée (relais électromagnétiques ou statiques, composants pneumatiques), ils constituent de plus en plus un maillon fiable et efficace entre le calculateur, qui a plutôt un rôle de gestion et

l'appareillage de terrain (capteurs et actionneurs). Cet appareillage pouvant lui-même aujourd'hui contenir un processeur, il nous faut préciser la définition de l'API, tout en sachant que dans ce domaine comme dans beaucoup d'autres touchant à l'informatique, les frontières sont floues et mouvantes.

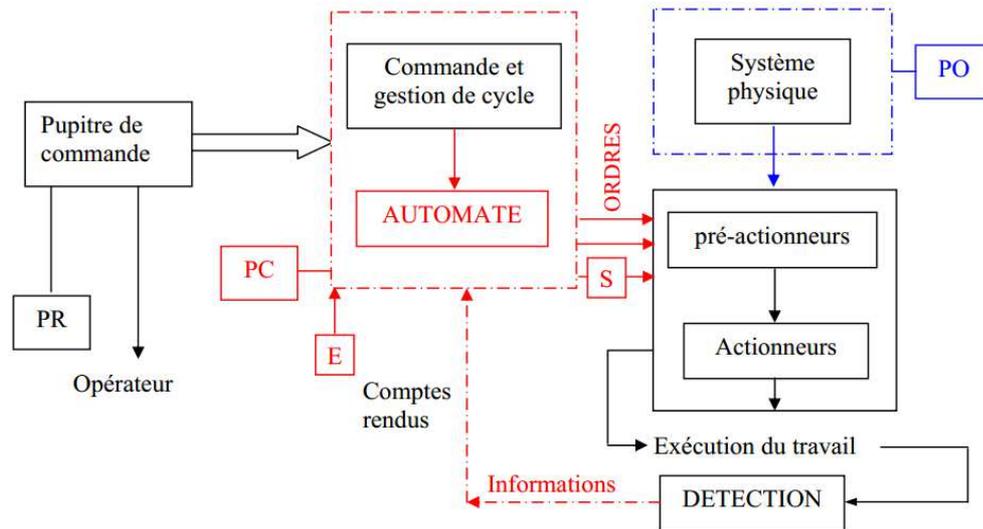


Figure V-4. Synoptique d'un système industriel automatisé [20]

Les automates peuvent être de type *compact* ou *modulaire*.

V.6.1.1. API de type compact

L'API de type *compact* (Fig. V-5) intègre le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage rapide, E/S analogiques ...) et recevoir des extensions en nombre limité. Ces automates, de fonctionnement simple, sont généralement destinés à la commande de petits automatismes [20].



Figure V-5. API de type compact

V.6.1.2. API de type modulaire

De type *modulaire* (Fig. V-6), le processeur, l'alimentation et les interfaces d'entrées / sorties résident dans des unités séparées (modules) et sont fixées sur un ou plusieurs racks contenant le "fond de panier" (bus plus connecteurs). Ces automates sont intégrés dans les automatismes complexes où puissance, capacité de traitement et flexibilité sont nécessaires.



Figure V-6. API de type modulaire

V.6.2. Aspect Intérieur des APIs

Pour une raison de simplicité et pour mieux comprendre les APIs, on donne sur la Figure (V-7) le schéma synoptique d'une structure simplifiée d'un API regroupant les principaux modules qui le constituent.

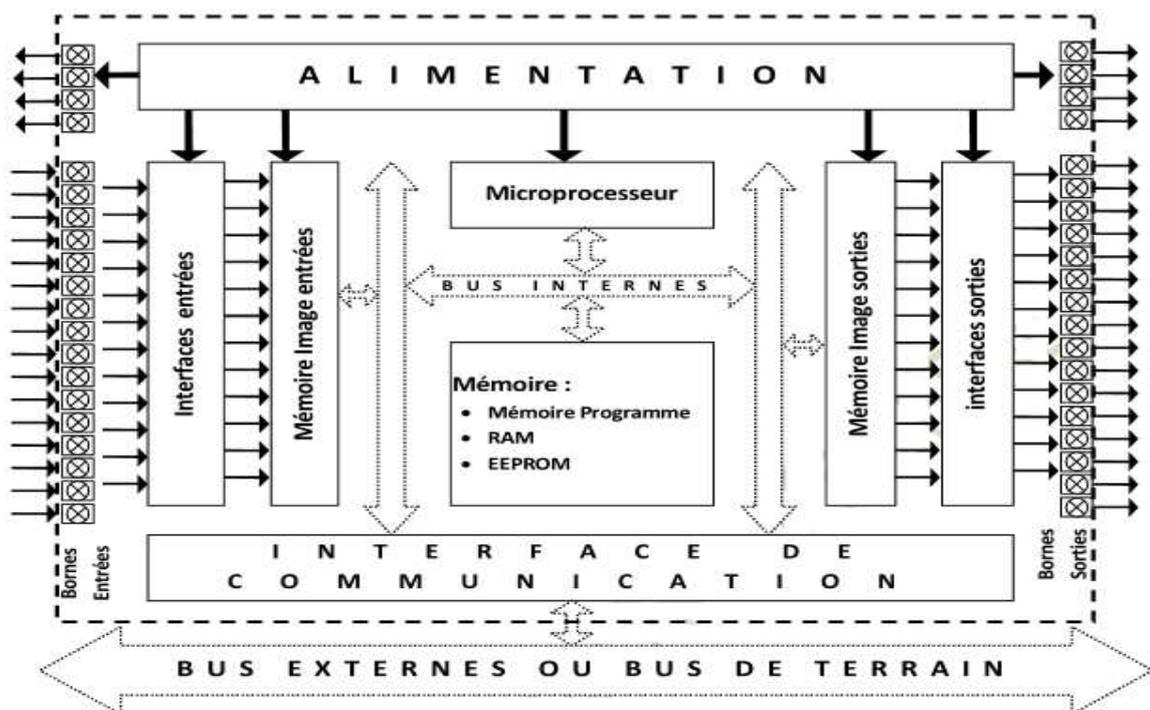


Figure V-7. Aspect intérieur d'un API

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

V.6.2.1. Processeur ou unité centrale de traitement (CPU, Central Processing Unit)

L'*unité de traitement* ou *unité centrale* constitue le noyau central de l'automate programmable. Le *microprocesseur* définit l'élément principal d'une unité centrale. C'est à ce niveau que s'effectuent les traitements et les prises de décisions suivant l'interprétation et l'exécution d'un programme écrit par un utilisateur [33].

Le *CPU* interprète les signaux d'entrée et effectue les actions de commande conformément au programme stocké en mémoire, en communiquant aux sorties les décisions sous forme de signaux d'action [23].

Dans le cas général, l'exécution d'un programme se fait instruction par instruction d'une façon séquentielle ; de la première instruction jusqu'à la dernière. Si on veut qu'un programme soit exécuté d'une façon cyclique continue, la dernière instruction devra être du type saut vers le début du programme principal [34].

Le temps réservé à l'exécution de chacune des instructions est fixé par un nombre de cycles de l'horloge du système. Les opérations effectuées par cette unité peuvent être réparties sur deux catégories :

- ❶ **Logiques** : AND, OR, XOR, NOT, SHIFT et ROTATE (ET, OU, OU EXCLUSIF, NON, DECALAGE et ROTATION) ;
- ❷ **Arithmétiques** : l'Addition, la Soustraction, la Multiplication, la Division.

V.6.2.2. Mémoire

La partie mémoire est destinée à la sauvegarde de programmes ou de données. Dans le cas des automates programmables la mémoire des données ou RAM est généralement divisée en espaces mémoires suivant la nature de leur fonction. On trouve la mémoire spéciale qui est destinée à la sauvegarde des états du système

lors de l'exécution d'un programme, elle affiche aussi les drapeaux indiquant l'état de la solution d'une opération tels que résultat nul, résultat négatif, etc.

Une mémoire répartie en registres joue le rôle d'image des entrées et des sorties numériques ; elle permet de verrouiller l'état des entrées ou des sorties afin de permettre au système d'y accéder aux entrées et d'envoyer les sorties avec un temps très réduit. Une troisième partie de la mémoire est réservée aux variables, on l'appelle dans la plus part des cas mémoire de l'utilisateur ou mémoire des données.

La mémoire d'un API peut être répartie sur différents types suivant la nature de son utilisation [20,35,36] :

- ① **ROM** ou **PROM** : elle est réservée au système d'exploitation du système. C'est ce qui est exécuté en premier dès la mise sous tension de l'API ;
- ② **EEPROM (FLASH)** : c'est la mémoire programme. Le programme écrit par un utilisateur est téléchargé dans cette mémoire pour être exécuté lors de l'activation du bouton RUN ;
- ③ **RAM** : c'est une mémoire volatile ; elle ne conserve pas son contenu après coupure de son alimentation.

Contrairement aux systèmes à microprocesseur à usage général, des parties de la mémoire sont réservées à des fonctions bien déterminées [37,38] :

- ① **Mémentos** : c'est une mémoire dont les bits seront utilisés comme des drapeaux de sauvegarde des résultats intermédiaires. Ils peuvent être aussi comme des états de conditions ;
- ② **Mémoire spéciale** : chacun des bits de cette mémoire est destiné à représenter un état lors de l'exécution d'un programme ;
- ③ **Mémoire image des entrées** : durant l'exécution d'un programme, l'API commence toujours par recopier les états des signaux issus des capteurs connectés aux différentes entrées dans les bits correspondant de la mémoire

images des entrées. Le programme de l'utilisateur manipule l'image d'une entrée au lieu de l'entrée elle-même ;

- ④ **Mémoire image des sorties** : Les sorties générées lors de l'exécution d'un programme sont sauvegardées dans les bits correspondants de la mémoire de sortie. A la fin de chaque cycle d'exécution du programme en cours les images sauvegardées dans la mémoire de sortie passent par l'interface de sortie à l'extérieur à travers les bornes correspondantes.

V.6.2.3. Interface d'entrée et de sortie

① Interface d'entrée et de sortie numérique

Les circuits d'interface d'entrée et de sortie ont deux fonctions principales ; la première concerne l'isolation de l'automate de son mode extérieur alors que la deuxième permet l'adaptation des niveaux des tensions de l'API avec celle qui vont ou viennent de l'extérieur.

Les schémas des Figures (V-8) et (V-9) montrent les circuits d'interface d'une entrée et d'une sortie respectivement où on constate une isolation optique et le passage du niveau de tension 5V à un niveau 24V et inversement. Sur le circuit de la Figure (V-9) on trouve aussi une isolation électromagnétique entre 24V et 220V alternatif.

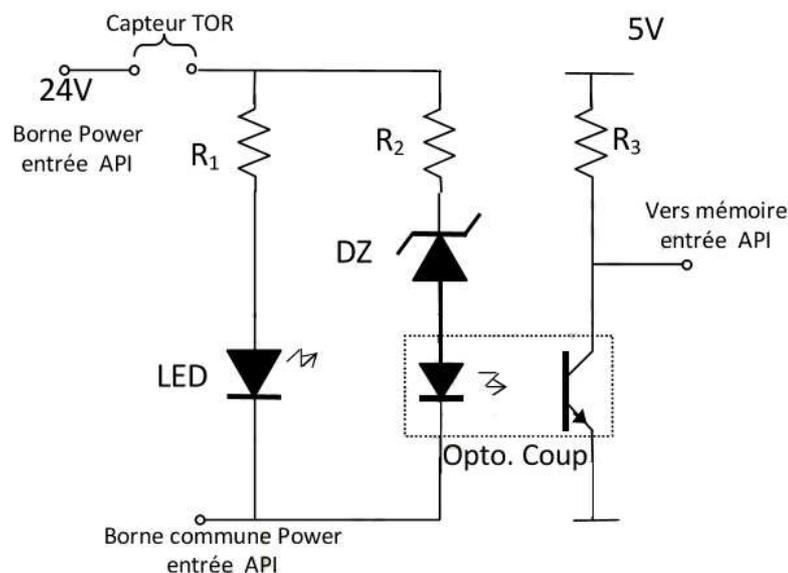


Figure V-8. Interface d'entrée [38]

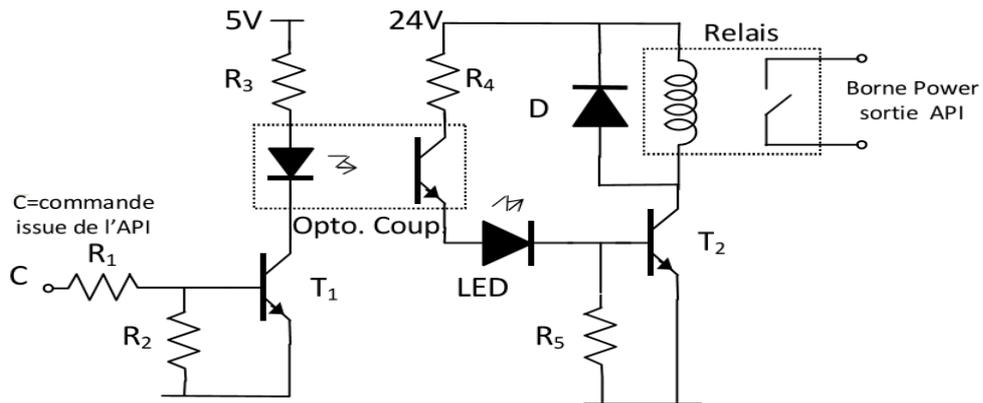


Figure V-9. Interface de sortie [38]

Les Figures (V-10) et (V-11) illustrent des exemples de connexion externes à un API.

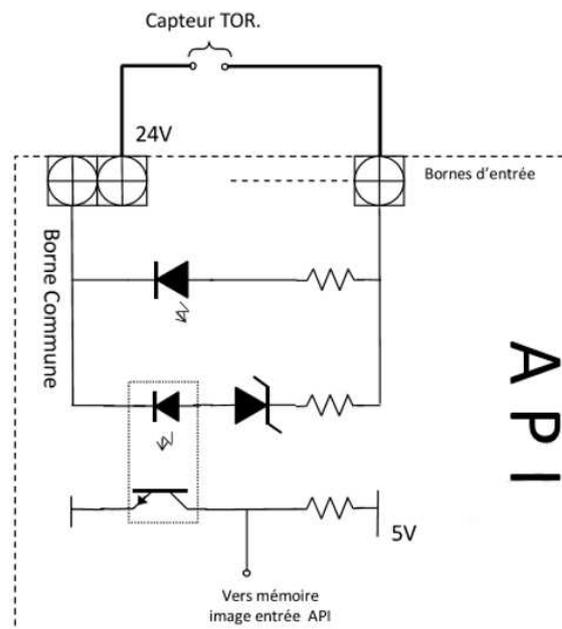


Figure V-10. Exemple de connexion d'un capteur TOR à une entrée de l'API [38]

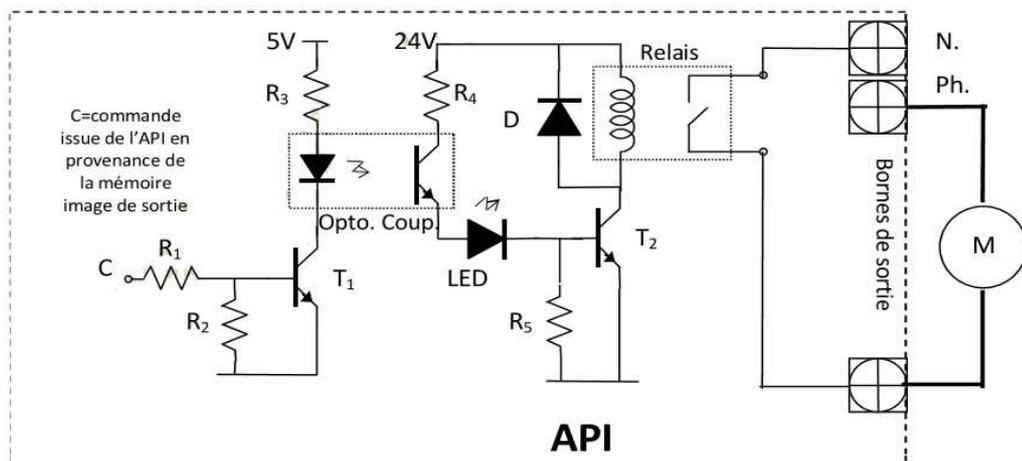


Figure V-11. Exemple de connexion d'un moteur à une sortie de l'API [38]

② *Interface d'entrée et de sortie analogique*

Actuellement, les systèmes informatiques occupent une place très importante dans le domaine du contrôle. L'opérateur humain est substitué par un système informatique, qui est capable de lire, de traiter et surtout de prendre une décision suivant l'état présent du processus contrôlé. Cependant, généralement les grandeurs contrôlées sont des grandeurs physiques traduites par des capteurs en grandeurs électriques analogiques, alors que les systèmes informatiques ne sont que des systèmes numériques pouvant manipuler des 1 et des 0.

Par conséquent, une fois la grandeur est conditionnée et prête à être traitée, elle doit être convertie en une suite de combinaisons numériques pour enfin être présentées au système numérique. Les circuits permettant une telle transformation sont appelés *Convertisseurs Analogiques Numériques CAN* ou plus exactement *ADC (Analog-Digital Converters)*.

Sur le marché nous trouvons une large gamme de CAN, des plus simples à ceux compatibles de travailler avec des microprocesseurs, et dont la liaison aux systèmes informatiques est directe.

V.6.2.4. Alimentation

Le module d'alimentation transforme l'énergie externe provenant du réseau en en la mettant en forme afin de fournir aux différents modules de l'API les niveaux de tension nécessaires à leur bon fonctionnement. Tous les automates actuels utilisent un bloc d'alimentation alimenté en 240Vac et délivrant une tension de 24Vcc, les tensions continues sont nécessaires au fonctionnement des circuits électroniques.

Le module d'alimentation assure la distribution d'énergie aux différents modules. Plusieurs niveaux de tension peuvent être utilisés par les circuits internes (3v, 5v, 12v, 24v ...) il sera dimensionné en fonction des consommations des différentes parties.

V.6.2.5. Modules complémentaires spéciaux

Les automates compacts permettent de commander des sorties en T.O.R et gèrent parfois des fonctions de comptage et de traitement analogique. Les automates modulaires permettent de réaliser de nombreuses autres fonctions grâce à des modules intelligents que l'on dispose sur un ou plusieurs racks. Ces modules ont l'avantage de ne pas surcharger le travail de la CPU car ils disposent bien souvent de leur propre processeur [39].

Nous rangerons dans cette catégorie des cartes qui assurent non seulement une liaison avec le monde extérieur mais aussi une partie du traitement pour soulager le processeur et donc améliorer les performances. De tels modules comportent donc un processeur spécifique ou une électronique spécialisée. Nous pouvons citer les suivants.

① Cartes de comptage rapide

Elles permettent de saisir des événements plus courts que la durée du cycle, travaillant à des fréquences qui peuvent dépasser 10 kHz. Elles permettent d'acquérir des informations de fréquences élevées incompatibles avec le temps de traitement de l'automate.

② Cartes de communication

Elles sont utilisées pour recevoir et transmettre des données sur des réseaux de communication qui relient l'API à d'autres API distants. Elles sont impliquées dans des opérations telles que la vérification d'un périphérique, l'acquisition de données, la synchronisation entre des applications et la gestion de la connexion.

③ Cartes d'acquisition

Dans l'API nous rangerons non seulement les acquisitions de mesures (entrées analogiques), mais aussi les sorties analogiques ; il existe des modules mixtes regroupant entrées et sorties. Dans les deux cas, il faut une conversion entre les valeurs analogiques et les mots manipulables par l'API. La valeur analogique

correspond à un nombre de bits (8 à 16) tel que l'erreur de quantification correspondante s'avère généralement inférieure à la résolution des capteurs ou à l'effet de seuil sur la commande des actionneurs. Les grandeurs analogiques reçues ou fournies obéissent à des standards électriques dont les plus fréquemment rencontrés sont le 4-20 mA, transmission en courant qui évite l'affaiblissement du signal [40].

④ *Bus interne*

Il permet la communication de l'ensemble des blocs de l'automate et des éventuelles extensions.

⑤ *Cartes de commande d'axe*

Elles permettent d'assurer le positionnement avec précision d'élément mécanique selon un ou plusieurs axes. La carte permet par exemple de piloter un servomoteur et de recevoir les informations de positionnement par un codeur. L'asservissement de position pouvant être réalisé en boucle fermée [31].

⑥ *Autres cartes*

- ☞ Cartes de régulation PID ;
- ☞ Cartes de pesage ;
- ☞ Cartes de surveillance et de contrôle.

V.7. FONCTIONNEMENT D'UN AUTOMATE

Les fonctions et les relations internes au système concernent essentiellement la gestion des informations, depuis leur acquisition ou saisie jusqu'à leur exploitation au niveau de la partie opérative (Fig. V-12) [40].

Cette gestion comporte les trois fonctions principales suivantes :

- ◆ Acquisition des informations ;
- ◆ Traitement des informations ;
- ◆ Exploitation des informations.

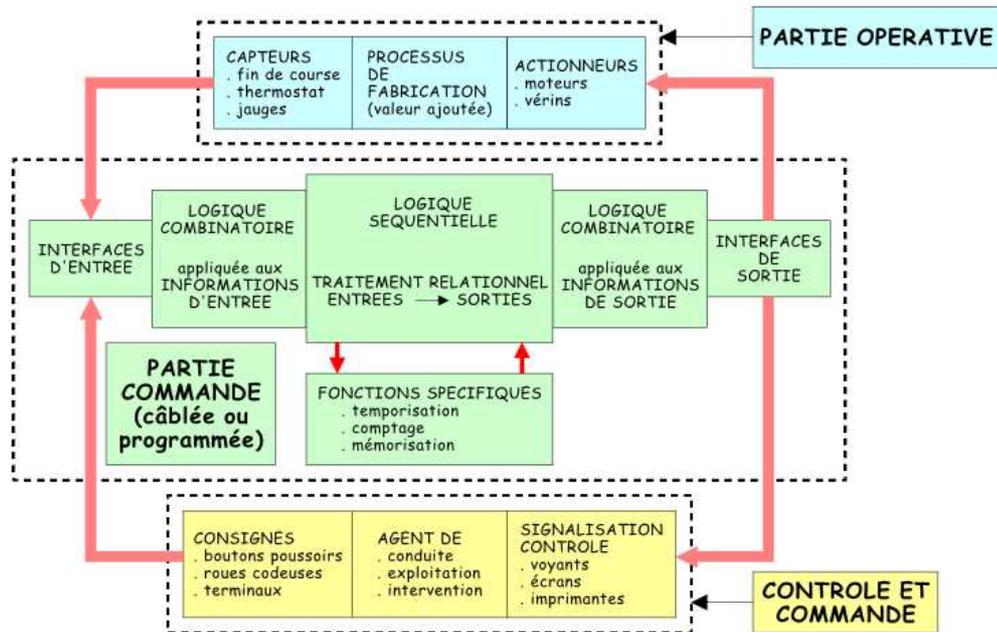


Figure V-12. Fonctions et relations internes au système automatisé [40]

V.7.1. Fonctions d'Acquisition des Informations

Cette fonction a pour objectif de fournir à la partie commande les informations relatives à l'état du système, c'est à dire au comportement à tout instant de la partie opérative et de contrôler les effets des ordres qu'elle lui adresse.

Les objets techniques qui permettent la saisie des informations sont les capteurs. Toutefois les signaux délivrés n'étant pas toujours compatibles avec les caractéristiques de l'unité centrale de traitement, il est nécessaire de les adapter. Cette opération fait appel à une fonction qui s'appelle fonction d'interfaçage d'entrée [41].

V.7.2. Fonctions de Traitement des Informations

Le traitement des informations est assurée par la partie commande (Fig. V-13) ; l'unité de traitement qui peut être en logique câblée ou en logique programmée, génère les signaux de commande ou ordres en direction de la partie opérative.

Là encore, les signaux délivrés par l'unité de traitement n'étant pas toujours compatibles avec les caractéristiques des pré-actionneurs il est nécessaire de les adapter au moyen de la fonction interfaçage. L'interfaçage qui permet

d'adaptation des signaux électriques générés par l'unité centrale pour leur exploitation dans le système commandé est un interfaçage de sortie.

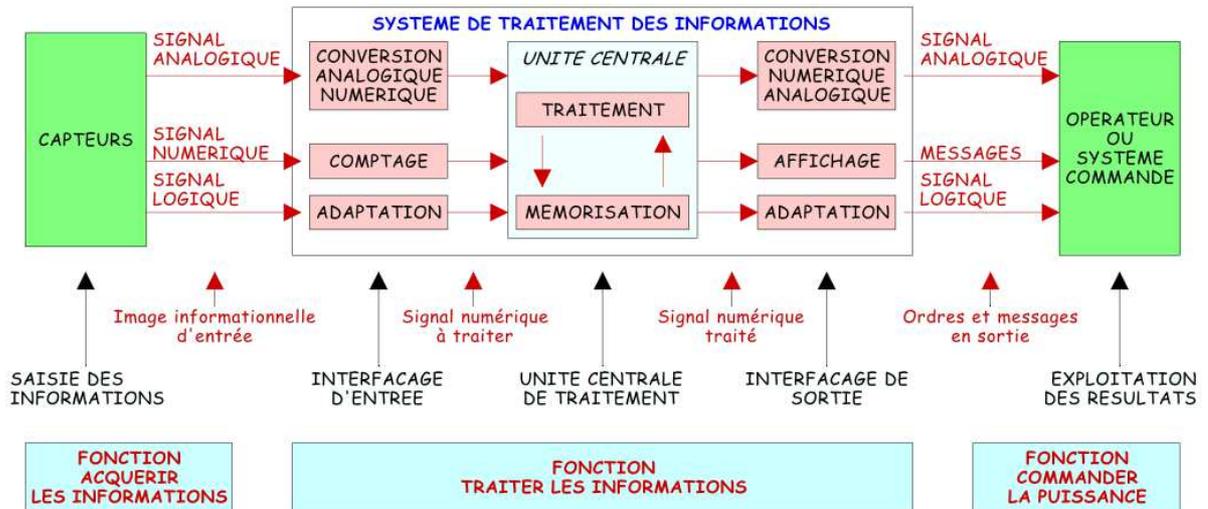


Figure V-13. Fonctions traitement d'un système automatisé [40]

Tous les automates fonctionnent selon le même mode opératoire qui assure en permanence un cycle composé de quatre tâches de la Figure (V-14) :

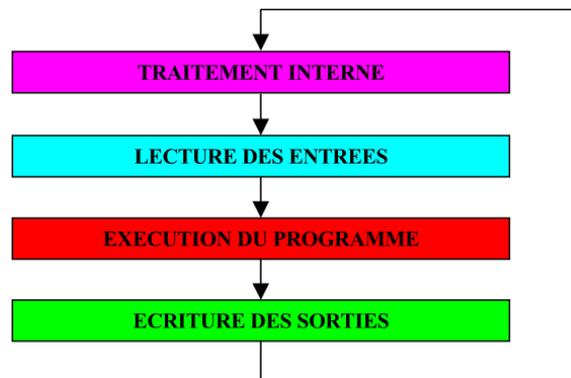


Figure V-14. Cycle de fonctionnement de l'API

V.7.2.1. Traitement interne

L'automate effectue des opérations de contrôle et met à jour certains paramètres systèmes (détection des passages en RUN / STOP, mises à jour des valeurs de l'horodateur, ...).

V.7.2.2. Lecture des entrées

L'automate lit les entrées (de façon synchrone) et les recopie dans la mémoire image des entrées.

V.7.2.3. Exécution du programme

L'automate exécute le programme instruction par instruction et écrit les sorties dans la mémoire image des sorties.

V.7.2.4. Ecriture des sorties

L'automate bascule les différentes sorties (de façon synchrone) aux positions définies dans la mémoire image des sorties.

Ces quatre opérations sont effectuées continuellement par l'automate (*fonctionnement cyclique*).

On appelle *scrutation* l'ensemble des quatre opérations réalisées par l'automate et le *temps de scrutation* est le temps mis par l'automate pour traiter la même partie de programme. Ce temps est de l'ordre de la dizaine de millisecondes pour les applications standards.

Le *temps de réponse total* (TRT) est le temps qui s'écoule entre le changement d'état d'une entrée et le changement d'état de la sortie correspondante (Fig. V-15).

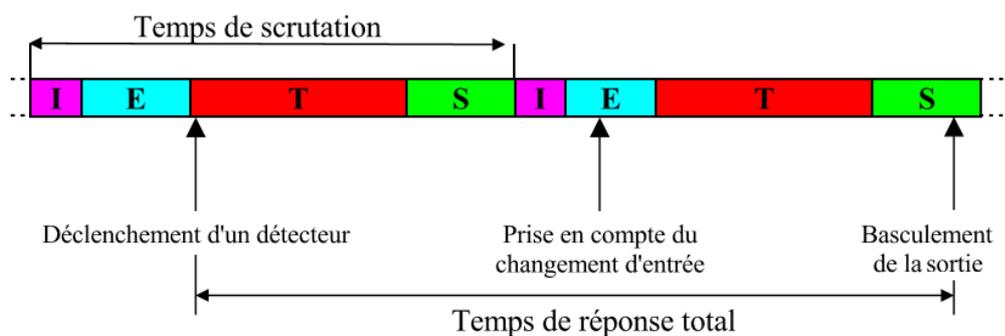


Figure V-15. Cycle de fonctionnement de l'API

Le temps de réponse total est au plus égal à deux fois le temps de scrutation (sans traitement particulier).

Le temps de scrutation est directement lié au programme implanté. Ce temps peut être fixé à une valeur précise (fonctionnement périodique), le système indiquera alors tout dépassement de période.

Dans certains cas, on ne peut admettre un temps de réponse aussi long pour certaines entrées : ces entrées pourront alors être traitées par l'automate comme des événements (traitement événementiel) et prises en compte en priorité (exemples : problème de sécurité, coupure d'alimentation ...).

Certains automates sont également pour vues d'entrées rapides qui sont prises en compte avant le traitement séquentiel mais le traitement évènementiel reste prioritaire.

V.7.3. Fonctions de Commande de la P.O.

Cette fonction qui permet à la partie opérative d'exécuter les ordres émis par l'automate programmable met en œuvre trois types d'objets techniques :

- ☞ l'effecteur ;
- ☞ l'actionneur ;
- ☞ le pré-actionneur.

V.7.4. Chaîne Fonctionnelle

On appelle chaîne fonctionnelle l'ensemble de constituants organisés en vue de l'obtention d'une tâche opérative (Fig. V-14).

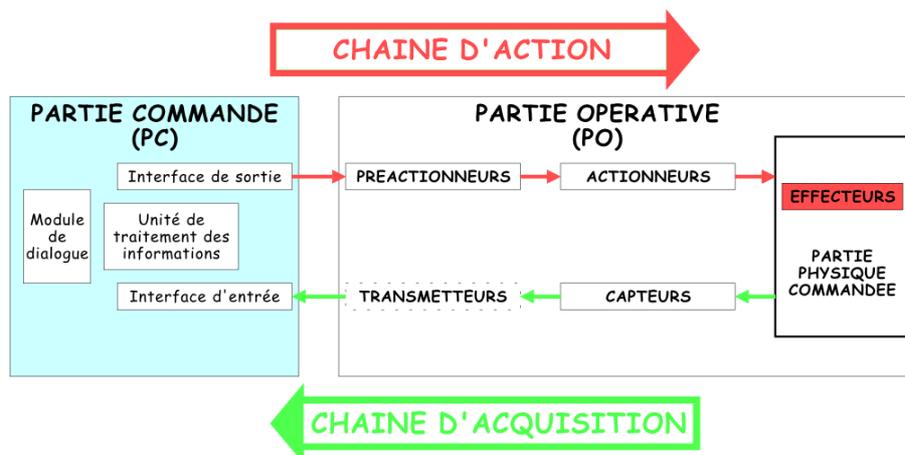


Figure V-14. Chaîne fonctionnelle

Décrivant une boucle, cette chaîne comporte généralement :

- ☞ Un constituant de traitement ;
- ☞ Une chaîne d'action ;

- ☞ La partie physique commandée ;
- ☞ Une chaîne d'acquisition qui rend compte de l'exécution de l'action.

V.7.5. Chaîne d'Action

Une chaîne d'action est constituée par l'ensemble des constituants qui permettent d'obtenir, à partir d'un ordre de la partie commande, un effet donné de la partie opérative d'un système automatisé sur la matière d'œuvre qu'il traite (Fig. V-15).

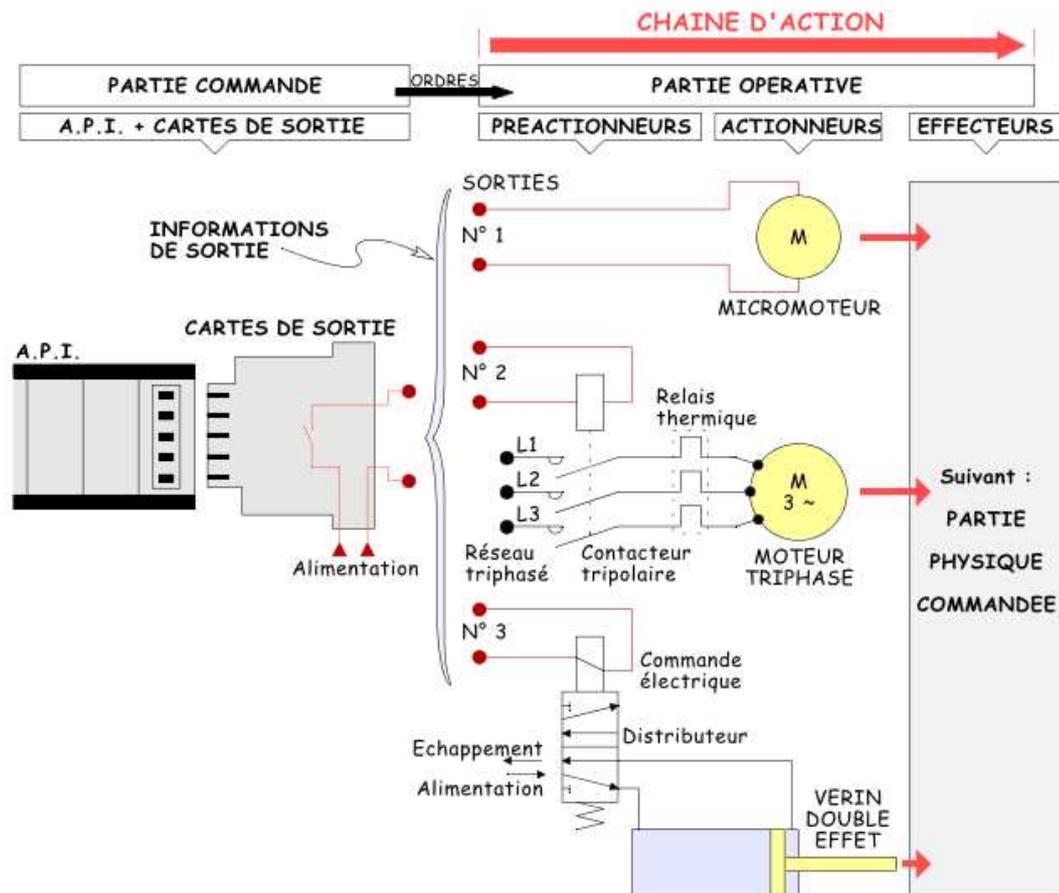


Figure V-15. Chaîne d'action

Les signaux émis par l'interface de sortie sont exploités, directement ou par l'intermédiaire de pré-actionneurs, pour la commande des actionneurs.

Lorsque la partie commande est assurée par un automate programmable, l'interface de sortie est matérialisée par des cartes ou module de sortie soit logiques (T.O.R.) soit analogiques.

V.7.6. Chaîne d'Acquisition

Une chaîne d'acquisition est constituée par l'ensemble des constituants qui permettent d'obtenir une image informationnelle exploitable par l'unité de traitement des informations en provenance de la partie opérative (Fig. V-16) [38].

La chaîne d'acquisition assure le transfert d'informations de la partie opérative vers la partie commande. Les constituants essentiels de cet échange informationnel sont :

- ☞ d'une part les capteurs ;
- ☞ d'autre part l'interface d'entrée.

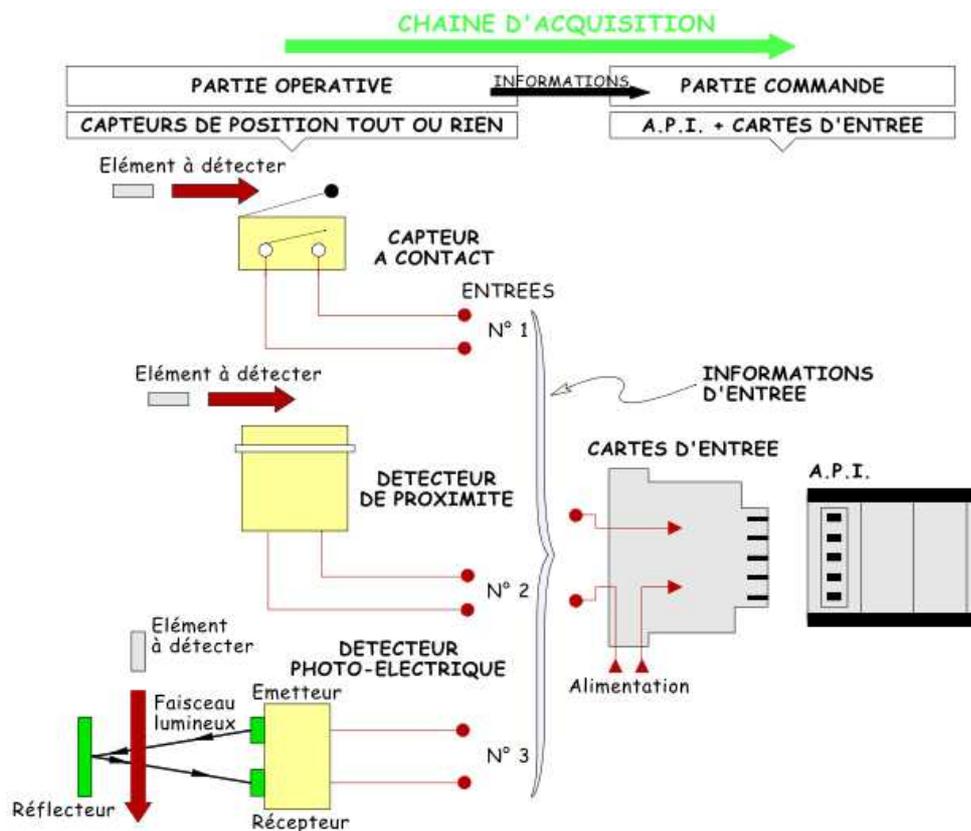


Figure V-16. Chaîne d'acquisition

V.8. COMMUNICATIONS DE L'AUTOMATE

La complexité des systèmes industriels automatisés rend indispensable :

- ⊙ le dialogue entre les opérateurs et la partie commande du système ;
- ⊙ la communication entre les parties commandes de divers systèmes automatisés intégrés dans une production à gestion centralisée.

Les échanges d'informations entre l'opérateur et la machine désignés par *dialogue homme-machine*, sont nécessaires au cours des différentes phases de vie du système. On distingue :

- Le *dialogue de programmation*, lors de la phase de développement et de mise au point du système ;
- Le *dialogue d'exploitation* au cours des phases de conduite, de réglage, de maintenance et de dépannage ;
- Le *dialogue de supervision* qui assure la coordination des systèmes automatisés concernés par une même production.

V.8.1. Dialogue de Programmation

Mis en œuvre essentiellement lorsque l'unité centrale de traitement est une machine informatique, automate programmable par exemple, le dialogue d'exploitation consiste à :

- Écrire et interpréter, sous une forme interactive l'ensemble des instructions d'un programme ;
- Mettre au point par simulation ou contrôle logiciel l'exécution d'un programme ;
- Sauvegarder le programme en mémoire, soit dans l'unité centrale de l'A.P.I., soit dans des mémoires de masses auxiliaires ; à partir d'un terminal, d'un PC ou d'une console de programmation.

V.8.2. Dialogue d'Exploitation

A partir d'un terminal d'exploitation l'opérateur peut, dans le langage utilisateur du système :

- ① Lire sur un écran un message relatif :
 - à l'état du système ;
 - à la nature du produit traité ;
 - à des mesures ;

- à des défauts de fonctionnement ;
 - à des ordres émis.
- ② *Commander par l'intermédiaire d'un clavier l'évolution du système :*
- sélection des modes de fonctionnement ;
 - saisie de consignes ;
 - émission d'ordres ;
 - modification du cycle dans les limites autorisées par le programme.
- ③ *Accéder et mettre en œuvre des procédures d'exploitation stockées dans une mémoire :*
- autotest du terminal ;
 - mise à jour de l'horodateur ;
 - impression de l'historique de recettes et de messages enregistrés.

V.8.3. Dialogue de Supervision

L'évolution des systèmes de production conduit à une automatisation de complexité croissante. Ces systèmes sont structurés en machines ou sous-machines comportant leur Partie Commande.

En plus des dialogues évoqués précédemment qui ont lieu au niveau de niveau de chaque machine ou sous-machine, il est nécessaire de coordonner les systèmes automatisés entre eux, d'en gérer le fonctionnement, la maintenance, etc ... : C'est le dialogue de supervision.

Moyens du dialogue de supervision : c'est le besoin des différents systèmes d'échanger des informations nécessite divers moyens de liaison. Ceux-ci sont assurés par transmissions série ou parallèle, ou mieux, par des réseaux appropriés.

V.9. PROTECTION DE L'AUTOMATE

Les systèmes automatisés sont, par nature, source de nombreux dangers (tensions utilisées, déplacements mécaniques, jets de matière sous pression, etc.).

Placé au cœur du système automatisé, l'automate se doit d'être un élément fiable car :

- ◆ un dysfonctionnement de celui-ci pourrait avoir de graves répercussions sur la sécurité des personnes ;
- ◆ les coûts de réparation de l'outil de production sont généralement très élevés ;
- ◆ un arrêt de la production peut avoir de lourdes conséquences sur le plan financier.

Aussi, l'automate fait l'objet de nombreuses dispositions pour assurer la sécurité :

- **Contraintes extérieures** : l'automate est conçu pour supporter les différentes contraintes du monde industriel et a fait l'objet de nombreux tests normalisés (tenue aux vibrations, CEM ...) ;
- **Coupsures d'alimentation** : l'automate est conçu pour supporter les coupures d'alimentation et permet, par programme, d'assurer un fonctionnement correct lors de la réalimentation (reprises à froid ou à chaud) ;
- **Mode RUN/STOP** : Seul un technicien peut mettre en marche ou arrêter un automate et la remise en marche se fait par une procédure d'initialisation (programmée) ;
- **Visualisation** : Les automates offrent un écran de visualisation où l'on peut voir l'évolution des entrées / sorties.

V.10. CRITÈRES DE CHOIX D'UN AUTOMATE

Le choix d'un automate programmable est en premier lieu le choix d'une société ou d'un groupe. Les contacts commerciaux et les expériences vécues sont déjà un point de départ. Les grandes sociétés privilégieront deux fabricants pour faire jouer la concurrence et pouvoir se retourner en cas de perte de vitesse de l'une d'entre elles. Le personnel de maintenance doit toutefois être formé sur ces matériels et une grande diversité des matériels peut avoir de graves répercussions.

Un automate utilisant des langages de programmation de type GRAFCET est également préférable pour assurer les mises au point et dépannages dans les meilleures conditions. La possession d'un logiciel de programmation est aussi source d'économies (achat du logiciel et formation du personnel), des outils permettant une simulation des programmes sont également souhaitables [31].

Il faut ensuite quantifier les besoins :

- **Nombre d'entrées / sorties** : le nombre de cartes peut avoir une incidence sur le nombre de racks dès que le nombre d'entrées / sorties nécessaires devient élevé ;
- **Type de processeur** : la taille mémoire, la vitesse de traitement et les fonctions spéciales offertes par le processeur permettront le choix dans la gamme souvent très étendue ;
- **Fonctions ou modules spéciaux** : certaines cartes (commande d'axe, pesage ...) permettront de soulager le processeur et devront offrir les caractéristiques souhaitées (résolution, ...) ;
- **Fonctions de communication** : l'automate doit pouvoir communiquer avec les autres systèmes de commande (API, supervision ...) et offrir des possibilités de communication avec des standards normalisés (Profibus ...) ;
- **Moyens de dialogue et le langage de programmation** ;
- **Moyens de sauvegarde** du programme ;
- **Fiabilité, robustesse, immunité** aux parasites ;
- **Documentation**, le *service* après-vente, *durée* de la garantie et la *formation*.

V.II. CABLAGE DES AUTOMATES

V.II.I. Présentation de l'Automate

L'alimentation intégrée dans l'API, fournit à partir des tensions usuelles des réseaux (230 V_{AC}, 24 V_{DC}) les tensions continues nécessaires au fonctionnement des circuits électroniques (Fig. V-17).

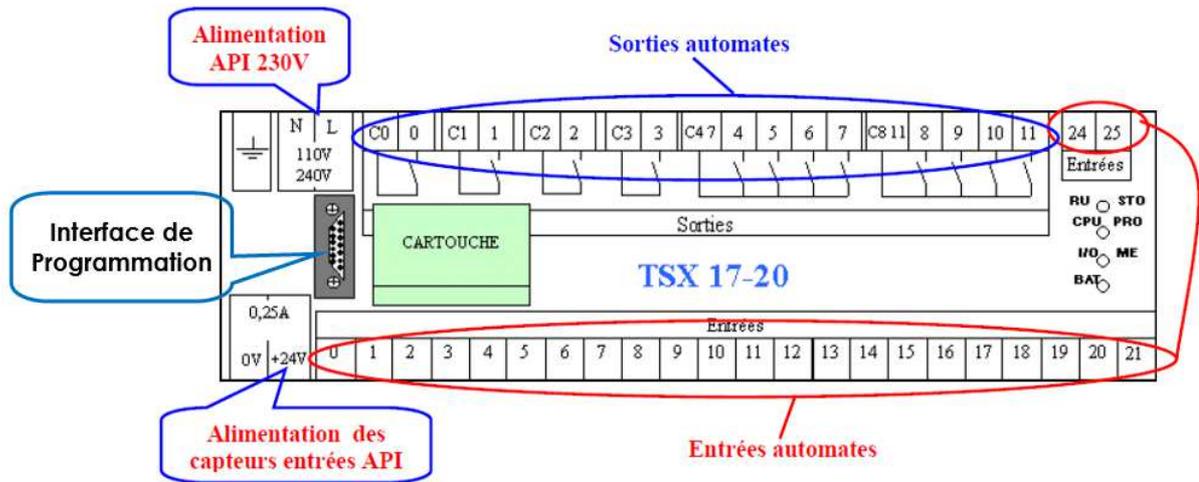


Figure V-17. Présentation de l'automate TSX 17-20

V.11.2. Alimentation de l'Automate

L'automate est alimenté généralement par le réseau monophasé 230V ; 50 Hz mais d'autres alimentations sont possibles (110V etc ...).

La protection est de type magnétothermique selon les caractéristiques de l'automate et les préconisations du constructeur. Il est souhaitable d'asservir l'alimentation de l'automate par un circuit de commande spécifique (contacteur KMI) (Fig.V-18). De même, les sorties seront asservies au circuit de commande et alimentées après validation du chien de garde.

V.11.3. Câblage des Entrées/Sorties

L'automate est pourvu généralement d'une alimentation pour les capteurs/détecteurs (logique positive ou négative). Les entrées sont connectées au 0V (commun) de cette alimentation. Les informations des capteurs/détecteurs sont traitées par les interfaces d'entrées (Fig.V-19).

V.11.4. Câblage de l'Automate/Actionneur

Le principe de raccordement consiste à envoyer un signal électrique vers le pré-actionneur connecté à la sortie choisie de l'automate dès que l'ordre est émis. L'alimentation électrique est fournie par une source extérieure à l'automate programmable (Fig.V-20).

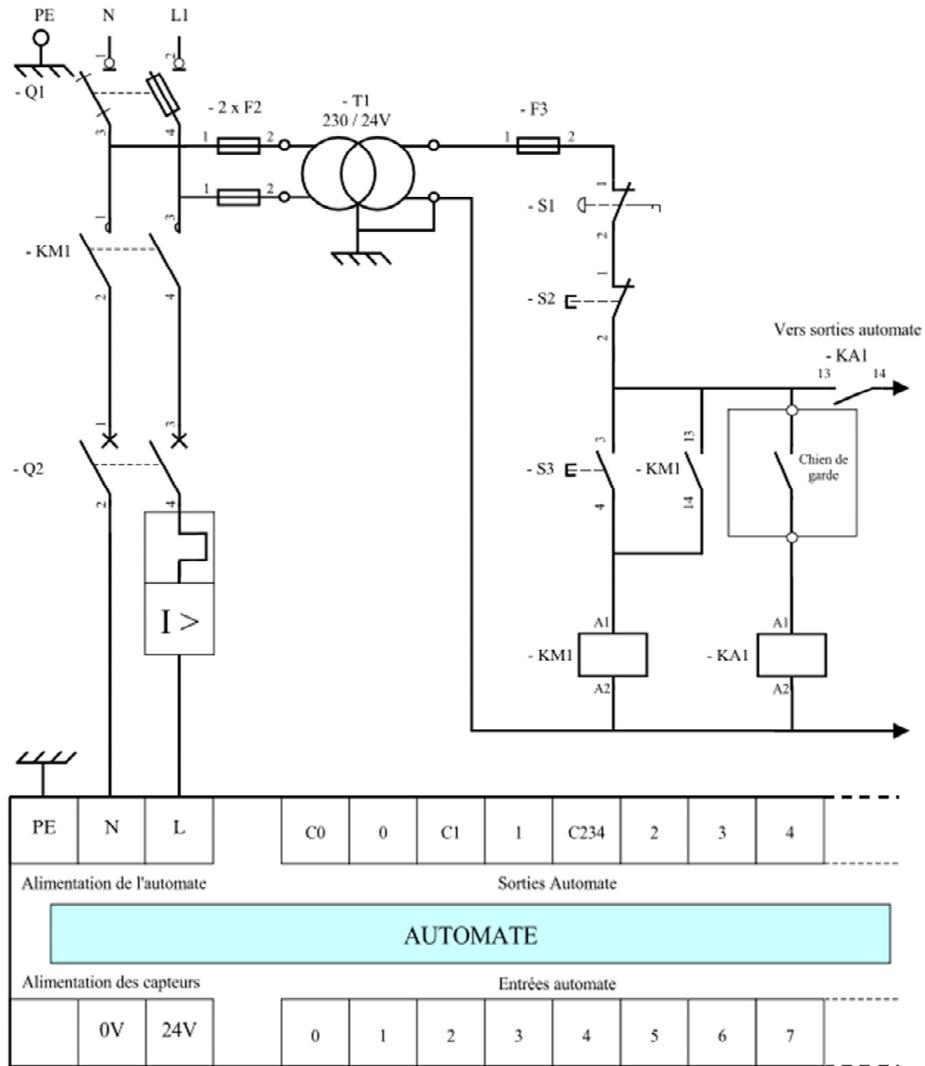


Figure V-18. Alimentation de l'automate

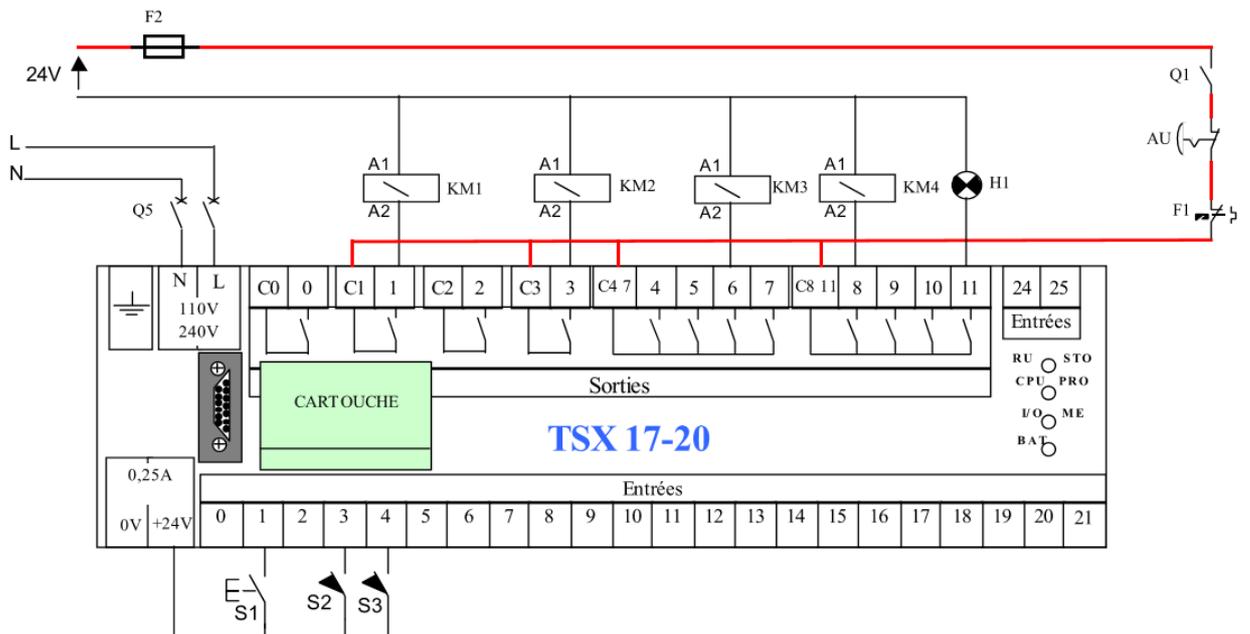


Figure V-19. Raccordement des entrées/sorties

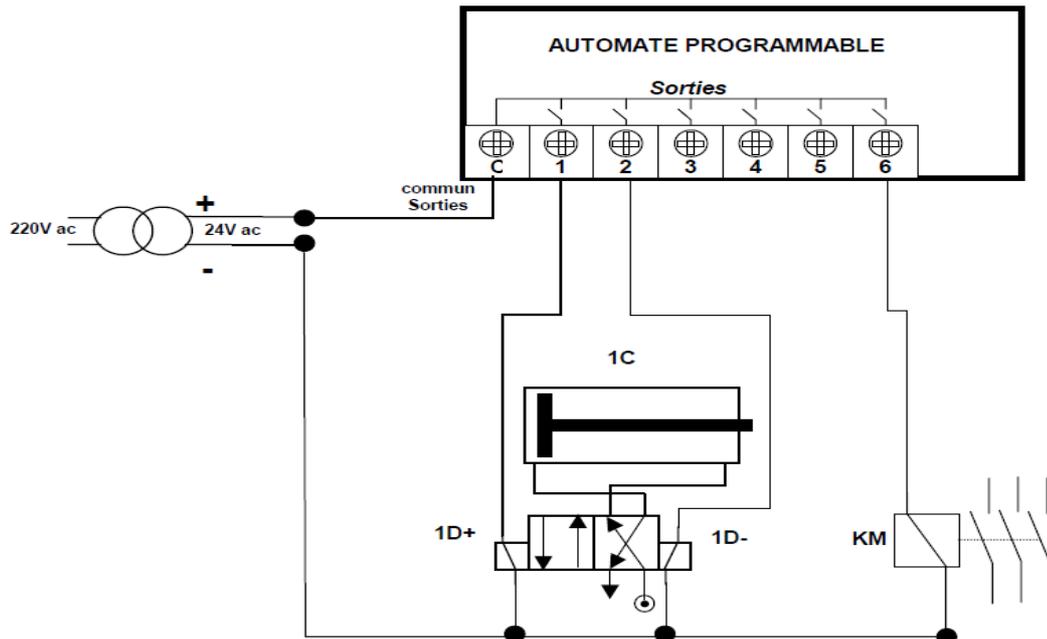


Figure V-20. Raccordement automate/vérin

V.12. RÉSEAUX DES AUTOMOMES

V.12.1. Principe

Avec le développement des systèmes automatisés et de l'électronique, la recherche de la baisse des coûts et la nécessité actuelle de pouvoir gérer au mieux la production et à partir du moment où tous les équipements sont de type informatique, il devient intéressant de les interconnecter à un mini-ordinateur ou à un automate de supervision (Fig.V-21).

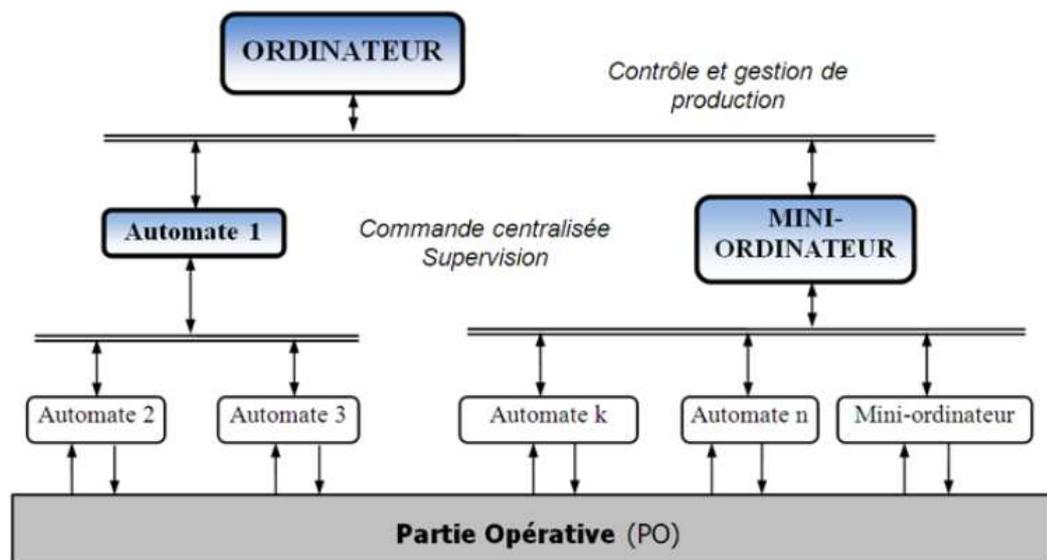


Figure V-21. Structure de contrôle et gestion de production

L'interconnexion entre deux automates peut être réalisée très simplement en reliant une ou plusieurs sorties d'un automate à des entrées de l'autre et vice-versa (Fig.V-22).

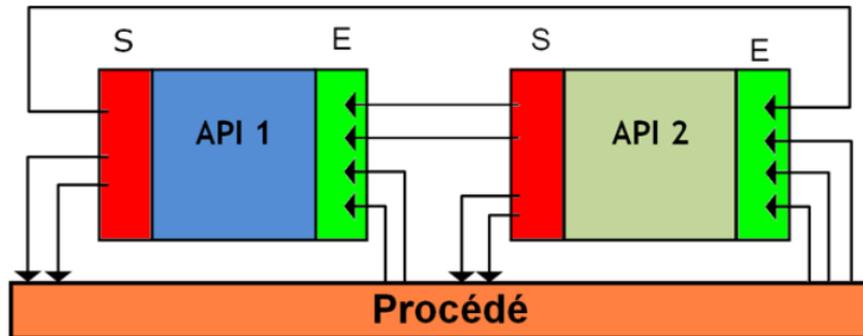


Figure V-22. Interconnexion simple (Entrées/Sorties) entre deux automates (API)

Cette méthode ne permet pas de transférer directement des variables internes d'un automate sur l'autre, de sorte que celles-ci doivent être converties par programme en variables de sortie avant leur transfert. Elle devient coûteuse en nombre d'entrées/sorties mobilisé pour cet usage et lourde du point de vue du câblage, lorsque le nombre de variables qui doivent être échangées devient important.

V.12.2. Bus de Terrain

Pour diminuer les coûts de câblage des entrées/sorties des automates, sont apparus les bus de terrains. L'utilisation de blocs d'entrées/sorties déportés a permis tout d'abord de répondre à cette exigence. Les interfaces d'entrées/sorties sont déportées au plus près des capteurs. Avec le développement technologique, les capteurs, détecteurs ... sont devenus intelligents et ont permis de se connecter directement à un bus (Fig.V-23).

Plusieurs protocoles de communication et des standards sont apparus pour assurer le *multiplexage* de toutes les informations en provenance des capteurs/pré-actionneurs par exemple le bus *ASi* (*Actuators Sensors interface*) est un bus de capteurs/actionneurs de type *Maître/Esclave* qui permet de raccorder 31 esclaves (capteurs ou pré-actionneurs) sur un câble spécifique (deux fils) transportant les

données et la puissance. Ce bus est totalement standardisé et permet d'utiliser des technologies de plusieurs constructeurs.

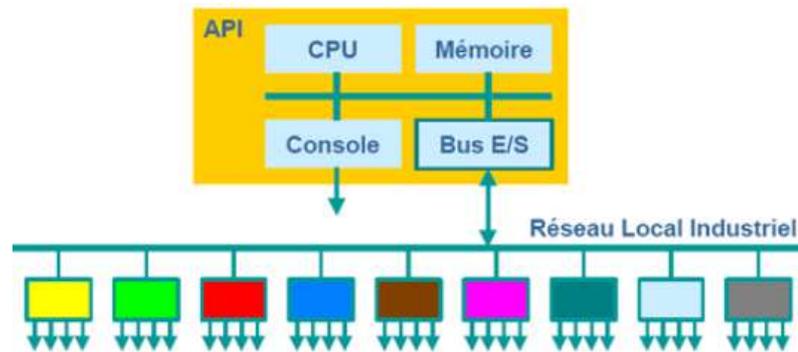


Figure V-23. Interconnexion par entrées/sorties déportées

V.12.2.1. Avantages

- ✓ Réduction des coûts de câblage et possibilité de réutiliser le matériel existant ;
- ✓ Réduction des coûts de maintenance.

V.12.2.2. Inconvénients

- * Taille du réseau limité ;
- * Latence dans les applications à temps critique ;
- * Coût global.

V.12.3. Différents Types de Réseaux d'Automates

V.12.3.1. Réseau en étoile

Un centre de traitement commun échange avec chacune des autres stations. Deux stations ne peuvent pas échanger directement entre elles (Fig.V-24). Exemple le réseau de terrain BITBUS de la société INTEL.

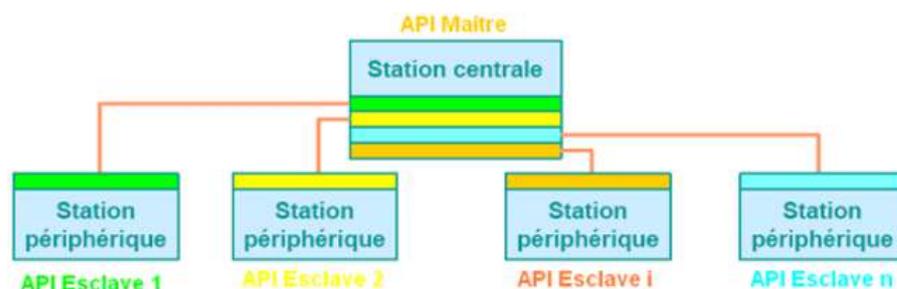


Figure V-24. Réseau des automates en étoile

➤ **Avantage :**

- ✓ Grande vitesse d'échange ;
- ✓ Différent types de supports de transmission ;
- ✓ Pas de gestion d'accès au support.

➤ **Inconvénients :**

- ✗ Coût global élevé ;
- ✗ Evolutions limitées ;
- ✗ Toute repose sur la station centrale.

V.12.3.2. Réseau en anneau

Chaque station peut communiquer avec sa voisine. Cette solution est intéressante lorsqu'une station doit recevoir des informations de la station précédente ou en transmettre vers la suivante (Fig.V-25).

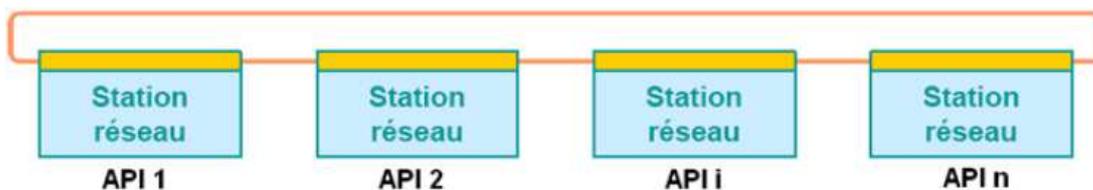


Figure V-25. Réseau des automates en anneau

➤ **Avantage :**

- ✓ Signal régénéré donc fiable ;
- ✓ Contrôle facile des échanges (le message revient à l'émetteur) ;

➤ **Inconvénients :**

- ✗ Chaque station est bloquante ;
- ✗ Une extension interrompt momentanément le réseau ;

V.12.3.3. Réseau en hiérarchisé

C'est la forme de réseaux la plus performante. Elle offre une grande souplesse d'utilisation, les informations pouvant circuler entre-stations d'un même niveau ou circuler de la station la plus évoluée (en général un ordinateur) vers la plus simple, et réciproquement (Fig.V-21).

V.13. CONCLUSION

Ces dernières années, les avancées technologiques ont conduit au développement des automates programmables industriels (API) et à une révolution importante dans l'automatique.

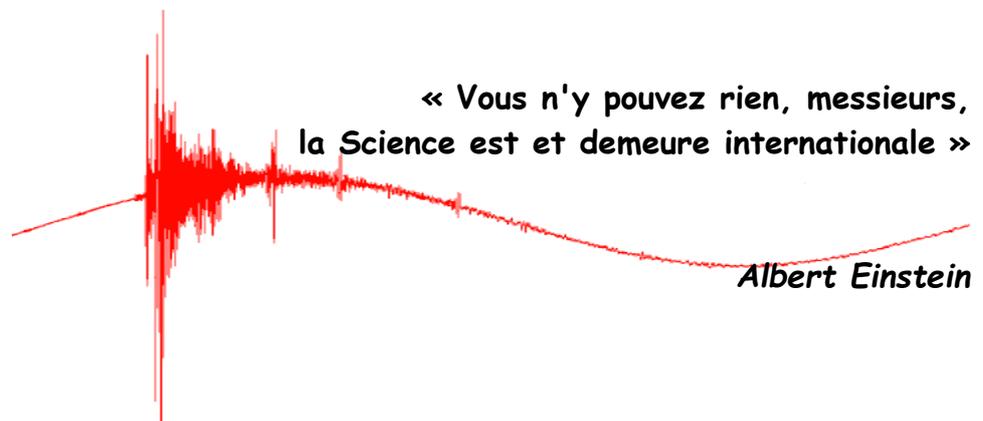
L'API est un bon produit s'il est bien choisi et bien employé. Ce qui peut apparaître comme une lapalissade nous a amené à attirer l'attention sur des aspects parfois jugés triviaux, tels les types d'E/S, le dimensionnement des alimentations électriques, les modes d'exécution d'un programme, les limites des divers types de communication, car ce sont des points où sont parfois commises des erreurs qui entraînent des surcoûts d'installation ou limitent fâcheusement les performances obtenues.

Ce chapitre constitue une introduction aux automates programmables industriels (API), ainsi qu'à leur fonction générale, leurs formes matérielles et leur architecture interne. Les API sont employés dans de nombreuses tâches d'automatisation, dans différents domaines, comme les processus de fabrication industriels.

Après avoir présenté les architectures des automates programmables industriels (API), on peut passer à leurs programmations, sujet du chapitre suivant.

CHAPITRE VI

PROGRAMMATION DES API



CHAPITRE VI

PROGRAMMATION DES API

VI.1. INTRODUCTION

La mise en œuvre d'un automate programmable comporte trois phases principales : la *conception* du système de contrôle, qui se fait à l'aide d'outils méthodologiques et de modes de représentation propres à l'automaticien ; la *programmation* de l'automate, qui consiste à transposer le système de contrôle obtenu dans le langage de programmation propre à l'automate et l'*exécution* du programme, enfin prise en charge par un logiciel interne à l'automate.

Comme nous l'avons dit au chapitre précédent, tout automate peut résoudre n'importe quel problème de logique combinatoire ou séquentiel. Il est bien certain cependant que la facilité d'utilisation d'un automate et ses performances seront directement fonction de la cohérence qui existera entre les trois phases principales mentionnées ci-dessus.

Dans le présent chapitre, j'essayerai de faire le point de la situation en insistant sur la programmation séquentielle qui a donné lieu à de nombreuses initiatives de la part des constructeurs, principalement des constructeurs européens.

VI.2. NORME CEI 1131

La normalisation des langages en informatique est un fait acquis (Pascal, C, C++, etc ...). C'est un critère d'achat, aujourd'hui il n'existe plus sur le marché des progiciels de ce type qui ne respecteraient pas les normes. En automatismes, bien que la part des automates programmables dans les systèmes automatisés soit très importante, la diversité est de règle. La part du logiciel devenant de plus en plus importante dans les applications, la nécessité de passer d'une programmation quasi artisanale à une certaine harmonisation s'est petit à petit imposée aux utilisateurs ainsi qu'aux constructeurs.

Les besoins d'une normalisation des langages pour l'API s'expriment pour les industriels, en termes de [42] :

- faciliter la formation des réalisateurs de configurations d'automates programmables ;
- obtenir un bon niveau de portabilité des programmes ;
- favoriser la création de bibliothèques de blocs fonctionnels fiables ;
- faciliter les configurations en réseau d'automates ;
- améliorer la qualité des applications (sûreté de fonctionnement, maintenabilité, extensibilité) ;
- réutiliser les outils de configuration et de programmation ;
- produire des dossiers d'applications homogènes ;
- faciliter la maintenance du logiciel d'application.

Le CENELEC (Comité Européen de Normalisation Electrotechnique) a adopté en 1993 le texte (65B) de la CEI (Commission Electrotechnique Internationale) comme norme EN 61131 pour la commande des processus industriels. Cette norme traite des automates programmables en cinq parties :

- ▶ CEI 1131-1 définitions, informations générales ;
- ▶ CEI 1131-2 spécifications et essais matériels ;
- ▶ CEI 1131-3 langages de programmations ;
- ▶ CEI 1131-4 documentations ;
- ▶ CEI 1131-5 communications.

L'originalité des langages de programmation pour automates est qu'ils sont généralement imagés par rapport à l'expression de la commande des machines automatisées. Ils sont souvent graphiques et depuis vingt ans des formes de langages se sont dégagés et sont mise à disposition par les constructeurs d'API (liste d'instruction mnémonique, réseau à contacts, GRAFCET).

La norme commence à être utilisée, des associations d'utilisateurs se sont créées en Europe (EXERA, PLCopen) pour certifier des produits à un niveau de

conformité de *base* et un niveau de *probabilité*. Chez les constructeurs des langages de programmation *proches* sont annoncés (XTel et PL7Micro pour télémécanique, SFCd'Allen Bradley, Step5 et 7 de Siemens, etc...) des tests de conformité sont fait par le CETIM relatifs aux prescriptions et à la syntaxe. Les nouveaux ateliers logiciels pour le Génie automatique sont développés pour fonctionner sur des PC avec des interfaces standard et des langages normalisés.

L'étude de la norme nous permettra de constater des principes largement éprouvés par le génie logiciel tels que la structuration et la modularité. De distinguer les modules logiciels des langages dans lesquels ils peuvent être écrits. De faire la différence entre le modèle de spécification et le langage de programmation (code automate).

La norme CEI 1131 s'applique aux automates programmables industriels et à leurs périphériques. Les objectifs sont [42] :

- ☞ donner les définitions et identifier les principales caractéristiques permettant de sélectionner et utiliser les automates programmables et leurs périphériques associés ;
- ☞ déterminer les prescriptions minimales relatives aux caractéristiques fonctionnelles, aux conditions de service, aux caractéristiques constructives, à la sécurité générale ainsi qu'aux essais applicables aux automates programmables et à leurs périphériques ;
- ☞ définir pour les langages de programmation les principaux champs d'applications, les règles syntaxiques et sémantiques ainsi que des ensembles de base simples mais exhaustifs d'éléments de programmation ;
- ☞ fournir à l'utilisateur des informations générales didactiques et des recommandations quant à son application ;
- ☞ définir les langages de programmation ;
- ☞ définir les modules logiciels ou unités d'organisation de programmes.

VI.3. FONCTIONNEMENT DE L'API

Lorsque l'API est en fonctionnement, trois phases se succèdent. L'enchaînement des trois phases se répète sans cesse de façon cyclique lorsque l'API est en fonctionnement (Fig. VI-1) [17,20,23,41] :

1. Lecture des entrées (%I)
2. Traitement du programme (T)
3. Ecriture des sorties (%Q)

Avant chaque cycle l'API effectue des traitements internes afin de vérifier ses circuits et les sollicitations extérieures. Le temps de cycle de l'ordre de quelques milli-secondes est surveillé par un circuit électronique appelé *chien de garde*.

Si pour une raison quelconque le temps de cycle mesuré par le chien de garde est supérieur au temps de cycle maxi configuré, l'API signale le défaut arrête le traitement.

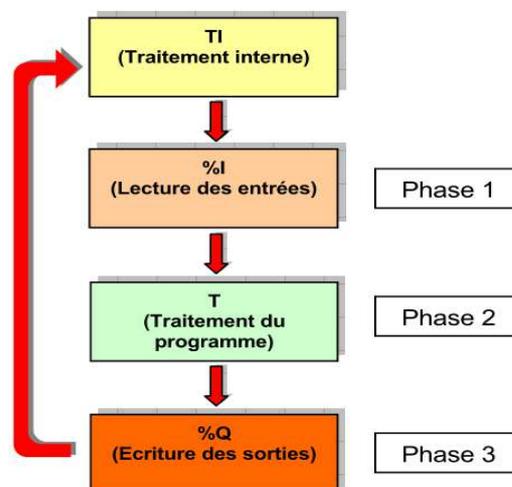


Figure VI-1. Phases de fonctionnement

VI.3.1. Entrées

Durant cette phase qui dure quelques micro-secondes (Fig. VI-2) :

- ◆ les entrées sont *photographiées* et leurs états logiques sont stockés dans une zone spécifique de la mémoire de donnée ;
- ◆ Le programme n'est pas scruté ;
- ◆ Les sorties ne sont pas mises à jour.

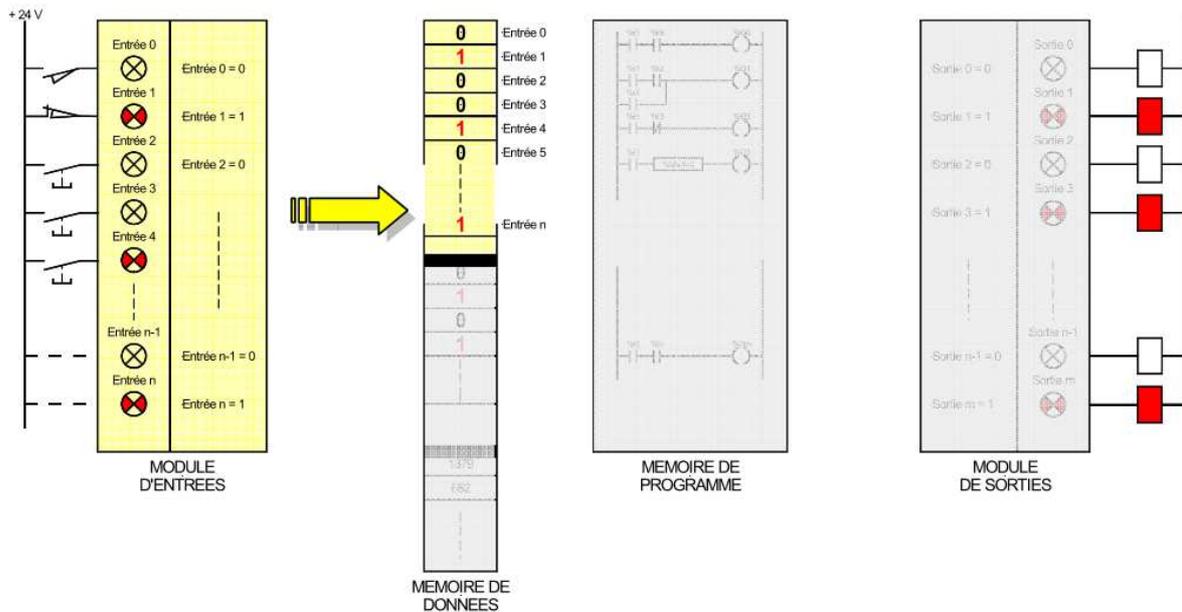


Figure VI-2. Entrées de l'automate programmable industriel

VI.3.2. Exécution du Programme

Durant cette phase qui dure quelques milli-secondes (Fig. VI-3) :

- ◆ Les instructions de programme sont exécutées une à une. Si l'état d'une entrée doit être lu par le programme, c'est la valeur stockée dans la mémoire de données qui est utilisée ;
- ◆ Le programme détermine l'état des sorties et stocke ces valeurs dans une zone de la mémoire de données réservée aux sorties ;
- ◆ Les entrées ne sont pas scrutées ;
- ◆ Les sorties ne sont pas mises à jour.

Notez que pendant cette phase, seules la mémoire de données et la mémoire programme sont mises à contribution. Si une entrée change d'état sur le module d'entrées, l'API ne voit pas ce changement.

VI.3.3. Mise à Jour des Sorties

Durant cette phase qui dure quelques milli-secondes (Fig. VI-4) :

- ◆ Les états des sorties mémorisés précédemment dans la mémoire de données sont reportés sur le module de sorties ;

- ◆ Les entrées ne sont pas scrutées ;
- ◆ Le programme n'est pas exécuté.

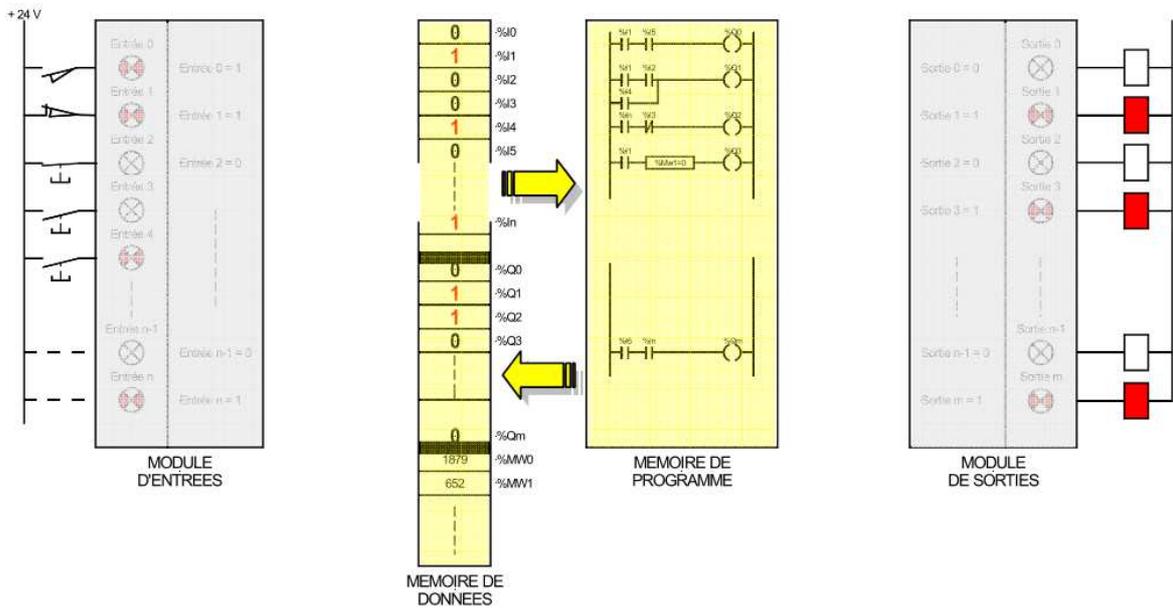


Figure VI-3. Exécution du Programme

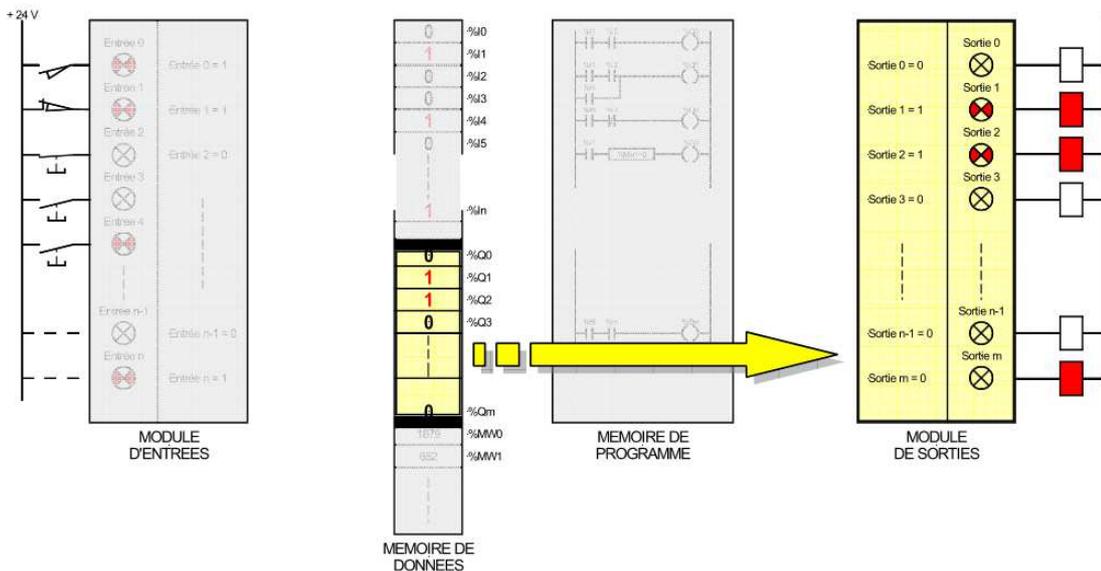


Figure VI-4. Mise à jour des sorties

VI.4. STRUCTURE LOGICIELLE MONOTACHE

Le programme d'une application monotâche est associé à une seule tâche utilisateur la tâche maître MAST. L'exécution de la tâche maîtresse peut être : cyclique ou périodique.

VI.4.1. Exécution Cyclique

La durée d'un cycle ne doit pas dépasser le temps réglé pour le chien de garde (Fig. VI-5) (Tableau VI-1).

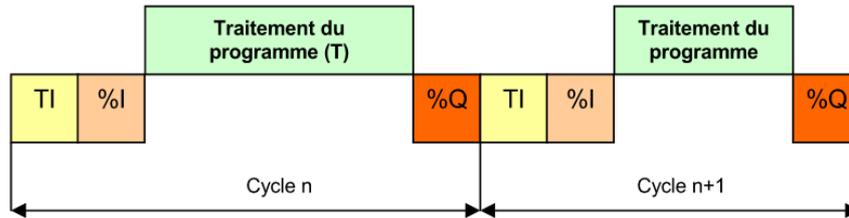


Figure VI-5. Exécution cyclique

Tableau VI-1. Description de l'exécution cyclique [43]

Repère	Phase	Description
TI	Traitement Interne	Le système réalise implicitement la surveillance de l'automate (gestion des bits et mots système, mise à jour des valeurs courantes de l'horodateur, mise à jour des voyants d'état, détection des passages RUN/STOP, ...) et le traitement des requêtes en provenance du terminal (modifications et animation).
% I	Acquisition des Entrées	Ecriture en mémoire de l'état des informations présentes sur les entrées des modules TOR et métier associées à la tâche.
T	Traitement du programme	Exécution du programme application, écrit par l'utilisateur.
% Q	Mise à jour des Sorties	Ecriture des bits ou des mots de sorties associés aux modules TOR et métier associés à la tâche selon l'état défini par le programme application.

VI.4.2. Exécution Périodique

Dans ce mode de fonctionnement, l'acquisition des entrées, le traitement du programme application et la mise à jour des sorties s'effectuent de façon périodique selon un temps défini en configuration (de 1 à 255 ms) (Fig. VI-6).

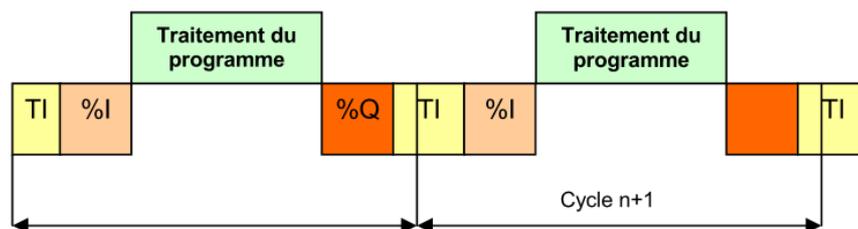


Figure VI-6. Exécution Périodique

En début de cycle automate, un temporisateur dont la valeur courante est initialisée à la période définie en configuration, commence à décompter.

Le cycle automate doit se terminer avant l'expiration de ce temporisateur qui relance un nouveau cycle.

Le processeur effectue dans l'ordre le traitement interne, l'acquisition des entrées, le traitement du programme application et la mise à jour des sorties. Si la période n'est pas encore terminée, le processeur complète son cycle de fonctionnement jusqu'à la fin de la période par du traitement interne. Si le temps de fonctionnement devient supérieur à celui affecté à la période, l'automate signale un débordement de période par la mise à l'état 1 du bit système %S19 de la tâche, le traitement se poursuit et est exécuté dans sa totalité (il ne doit pas dépasser néanmoins le temps limite du chien de garde). Le cycle suivant est enchaîné après l'écriture implicite des sorties du cycle en cours [43].

VI.5. STRUCTURE LOGICIELLE MULTI-TACHES

La tâche maître est par défaut active. La tâche rapide est par défaut active si elle est programmée. Le traitement événementiel est activé lors d'apparition de l'événement qui lui a été associé (Fig. VI-7) (Fig. VI-8).

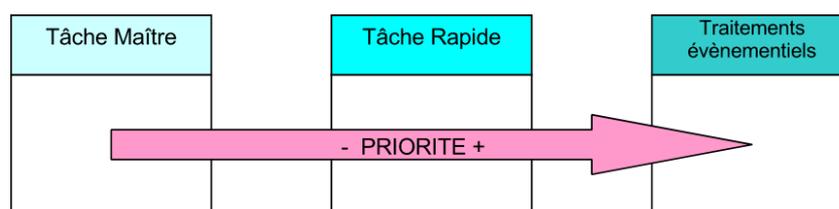


Figure VI-7. Mise à jour des sorties

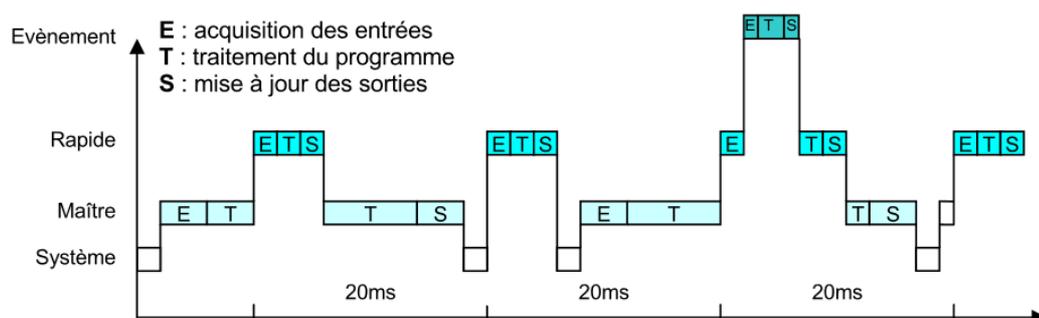


Figure VI-8. Traitement événementiel multi-tâches

VI.6. LANGAGES DE PROGRAMMATION POUR API

Les langages de programmation utilisés pour les automates programmables évoluent depuis la création des APIs à la fin des années 1960. Dans ce contexte, la commission internationale d'électrotechnique (CIE) définit six langages de programmation des automates programmables (IEC 61131-3), celles-ci consistent en [20,22-29] :

➤ *Deux langages textuels :*

- ☞ IL : liste d'instructions ;
- ☞ ST : texte structuré.

➤ *Quatre langages graphiques :*

- ☞ LD : schéma à contacts ;
- ☞ FBD : schéma fonctionnel ;
- ☞ SFC : Sequential Function Chart ;
- ☞ GRAFCET.

Ils sont définis pour structurer l'organisation interne des programmes et des blocs fonctionnels. Dans ce chapitre, je présenterai que trois types de langages utilisés dans les automates aujourd'hui : IL, FBD et LADDER.

VI.6.i. C'est quoi un Programme ?

Le programme est défini comme *“un ensemble logique de tous les éléments et constructions des langages de programmation nécessaires pour le traitement des données requis pour contrôler une machine ou un processus au moyen d'une configuration d'automate programmable”*. Un programme est construit à l'aide de Fonctions et de Blocs Fonctionnels.

Un programme est une unité logique de programmation qui décrit des opérations entre les variables de l'application. Un programme décrit des opérations séquentielles ou cycliques. Un programme cyclique est exécuté systématiquement à chaque cycle automate.

L'exécution d'un programme séquentiel respecte les règles d'évolution du langage utilisé. Un programme ne peut être instancié que dans des ressources, alors que les blocs fonctionnels ne peuvent être instanciés que dans des programmes ou dans d'autres blocs fonctionnels.

Une unité de programmation peut être un appareil portatif (Fig.VI-9), un terminal de bureau ou un ordinateur. Après que le programme a été conçu et testé sur l'unité de programmation, il est prêt à être transféré dans la mémoire de l'API.

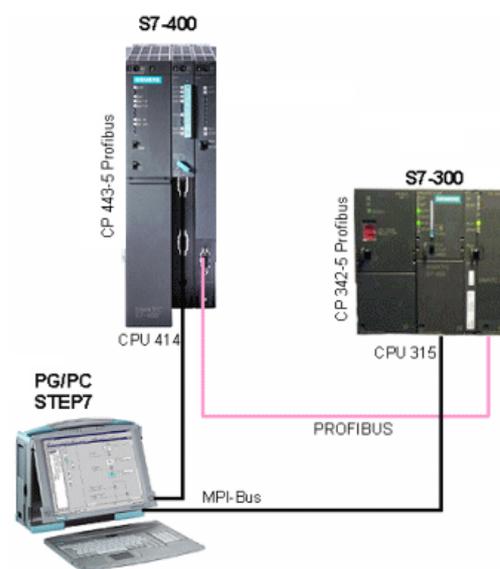


Figure VI-9. Unité de programmation

Un appareil de programmation portatif généralement d'une quantité de mémoire suffisante pour conserver les programmes afin de les déplacer d'un endroit à un autre. Les terminaux de bureau sont généralement équipés d'un système d'affichage graphique, avec un clavier et un écran.

Les ordinateurs personnels sont souvent employés pour la programmation des API. L'utilisation d'un ordinateur présente plusieurs avantages : le programme peut être stocké sur le disque dur ou sur un CD-ROM et les copies sont faciles à réaliser. L'ordinateur est relié à l'API par l'intermédiaire d'un câble Ethernet, RS-232, RS-485 ou RS-422 [20,29].

VI.6.2. Adressages des Entrées et des Sorties d'un API

Avant d'entamer les trois langages de programmation des API, nous devons savoir l'adressage des entrées et des sorties des API. Il est très important de réaliser l'affectation des entrées et des sorties de l'automate et connaître son adressage. Pour ce faire, il attribue des adresses à chaque entrée et à chaque sortie. Avec un petit automate, il s'agit probablement d'un nombre, préfixé par une lettre pour indiquer s'il s'agit d'une entrée ou d'une sortie. Par exemple, pour l'automate Mitsubishi, nous pourrions avoir des entrées avec les adresses X400, X401, X402, etc., et des sorties avec adresses Y430, Y431, Y432, etc., le symbole X indiquant une entrée et le symbole Y une sortie.

Pour les automates les plus grands dont composés plusieurs racks, les racks sont numérotés. L'exemple avec le PLC-5 Allen-Bradley, le rack contenant le processeur reçoit le numéro 0 et les adresses des autres racks sont numérotées 1, 2, 3, etc. Chaque rack peut avoir un certain nombre de cartes et chacune contient un certain nombre d'entrées et / ou de sorties. Ainsi, les adresses peuvent être de la forme illustrée à la Figure (VI-10). Par exemple, nous pourrions avoir une entrée avec l'adresse $I : 023/04$. Cela indiquerait *une entrée, rack 02, module 3 et l'entrée 04*.

$I : 02\ 3 / 04$	$I : \text{Entrée} ; Q : \text{Sortie}$
$Q : 02\ 3 / 04$	$02 : \text{Rack Numéro } 02$
	$3 : \text{Module Numéro } 3$
	$04 : \text{Entrée Numéro } 04$

Figure VI-10. Adressage PLC-5 Allen-Bradley

Pour le constructeur allemand Siemens SIMATIC S7, les entrées et les sorties sont réparties en groupes de huit. Chacun de ces groupes est appelé un octet et chaque entrée ou sortie d'un groupe de huit est appelée bit. Les entrées et les

sorties ont donc leurs adresses en nombre d'octets et de bits, ce qui donne un numéro de module suivi d'un numéro de terminal, et un point "." Séparant les deux numéros.

La Figure (VI-11) montre l'adressage d'une entrée/sortie dans un automate Siemens. Par exemple, **Io.1** est une *entrée* au *bit 1* dans l'*octet 0* et **Q2.0** est une *sortie* au *bit 0* dans l'*octet 2*.

I o.1
Q o.2

I : Entrée ; Q : Sortie
o : Numéro du bit
1 : Numéro d'octet
2 : Numéro d'octet

Figure VI-11. Adressage Siemens SIMATIC S7

L'adressage consiste à identifier les variables d'E/S et les variables internes par des adresses. La notion d'adressage d'un automate permet de connaître le type d'objet, le format des données que l'on va pouvoir former et son emplacement (ou numéro). *Le mode d'adressage dépend du constructeur de l'API. Il faut donc consulter la notice de cet automate.*

Le tableau (VI-2) montre la signification des différents symboles que l'on peut trouver dans une adresse.

Tableau VI-2. Signification des différents symboles d'une adresse [20]

Syntaxes	Significations	Exemples
I (ou E)	Lecture de l'état d'une entrée.	Io.4 : bit d'entrée 4 module 0
Q(ou A)	Lecture/ Ecriture de l'état d'une sortie	IW125 : mot d'entrée 125
M et V	Lecture/ Ecriture de l'état d'une variable interne	Qo.4 : bit de sortie 4 module 1
SM	Lecture/ Ecriture d'un bit Système	QB17 : Octet de sortie 17
S	Lecture/Ecriture d'un bit Relai séquentiel	M12.15 : bit mémoire interne 12.15
C ou (Z)	Compteur	MD48 : mot double mémoire interne 48
SC	Compteur rapide	AIW288 : Entrée Analogique 288
T	Temporisateur	AQW304 : Sortie Analogique 304
A ou (P)	Analogique	
%	Norme IEC 1131-3	
B	Taille d'un byte ou octet	
W	Taille d'un word : mot de 16 bits	
D	Taille d'un double word : mot double de 32 bits	

VI.6.3. Langage de Programmation LADDER

Ladder Diagram (LD) ou *Langage Ladder* ou *schéma à contacts* est un langage graphique très populaire auprès des automaticiens pour programmer les APIs. Il ressemble un peu aux schémas électriques, et est facilement compréhensible. L'idée initiale du Ladder est la représentation de fonction logique sous la forme de schémas électriques. Par exemple, pour réaliser un ET logique avec des interrupteurs, il suffit de les mettre en série. Pour réaliser un OU logique, il faut les mettre en parallèle.

Le Ladder a été créé et normalisé dans la norme CEI 61131-3. Il est encore aujourd'hui souvent utilisé dans la programmation des automates programmables industriels, bien qu'ayant tendance à être délaissé en faveur de langages plus évolués, et plus adaptés aux techniques modernes de programmation, tels que le ST par exemple, ou encore le Grafcet, plus adapté à la programmation de séquences. Aussi l'avantage de ce langage et dans sa puissance en point de vue diagnostic. La Figure (VI-12) présente quelques symboles utilisés dans le langage Ladder.

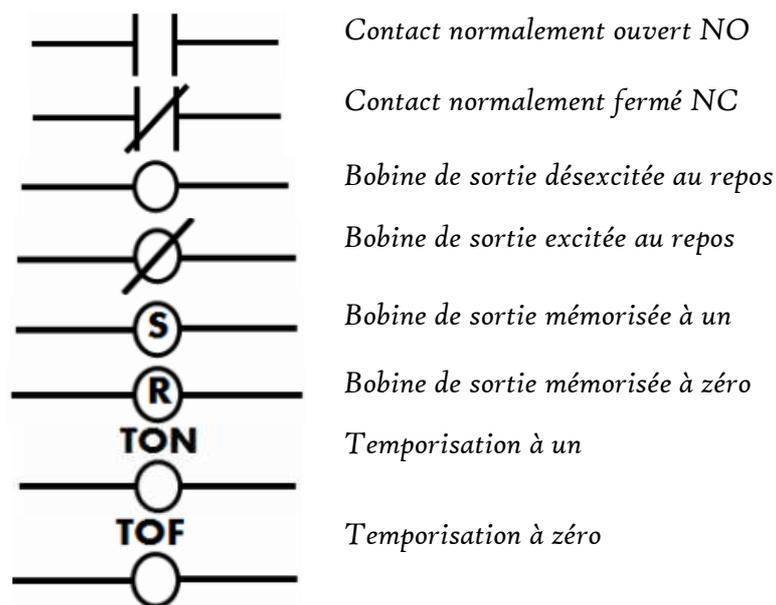


Figure VI-12. Symboles de le langage Ladder

VI.6.4. Présentation du Langage LADDER

Les fonctions principales d'un programme *Ladder* sont de contrôler les sorties et d'effectuer des opérations fonctionnelles en fonction des conditions d'entrée. Le Ladder utilise des barreaux (Rungs) pour réaliser ce contrôle. La Figure (VI-13) montre un exemple du langage Ladder. En général, la ligne droite (L1) consiste en un ensemble des conditions d'entrée (représentées par des instructions de contact) et les instructions de sortie (bobines) sont connectées dans la ligne gauche (L2).

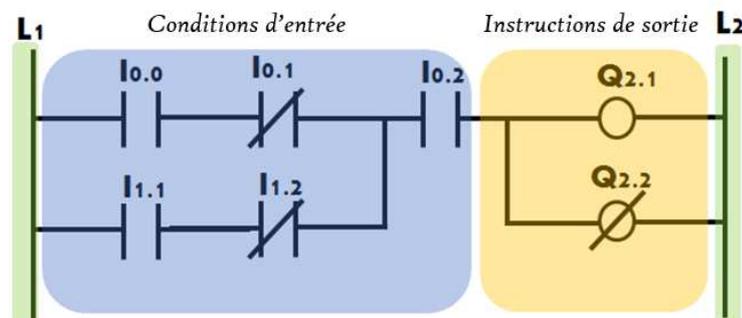


Figure VI-13. Présentation du langage Ladder

VI.6.5. Présentation du Langage FBD

Le diagramme *FBD* (*Function Block Diagram*) décrit une fonction entre des variables d'entrée et des variables de sortie. Une fonction est décrite comme un réseau de blocs élémentaires. Les variables d'entrée et de sortie sont connectées aux blocs par des arcs de lien. Une sortie d'un bloc peut aussi être connectée sur une entrée d'un autre bloc (Fig. VI-14).

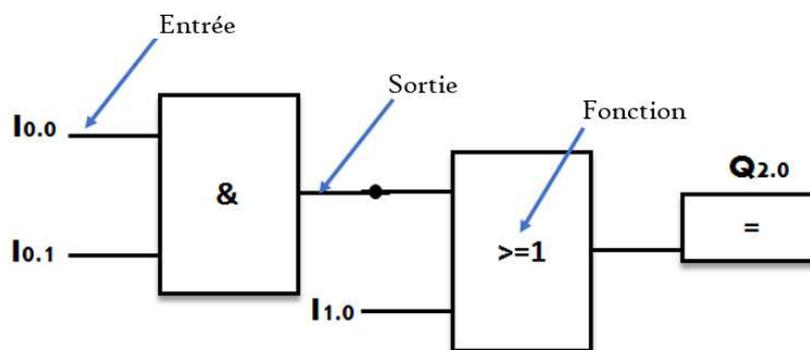


Figure VI-14. Présentation du langage FBD

VI.6.5. Présentation du Langage IL

Le langage de programmation *textuelle LIST* permet de créer des programmes d'application à un niveau proche du matériel et en optimisant le temps d'exécution et la place en mémoire (Fig. VI-15).

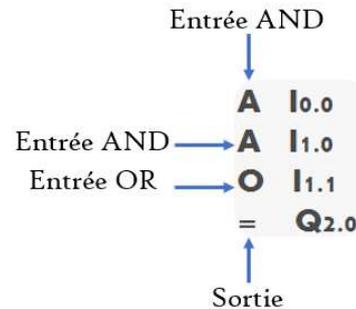


Figure VI-15. Présentation du langage IL

Le langage à liste d'instructions est composé d'une suite d'instruction, chaque instruction doit débiter une nouvelle ligne de programme et doit contenir un opérateur suivi d'une ou plusieurs opérandes. Les opérateurs standard du langage IL normalisés sont repris au Tableau (VI-3).

Tableau VI-3. Opérations de base du langage Liste d'Instructions

Opération	Commentaire
LD	Charger la valeur de l'opérande
LDN	Charger la valeur inverse de l'opérande
A	ET logique entre le résultat précédent et l'état de l'opérande
AN	ET logique entre le résultat précédent et l'état inverse de l'opérande
O	OU logique entre le résultat précédent et l'état de l'opérande
ON	OU logique entre le résultat précédent et l'état inverse de l'opérande
NOT	Inverser le résultat
EU	Détecter front montant
ED	Détecter front descendant
=	Ecrire la sortie
S	Positionner l'opérande à un
R	Remettre l'opérande à un
LDB=	Comparaison de type égalité
AB=	Comparaison de type égalité précédé d'un contact en série
OB=	Contact en parallèle avec la Comparaison de type égalité
BTI, ITB	Conversion d'un octet en entier, Conversion d'un entier en octet.
+I ; -I ; *I ; /I	Addition, Soustraction, Multiplication et Division de deux nombres entiers
INCB ; DECB	Incrémentatation et décrémentation d'un octet
INVB ; ANDB ; ORB	Inversion d'un octet, et logique de deux octets, ou logique de deux octets
MOVB	Transférer un octet
SLB ; SRW	Décalage vers la gauche d'un octet, Décalage vers la droite d'un mot
RLB ; RRD	Rotation vers la gauche d'un octet, Décalage vers la droite d'un double mot

🔧 Exemple de programme en Ladder et IL :

Soit un circuit électrique d'allumage d'une lampe de 220V. La lampe s'allume que si les deux interrupteurs K₁ et K₂ sont fermés. Si l'un ou les deux s'ouvrent la lampe s'éteint. Comme on peut le constater sur la Figure (VI-16), le fait que les deux interrupteurs K₁ et K₂ sont en série, le courant arrivera à la lampe (le circuit se fermera) que si les deux interrupteurs sont fermés. Dans ce cas de figure, la lampe affiche le résultat d'un ET logique (AND) de K₁ et K₂.

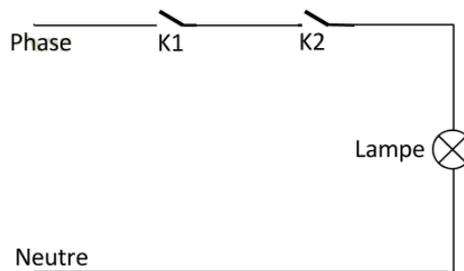


Figure VI-16. Circuit d'une lampe actionnée par deux interrupteurs en série

Pour automatiser cet exemple en utilisant un API, deux points seront considérés :

- ◆ la connexion (Fig.VI-17) :
 - en entrées des deux capteurs traduisant les états de K₁ et K₂.
 - en sortie de la lampe.
- ◆ Le programme à écrire pour l'API pour réaliser cette tâche.

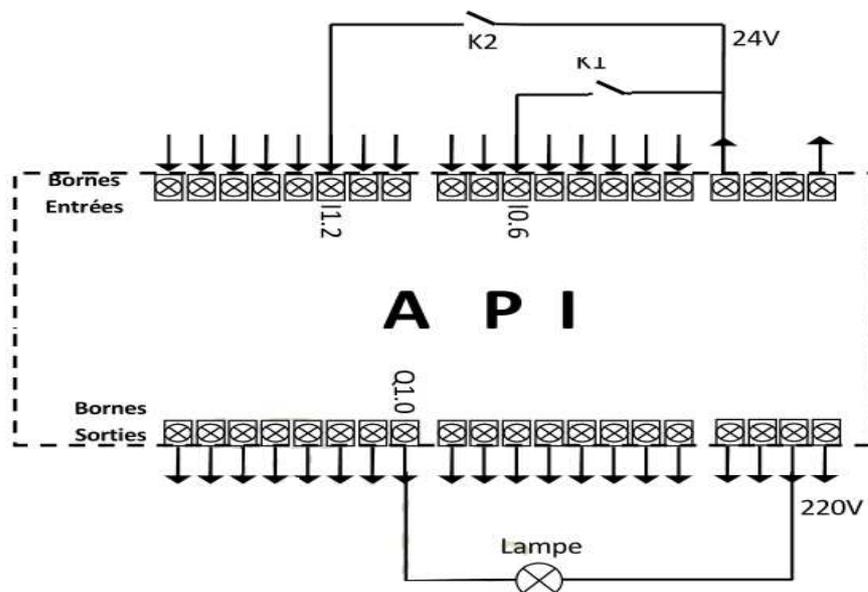


Figure VI-17. Câblage d'une lampe actionnée par deux interrupteurs à un API

La Figure (VI-18) présente le programme de la Figure (VI-17) avec les deux langages Ladder et IL :

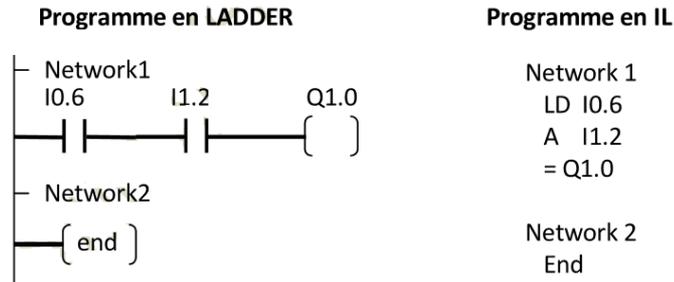


Figure VI-18. Programme avec les deux langages Ladder et IL a un API

VI.7. PROGRAMMATION DES API

Dans cette partie je vais présenter quelques applications le plus utilisé dans l'industrie et chaque application est accompagner avec des exemples en trois langages cités auparavant.

VI.7.1. Fonctions Logiques

Dans ce qui suite je vais donner quelques fonctions logiques le plus utilisés en trois langages tel que Ladder, FBD et LIST.

VI.7.1.1. Fonction logique AND

Le Tableau (VI-4) montre, la table de vérité de la fonction AND :

Tableau VI-4. Table de vérité de la fonction AND

Entrée 1	Entrée 2	Sortie Y = A.B
I _{0.0}	I _{0.1}	Q _{1.0}
0	0	0
0	1	0
1	0	0
1	1	1

La Figure (VI-19) présente la fonction AND avec les trois langages :

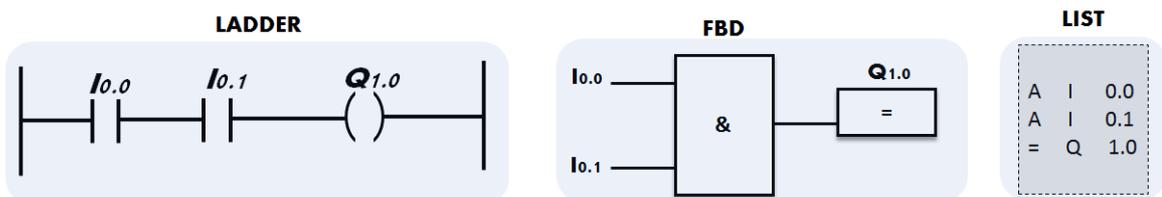


Figure VI-19. Fonction AND sous l'API

VI.7.1.2. Fonction logique OR

Le Tableau (VI-5) montre, la table de vérité de la fonction OR :

Tableau VI-5. Table de vérité de la fonction OR

Entrée 1	Entrée 2	Sortie $Y=A+B$
$I_{0.0}$	$I_{0.1}$	$Q_{1.0}$
0	0	0
0	1	1
1	0	1
1	1	1

La Figure (VI-20) présente la fonction OR avec les trois langages :

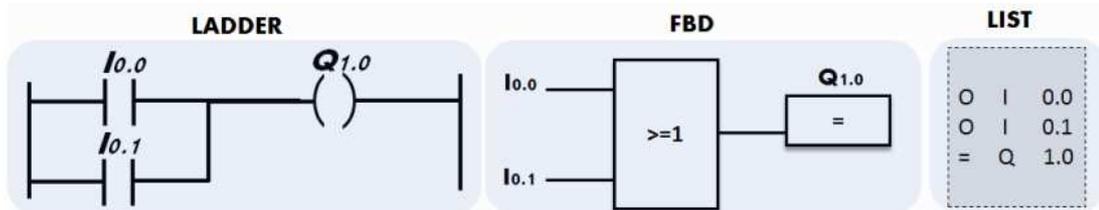


Figure VI-20. Fonction OR sous l'API

VI.7.1.3. Fonction logique NOT

Le Tableau (VI-6) montre, la table de vérité de la fonction NOT :

Tableau VI-6. Table de vérité de la fonction NOT

Entrée 1	Sortie $Y=\bar{A}$
$I_{0.0}$	$Q_{1.0}$
0	1
1	0

La Figure (VI-21) présente la fonction NOT avec les trois langages :

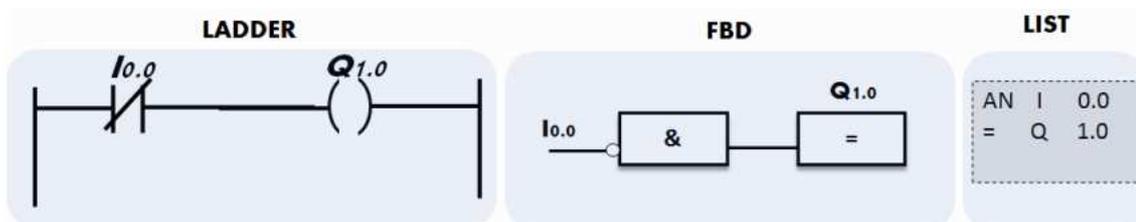


Figure VI-21. Fonction NOT sous l'API

VI.7.1.4. Fonction logique NAND

Le Tableau (VI-7) montre, la table de vérité de la fonction NAND :

Tableau VI-7. Table de vérité de la fonction NAND

Entrée 1	Entrée 2	Sortie $Y = \overline{A \cdot B}$
$I_{0.0}$	$I_{0.1}$	$Q_{1.0}$
0	0	1
0	1	1
1	0	1
1	1	0

La Figure (VI-22) présente la fonction NAND avec les trois langages :

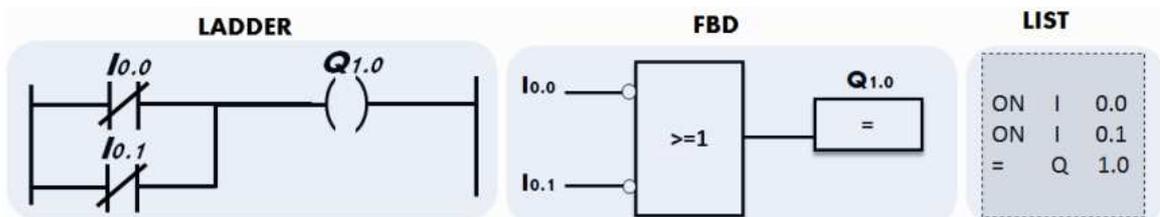


Figure VI-22. Fonction NAND sous l'API

VI.7.1.5. Fonction logique NOR

Le Tableau (VI-8) montre, la table de vérité de la fonction NOR :

Tableau VI-8. Table de vérité de la fonction NOR

Entrée 1	Entrée 2	Sortie $Y = \overline{A + B}$
$I_{0.0}$	$I_{0.1}$	$Q_{1.0}$
0	0	1
0	1	0
1	0	0
1	1	0

La Figure (VI-23) présente la fonction NOR avec les trois langages :

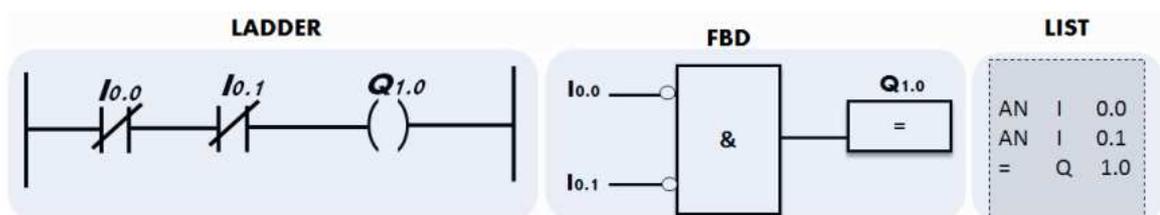


Figure VI-23. Fonction NOR sous l'API

VI.7.1.6. Fonction logique XOR

Le Tableau (VI-9) montre, la table de vérité de la fonction XOR :

Tableau VI-9. Table de vérité de la fonction XOR

Entrée 1	Entrée 2	Sortie $Y = A \oplus B$
$I_{0.0}$	$I_{0.1}$	$Q_{1.0}$
0	0	0
0	1	1
1	0	1
1	1	0

La Figure (VI-24) présente la fonction XOR avec les trois langages :

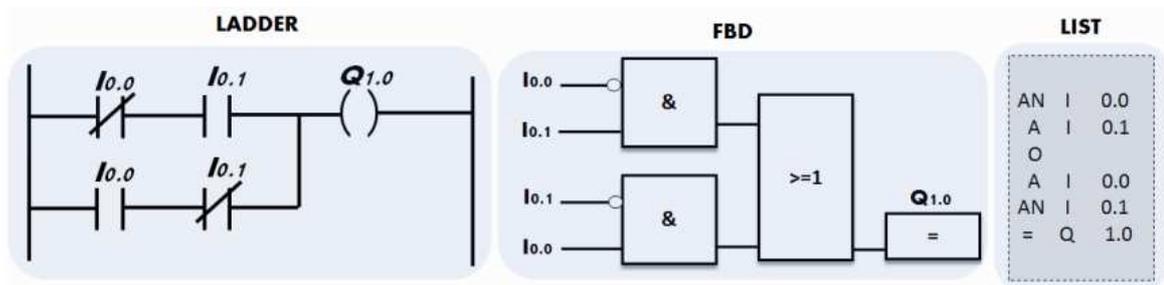


Figure VI-24. Fonction XOR sous l'API

VI.7.2. Fonctions de Mémorisation

Il existe souvent des situations dans lesquelles il est nécessaire de maintenir une sortie sous tension, quelque soit l'état de l'entrée. La Figure (VI-25) illustre un exemple simple d'une telle situation dans le cas de démarrage du moteur, en appuyant sur un bouton poussoir marche (start). Bien que le contact du bouton poussoir ne reste pas fermé, le moteur doit continuer à fonctionner jusqu'à l'appui sur le bouton poussoir d'arrêt (stop), On appelle le contacte $Q_{1.0}$ l'auto-maintien.

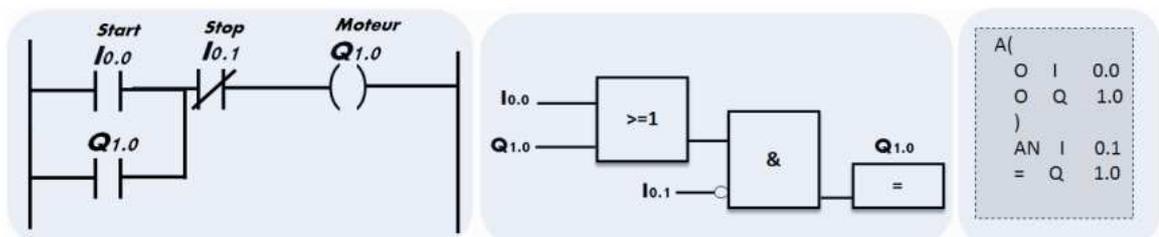


Figure VI-25. Fonction de mémorisation

La Figure (VI-26) montre un exemple réel, du programme sous l'API Siemens S7, d'un démarrage direct d'un moteur avec des lampes de signalisation d'état du moteur (Marche, Arrêt).

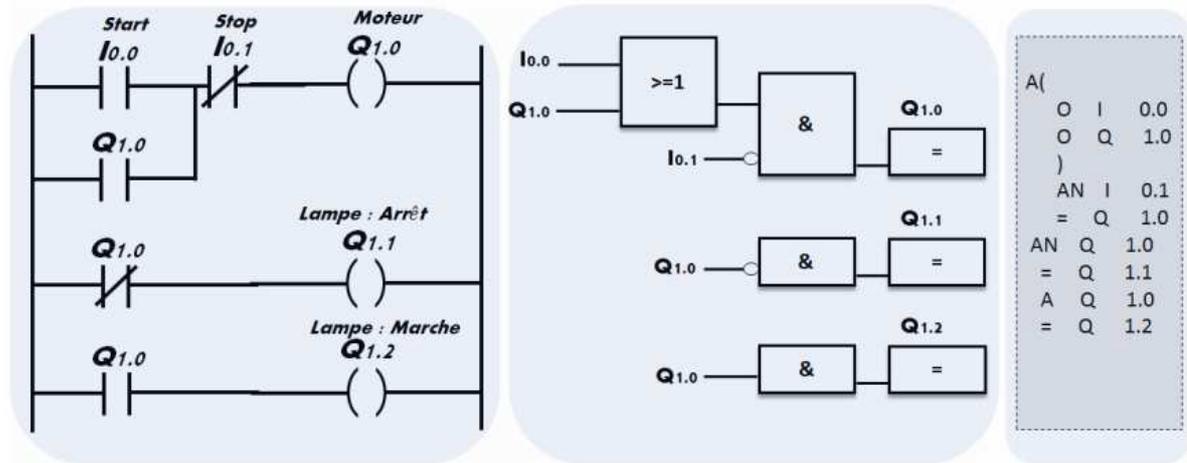


Figure VI-26. Fonction de mémorisation pour le démarrage d'un moteur

VI.7.3. Temporisations

Dans de nombreuses applications de contrôle à base d'automates programmables, il est nécessaire de contrôler le temps. Par exemple, un moteur doit fonctionner pendant un intervalle de temps particulier ou peut être allumé après un certain intervalle de temps ... etc.

Les automates programmables ont donc des minuteries en tant que périphériques intégrés. Les minuteries comptent les secondes ou fractions de seconde en utilisant l'horloge interne du processeur. Dans cette partie je vais montrer les différents types des minuteries les plus utilisés.

VI.7.3.1. Temporisateur de travail (ON-Delay Timer)

Généralement tous les APIs sont dotés de ce type de temporisateur dont le fonctionnement est présenté dans la Figure (VI-27). Si l'entrée IN est activée le temporisateur TON va déclencher la temporisation pendant le temps alloué TV, après ce temps, le TON va activer la sortie OUT.

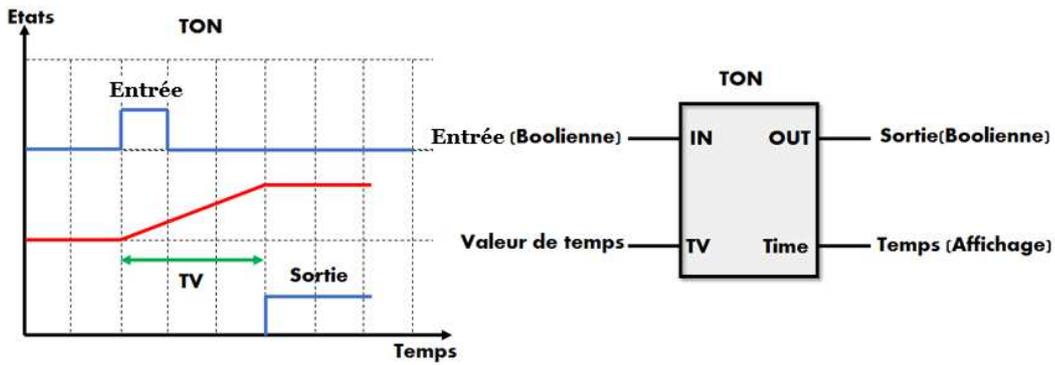


Figure VI-27. Temporisateur TON

La Figure (VI-28) montre une application de temporisateur de type TON dans trois langages de programmation Ladder, FBD et LIST. Dans cet exemple, si on appuie sur le bouton *start* dont l'adresse *I0.0*, le moteur *Q1.2* ne démarre qu'après cinq secondes.

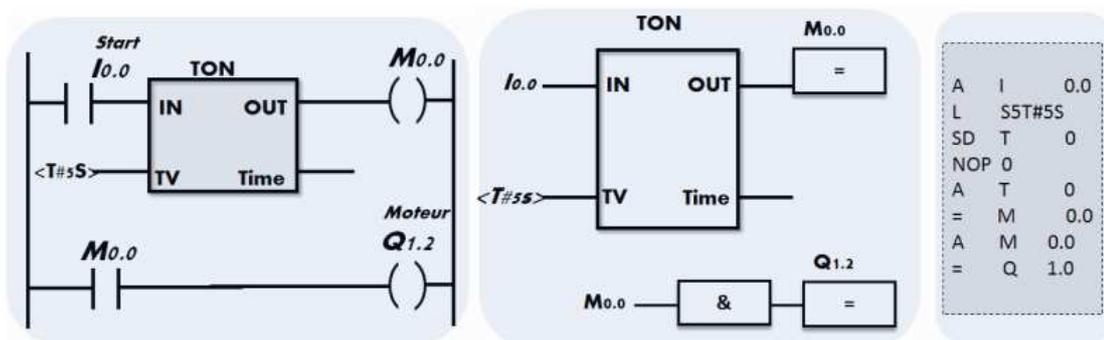


Figure VI-28. Application du temporisateur TON

VI.7.3.2. Temporisateur de repos (OFF-Delay Timer)

Un temporisateur de type TOFF (minuterie à retard d'excitation) fournit une action retardée. Si la ligne de commande n'a pas de continuité, le temporisateur commence à compter les intervalles de temps jusqu'à ce que la valeur de temps cumulé soit égale à la valeur pré-réglée TV (Fig.VI-29). Une fois l'entrée IN est désactivée le temporisateur TOF va déclencher la temporisation pendant le temps alloué TV, après ce temps, le TOF va désactiver la sortie OUT.

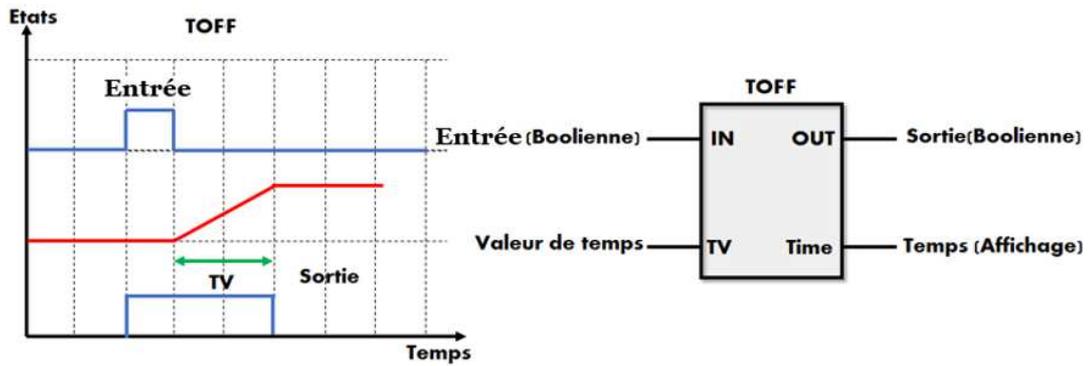


Figure VI-29. Temporisateur TOFF

Dans l'exemple de refroidissement d'un moteur pendant son état de marche, et même après son arrêt, dans ce cas-là, le processus de refroidissement va continuer à marcher après l'arrêt du moteur pendant cinq minutes. Le programme de fonctionnement sur l'API en trois langages est montré dans la Figure (VI-30).

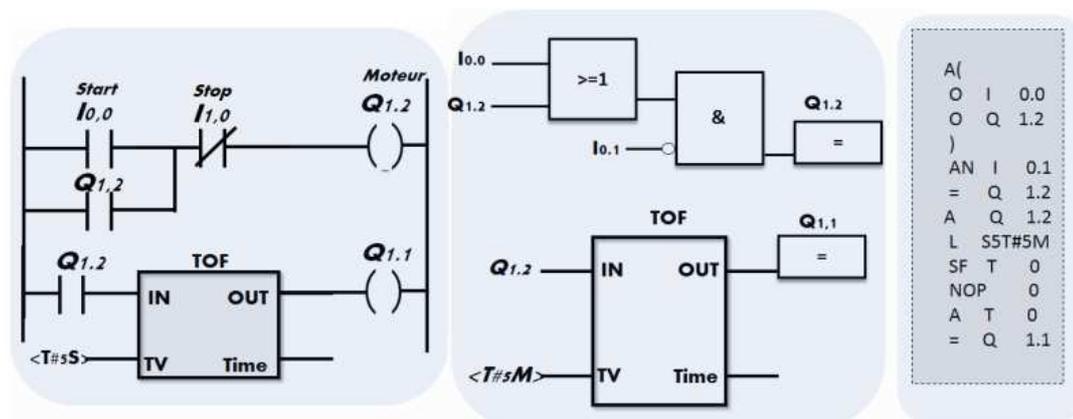


Figure VI-30. Application du temporisateur TOFF

VI.7.3.3. Temporisateur d'impulsion (Pulse Timer)

Ce type de temporisation est utilisé pour produire une sortie de durée fixe à partir d'une entrée. La Figure (VI-31) montre le diagramme d'échelle pour ce type de temporisateur qui donnera une sortie de out 1 pour une durée fixe prédéterminée de temps quand il y a une entrée a 1, après cette temporisation la sortie devient 1.

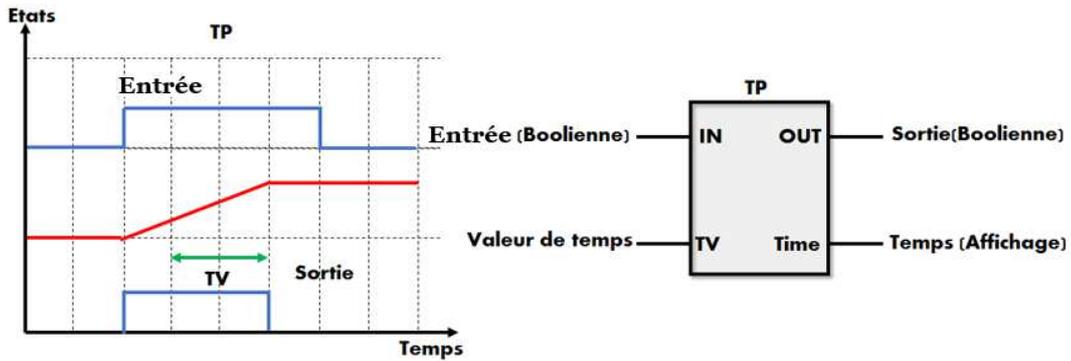


Figure VI-31. Temporisateur d'impulsion

Dans cet exemple, un moteur de lancement d'un turbo-générateur, démarre avec le lancement de la séquence par l'ordre (entrée $I_{0.0}$), ce moteur reste active $Q_{1.1}$ pendant trois minutes et relâche le turbo-générateur, après cette temporisation, même si l'entrée $I_{0.0}$ reste active, la sortie $Q_{1.1}$ reste désactivée.

Le programme de fonctionnement sur API en trois langages est montré dans la Figure (VI-32).

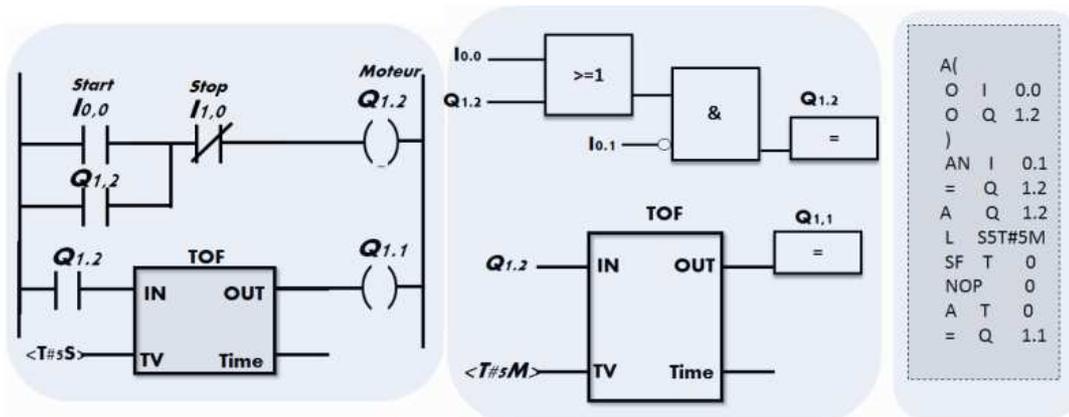


Figure VI-32. Application du temporisateur d'impulsion

VI.7.3.4. Temporisateur retentive (RTO)

Le type de temporisateur RTO est très similaire au type TON et TOFF, à l'exception du fait que la valeur cumulée (TV) est conservée même si les conditions des entrées sont à l'état zéro (Fig.VI-33).

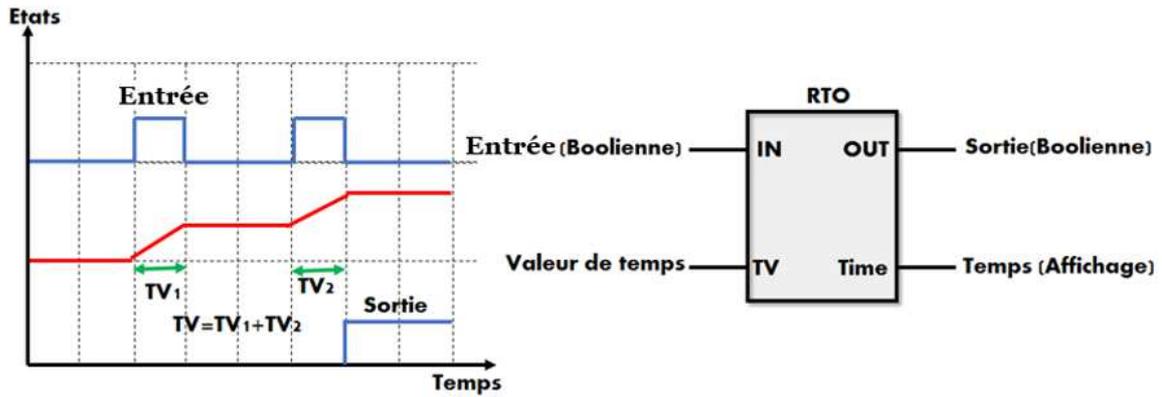


Figure VI-33. Temporisateur RTO

Dans certaines applications on a besoin de ne pas faire marcher un appareil ou une machine, plus de temps (cumulative), dans ce cas-là, on utilise le temporisateur de type RTO.

Le programme de fonctionnement sur API en trois langages est montré dans la Figure (VI-34).

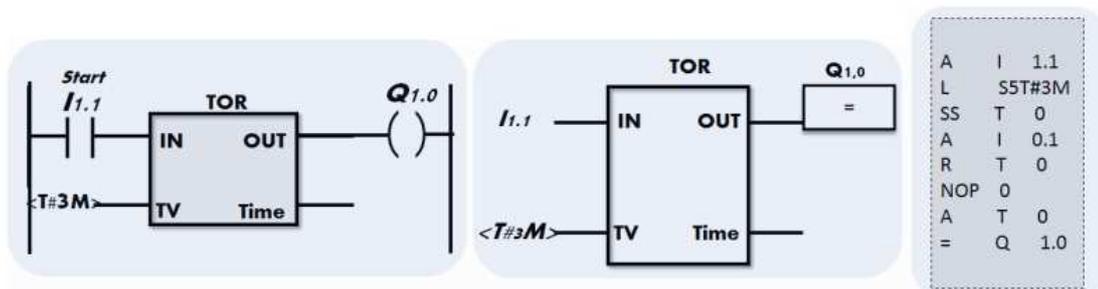


Figure VI-34. Application du temporisateur RTO

VI.7.4. Fonctions de Comptage

Les compteurs sont fournis en tant qu'éléments intégrés dans les automates et permettent de compter le nombre d'occurrences de signaux d'entrées. Par exemple pour compter le nombre des bouteilles produites ou bien combien de bouteilles sont passées par le convoyeur ou peut-être le nombre de personnes passant par une porte. Cette partie décrit comment ces compteurs peuvent être programmés. Généralement dans les API on trouve deux types des fonctions de comptage, le compteur et le décompteur.

VI.7.4.1. Compteurs (Up-counter CTU)

Le fonctionnement de ce type de compteur (CTU) est d'ajouter un nombre, chaque fois qu'un événement se produit, ce compteur incrémente à un, à la fin il active la sortie (Fig.VI-35).

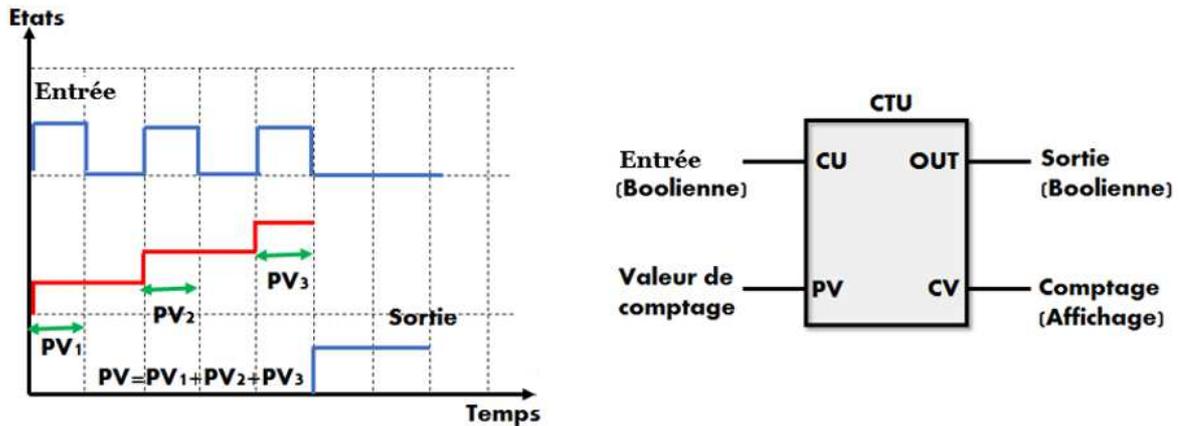


Figure VI-35. Fonction comptage CTU

La Figure (VI-36) représente le programme de comptage du nombre des bouteilles produites d'une chaîne de production, à la sortie de la chaîne il y a un capteur de détection de bouteilles (I0.0), une fois le nombre atteint six, le compteur donne l'ordre pour emballer ces six bouteilles (Q1.0).

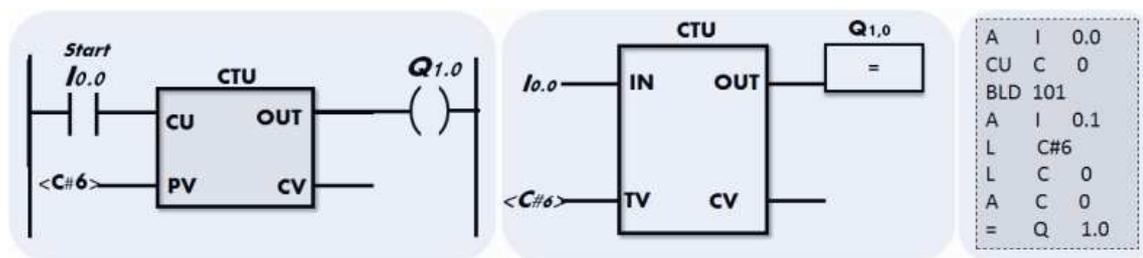


Figure VI-36. Application du compteur CTU

VI.7.4.2. Décompteurs (Up-counter CTD)

Le fonctionnement du décompteur (CTD) est de diminuer un nombre, chaque fois qu'un événement se produit, ce décompteur décrémente à un, à la fin il active la sortie (Fig.VI-37).

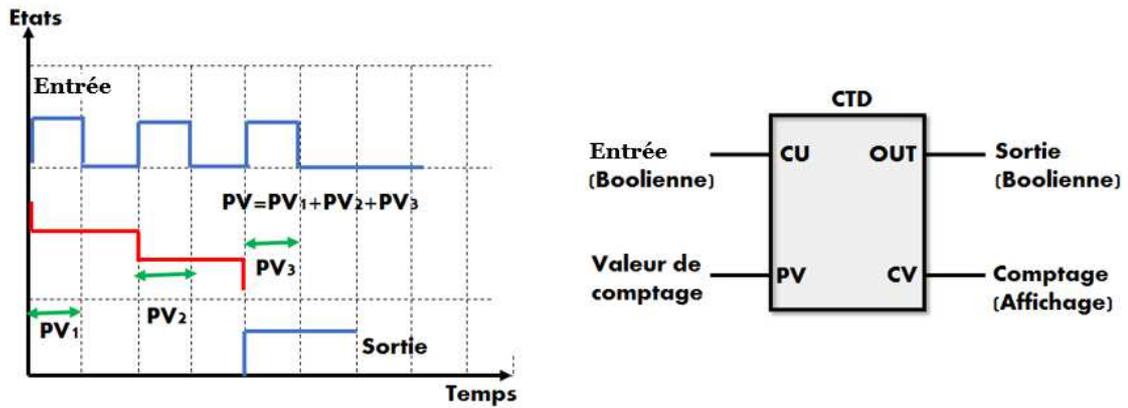


Figure VI-37. Fonction comptage CTD

Dans certaines applications on a besoin de répéter une séquence plusieurs fois, par la suite il continu la séquence, dans ce cas-là on exploite le fonctionnement du décompteur CTD. La Figure (VI-38) montre un tel exemple où la séquence est répétée cinq fois (I0.0), à la fin le décompteur active la sortie qui représente la réceptivité pour la prochaine étape.

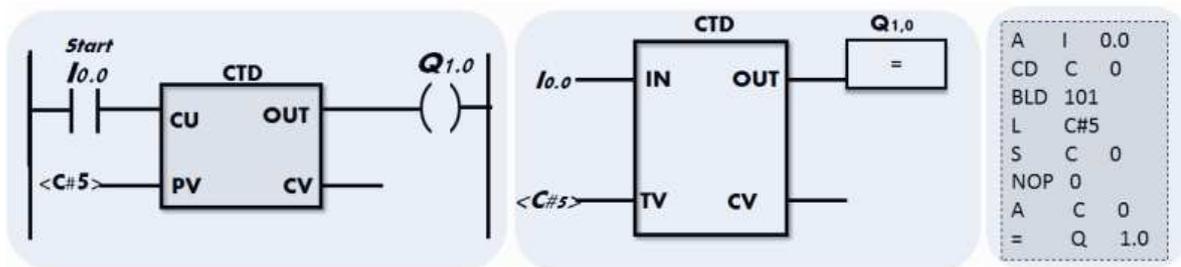


Figure VI-38. Application de décompteur CTD

VI.7.5. Fonctions de Régulation

La fonction de régulation est une action indispensable dans les automates programmables afin de contrôler les processus industriels, par exemple la régulation de la température d’une chaudière, la régulation de niveau dans un bac de stockage ... etc.

Généralement dans les APIs, il existe deux types de régulateurs, des régulateurs TOR et des régulateurs PID.

VI.7.5.1. Régulation TOR (ON-OFF)

Dans ce type de régulation, on peut distinguer deux autres types, le type de *régulation ON-OFF*, Figure (VI-39.a), et le type *ON-OFF à hystérésis* Figure (VI-39.b), dans les deux cas le contrôleur est essentiellement considéré comme un interrupteur qui fournit un signal de sortie marche/arrêt selon la valeur de l'erreur.

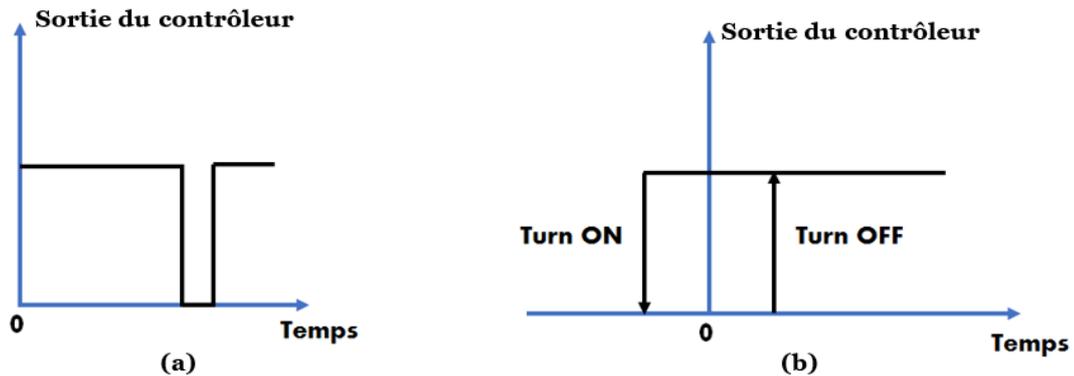


Figure VI-39. Régulation ON-OFF

La structure de ces types de contrôleurs (la régulation ON-OFF) est assez simple du point de vue programmation sur l'API, tout simplement il suffit d'utiliser des comparateurs afin de réguler le paramètre physique. Dans les exemples suivants on veut contrôler la température d'un four électrique industriel, à une consigne de 500°C, par les deux types de contrôleurs ON-OFF (Fig.VI-40) et ON-OFF à hystérésis (Fig.VI-41).

La Figure (VI-40) montre le programme de régulateur ON-OFF.

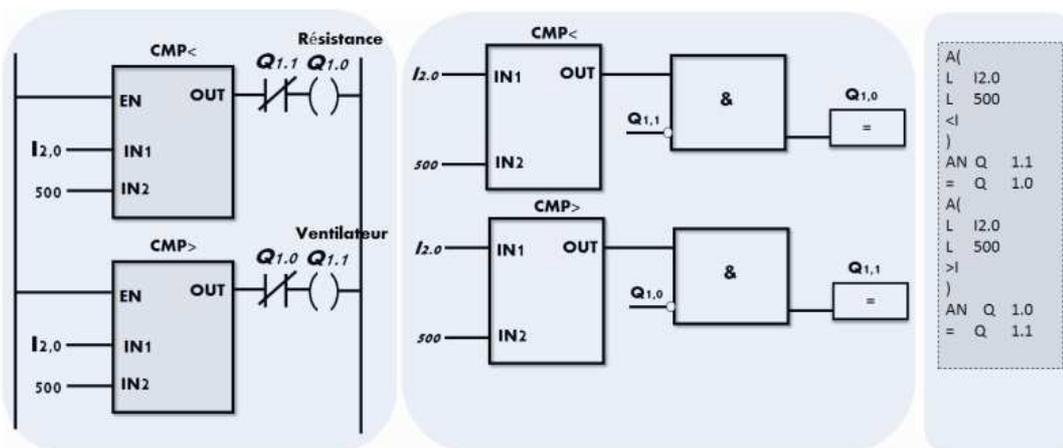


Figure VI-40. Application régulation ON-OFF

Le rôle du premier comparateur est de comparer la valeur mesurée de la température ($I_{2.0}$) avec la référence de 500°C , si elle est inférieure de cette référence le comparateur va activer la sortie $Q_{1.0}$, qui représente une résistance chauffante. Si la température mesurée est supérieure par rapport à la référence, le deuxième comparateur va activer la sortie $Q_{1.1}$ qui représente la ventilation. On note que les deux comparateurs ne fonctionnent pas au même temps.

Afin de résoudre le problème du premier régulateur le problème du basculement successif, on essaye de créer une hystérésis de 10°C pour éviter ce problème.

La Figure (VI-41) montre le programme de régulateur ON-OFF à hystérésis.

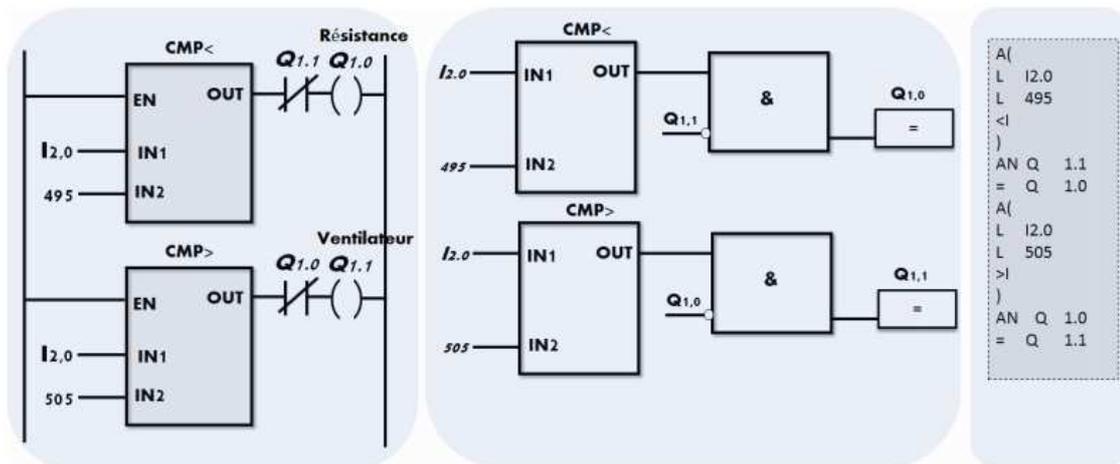


Figure VI-41. Application régulation ON-OFF à hystérésis

VI.7.5.2. Régulation PID

Les automates programmables sont capables d'effectuer un contrôle analogique à l'aide de l'algorithme PID à travers l'utilisation des blocs fonctionnels basés sur les actions, dérivé intégral et proportionnel (PID). L'utilisateur spécifie certains paramètres associés à l'algorithme pour contrôler le processus correctement. La Figure (VI-42) illustre un bloc PID typique.

Avec :

- ☞ **EN** : Permet d'activer ou désactiver le régulateur PID ;
- ☞ **AUTO** : Permet de sélectionner le mode automatique ou le mode manuel du régulateur ;

- ☞ **PV** : La mesure du processus ;
- ☞ **SP** : La consigne (référence) ;
- ☞ **Kp** : L'action proportionnel ;
- ☞ **Ti** : L'action intégral ;
- ☞ **Td** : L'action dérivée ;
- ☞ **CYCLE** : Permet de définir le fonctionnement cyclique du régulateur par rapport au programme principal ;
- ☞ **OUT** : L'action de correction du régulateur PID.

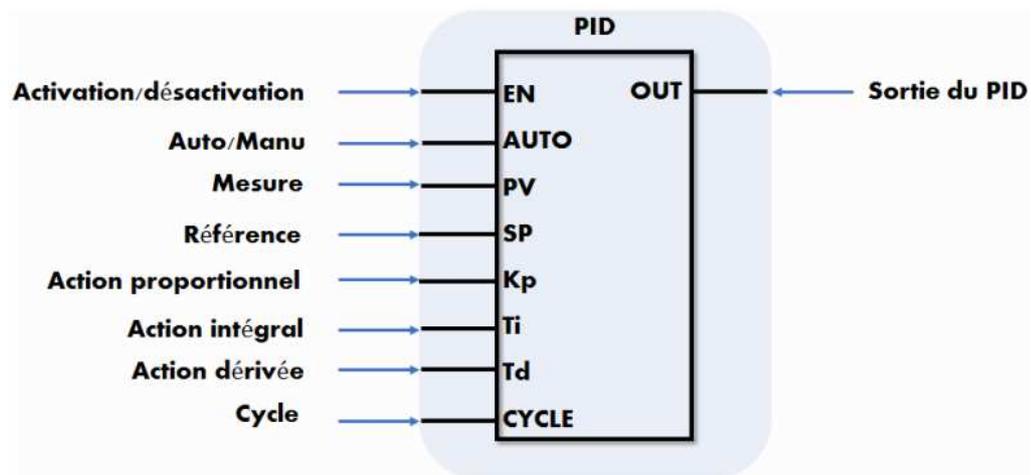


Figure VI-42. Régulateur PID

Le but de cet exemple est d'implémenter un régulateur de type PID dans un API, dans ce cas-là, la Figure (VI-43) présente le schéma de régulation du niveau d'un bac de stockage.

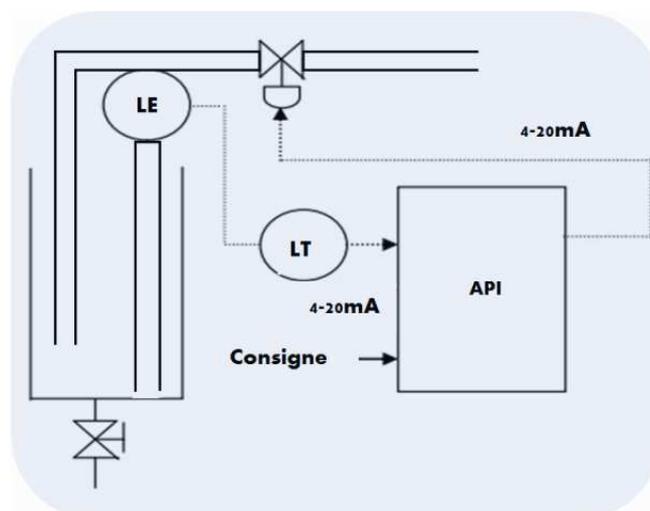


Figure VI-43. Régulation du niveau par l'API

Le programme correspondant à l'exemple de la Figure (VI-43) est montré dans la Figure (VI-44).

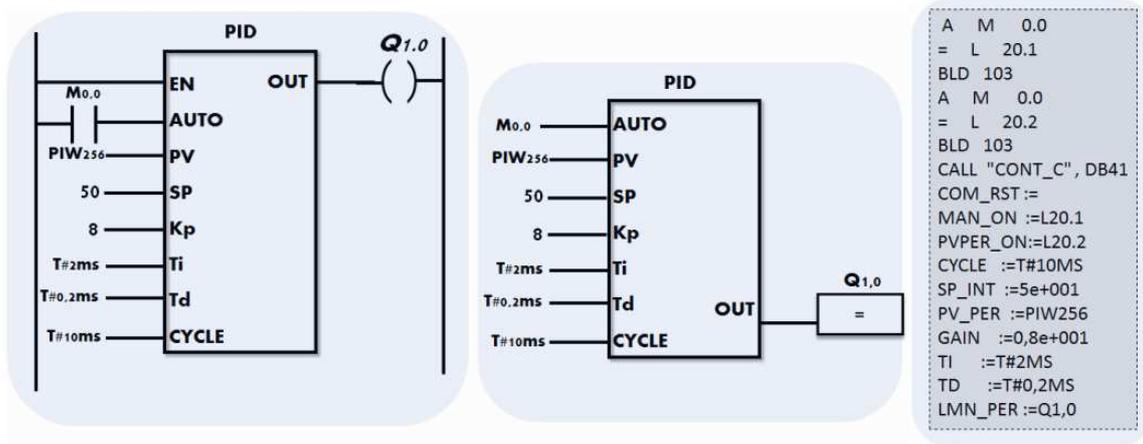


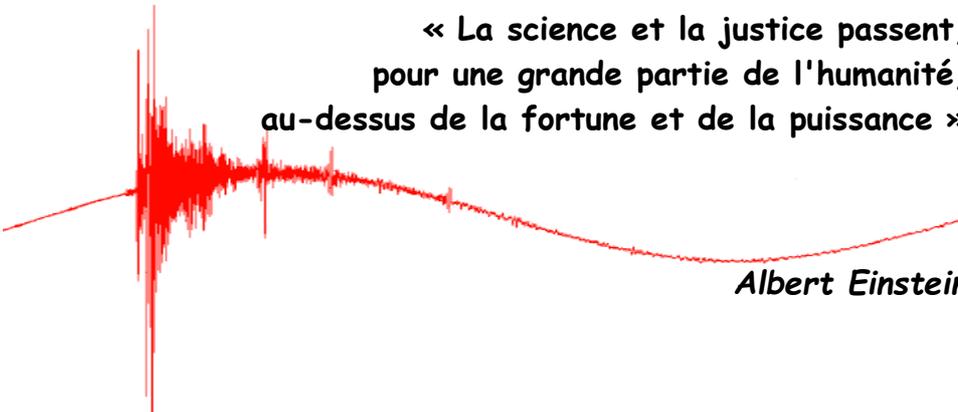
Figure VI-44. Programmation du régulateur PID

VI.8. CONCLUSION

Dans ce chapitre j'ai présenté les langages de programmation de l'API à savoir IL, FBD et LADDER. Ainsi que les principes du Ladder. J'ai montré l'adressage des entrées et des sorties des API. Dans cette partie j'ai présenté quelques applications le plus utilisé dans l'industrie et chaque application est illustré avec des exemples en trois langages : IL, FBD et LADDER.

CHAPITRE VII

TRAVAUX DIRIGÉS



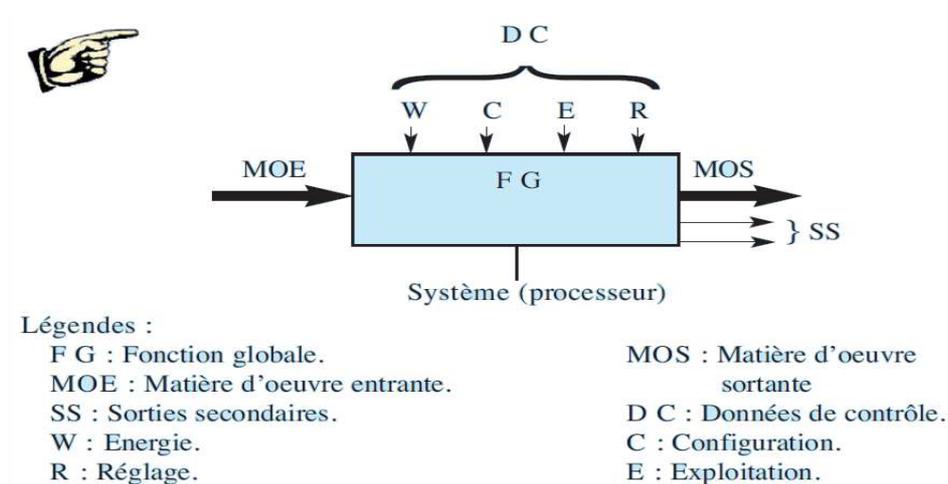
« La science et la justice passent,
pour une grande partie de l'humanité,
au-dessus de la fortune et de la puissance »

Albert Einstein

TRAVAUX DIRIGÉS N° 1

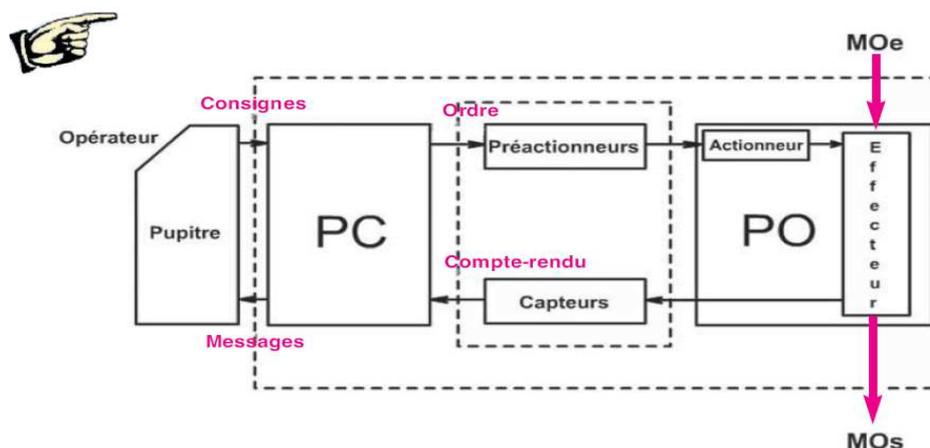
Modélisation :

Modéliser un système technique revient à lui donner une représentation graphique qui décrit le système et qui met en évidence ces caractéristiques :



Convention :

On représente la structure d'un système technique automatisé par le schéma suivant :



Un system automatisé est généralement constitue par :

- ⊙ Une partie commande (PC).
- ⊙ Une partie opérative (PO).
- ⊙ Des éléments d'interfaces qui relient la PC à la PO.
- ⊙ Un pupitre permettant le dialogue entre l'opérateur et le système automatisé.

Activité 1 : SYSTÈME « BARRIÈRE AUTOMATIQUE DE PARKING »

- ▣ Observer la barrière automatique de parking à l'arrêt.
- ▣ Mettre le système en fonctionnement.
- ▣ Observer le système en fonctionnement.
- ▣ Définir la frontière d'étude de ce système :

Voiture en attente, voiture passée, signalisation, énergie électrique, opérateur, barrière automatique, Micro-ordinateur.

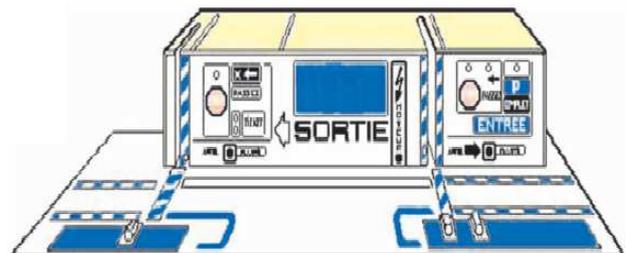
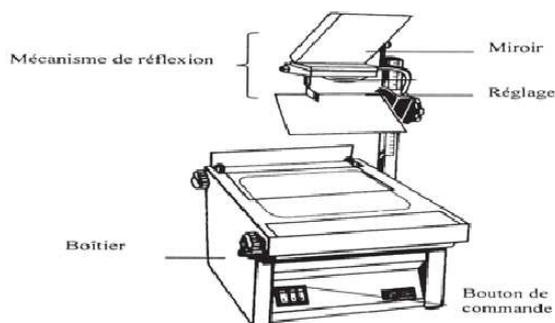
- ▣ Donner les matières d'œuvre entrante (MOE) et les matières d'œuvre sortante (MOS).
- ▣ Indiquer la nature de la matière d'œuvre (matérielle, énergétique, informationnelle).
- ▣ Préciser la valeur ajoutée (VA) apportée par le système à la matière d'œuvre.
- ▣ Modéliser le système.

Activité 2 : SYSTÈME « RÉTROPROJECTEUR »

- ▣ Observer le rétroprojecteur à l'arrêt.
- ▣ Mettre le système en fonctionnement.
- ▣ Observer le système en fonctionnement.
- ▣ Définir la frontière d'étude de ce système :

Rétroprojecteur, énergie électrique, opérateur, barrière automatique, bruit et chaleur, information projetées, information sur transparent.

- ▣ Donner les matières d'œuvre entrante (MOE) et les matières d'œuvre sortante (MOS).
- ▣ Indiquer la nature de la matière d'œuvre (matérielle, énergétique, informationnelle).
- ▣ Préciser la valeur ajoutée (VA) apportée par le système à la matière d'œuvre.
- ▣ Modéliser le système.



TRAVAUX DIRIGÉS N° 2

Exercice N° 1 :

Soit le système de production décrit par la figure *Exercice N° 1*. En entrée, les produits sont mis en palette. Cette opération prend un temps tp . Ils subissent ensuite une transformation sur une machine M pendant un temps tn . Les produits sont ensuite enlevés des palettes. Cette opération prend un temps td . La machine M est susceptible de tomber en panne. Dans ce cas, les produits sont redirigés sur une machine de secours M' , dont le temps de cycle est ts . La réparation de la machine M prend un temps tr . Le nombre de palettes circulant dans le système à tout moment est N .

- ▣ Modéliser ce système de production à l'aide d'un RdP temporisé.

Exercice N° 2 :

Soit le RdP de la figure *Exercice N° 2*, représentant deux chaînes de production en parallèle, utilisant les mêmes machines. Les transitions t_{01} et t_{02} représentent la mise en palette des produits. Les transitions t_{11} et t_{12} représentent une première transformation sur la machine M_1 . Les transitions t_{21} et t_{22} représentent une seconde transformation sur la machine M_2 . Les transitions t_{31} et t_{32} représentent le déchargement des produits finis.

- 1) Calculer la matrice d'incidence U .
- 2) Déterminer les p-invariants minimaux. Interpréter.
- 3) Déterminer les t-invariants minimaux. Interpréter.

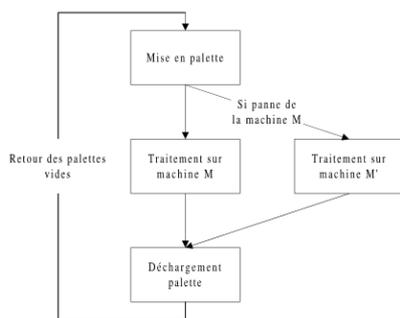
Exercice N° 3 :

Le RdP de la figure *Exercice N° 3* est le modèle d'un système dont la production est gérée par la méthode KANBAN. Les jetons situés dans la place P_i représentent les KANBAN libres. S'il y a des produits bruts à l'entrée du

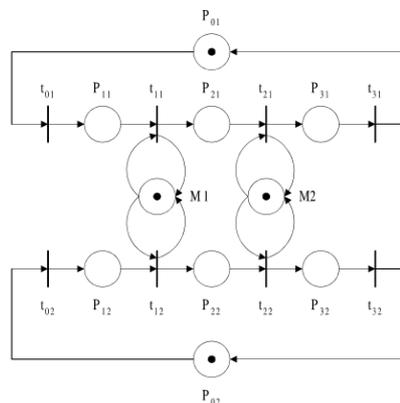
ystème, la transition t_1 est franchie. Un jeton apparaît alors dans la place P_2 , ce qui implique qu'un KANBAN libre est attaché à ce produit brut et que sa fabrication peut commencer. Après la fabrication de ce produit, représentée par le franchissement de la transition t_2 , un jeton apparaît dans la place P_3 . La transition t_3 est alors franchie : le produit sort du système et que le KANBAN qui lui est attaché est restitué.

Un contrôle de qualité est effectué à l'entrée du système. Si la qualité d'un produit brut n'est pas satisfaisante, la transition t_4 est franchie (rejet du produit).

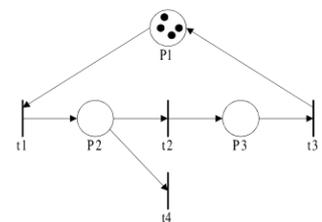
- 1) Déterminer l'ensemble des p-invariants et des t-invariants minimaux.
- 2) Vérifier les propriétés structurelles du RdP.
- 3) Mettre en évidence l'erreur de modélisation de ce système.
- 4) Proposer un modèle corrigeant cette erreur.



Exercice N° 1



Exercice N° 2

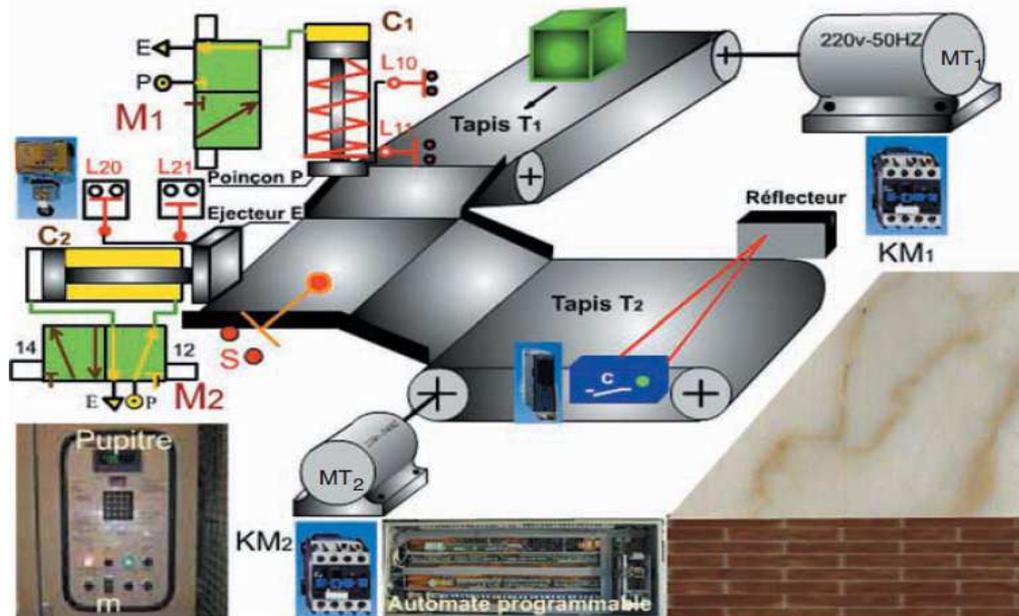


Exercice N° 3

TRAVAUX DIRIGÉS N° 3

« POSTE AUTOMATIQUE DE MARQUAGE DE SAVONS »

- 1) **Description du système** : Ce système est conçu pour le marquage de savon sur sa face supérieure par un poinçon.



Le système est constitué par :

- Deux vérins pneumatiques C1 et C2.
- Deux distributeurs pneumatiques M1 et M2.
- Quatre capteurs de position pneumatiques L10, L11, L20 et L21.
- Un capteur «s» de présence du savon.
- Un capteur «c» de passage du savon marqué.
- Un poinçon «P» et un éjecteur «E».
- Deux contacteurs KM1 et KM2.
- Deux moteurs électriques M1 et M2.
- Deux tapis T1 et T2.

- 2) **Fonctionnement du système** : Une action sur «m» provoque le cycle suivant :

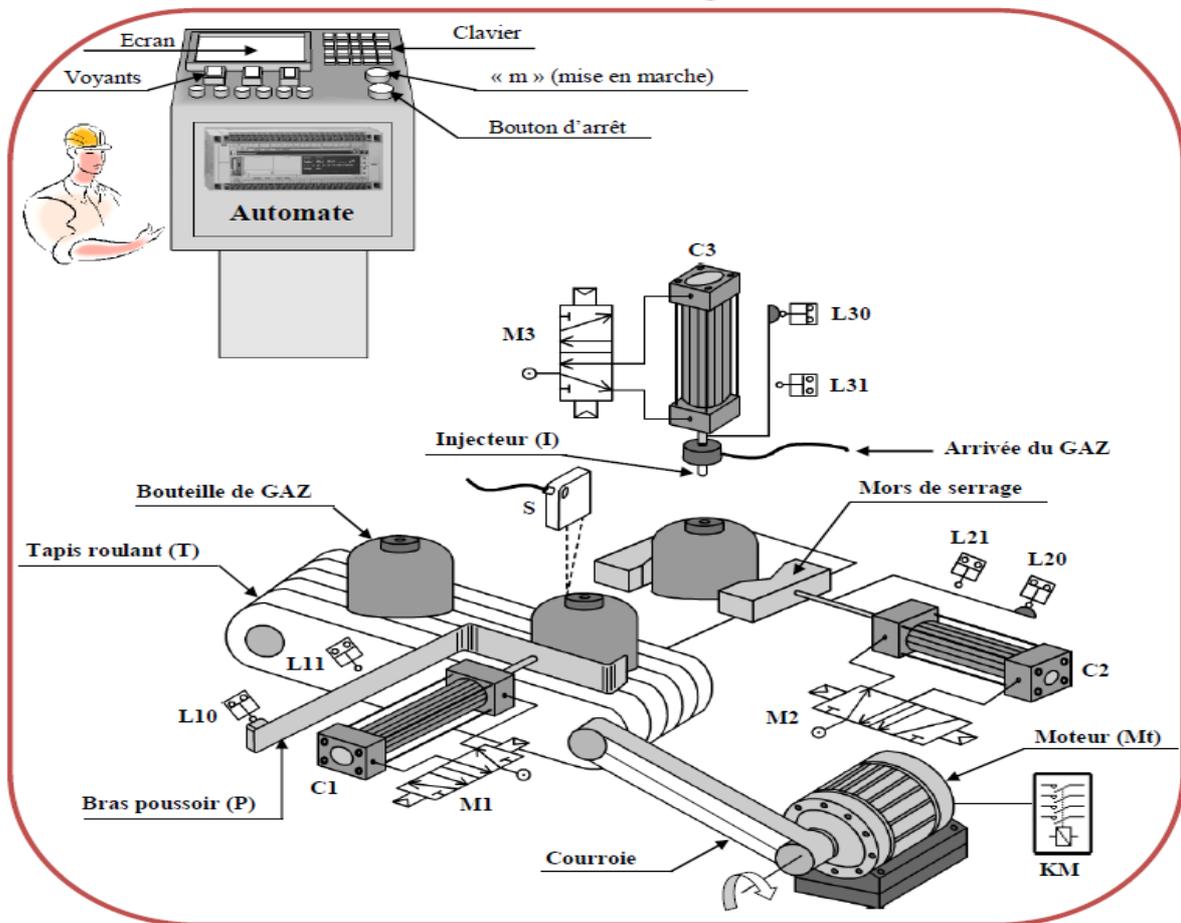
- Arrivée du savon sous le poinçon «P» par le tapis «T1».
- Marquage du savon par le poinçon «P».
- Evacuation du savon marqué vers le tapis roulant T2 par l'éjecteur E.

3) **Travail demandé** :

- a) Lire le dossier technique du système.
- b) Modéliser le système.
- c) Identifier les éléments du système.
- d) Compléter la structure d'un système technique automatisé.
- e) Tracer les GRAFCETs point de vue P.O, P.C et point de vue système.

TRAVAUX DIRIGÉS N° 4

« UNITE DE REMPLISSAGE AUTOMATIQUE DE BOUTEILLE DE GAZ »



I- FONCTIONNEMENT :

L'appui sur le bouton (m) de mise en marche provoque le départ du cycle de la façon suivante

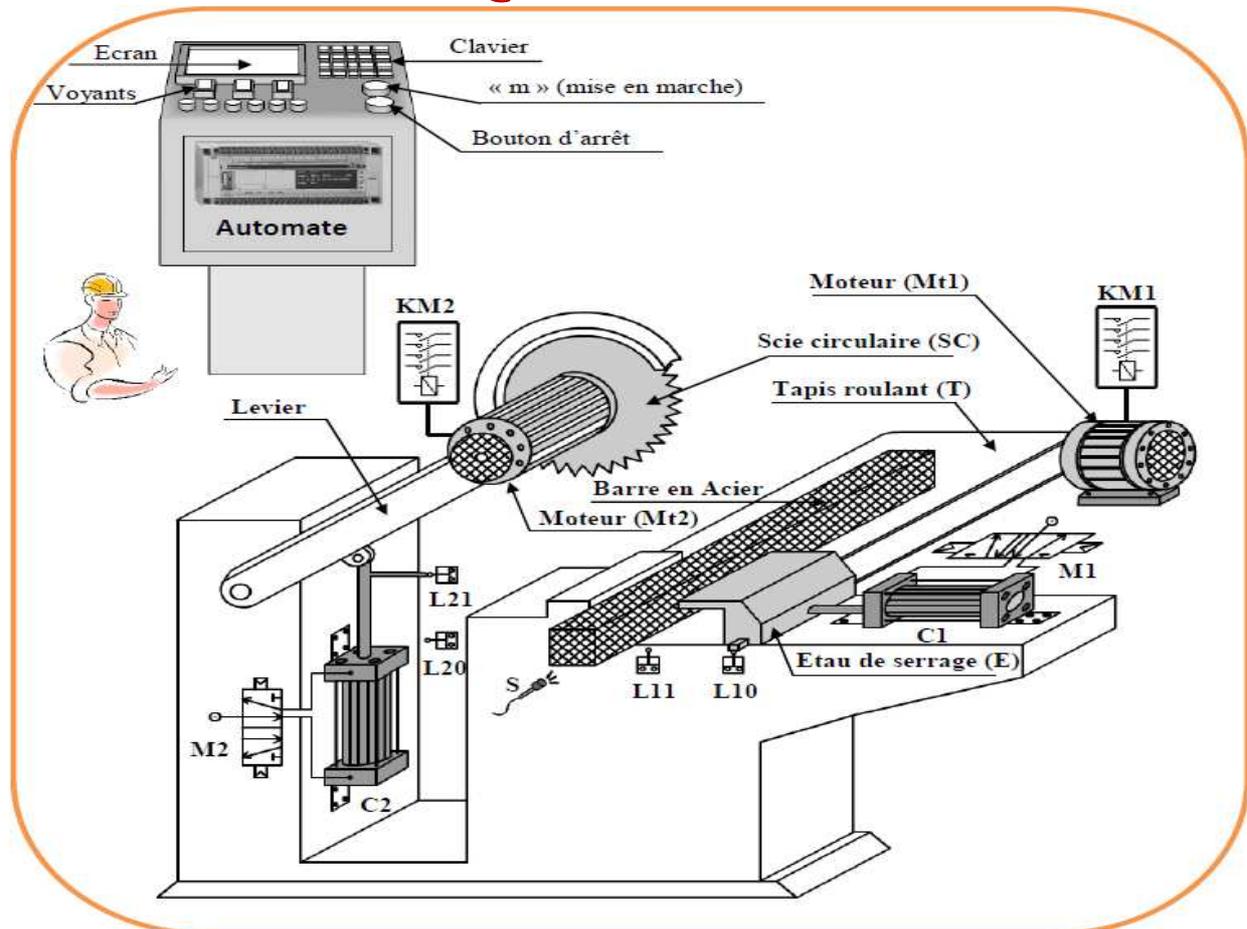
- L'amenée de la bouteille de gaz vide par le tapis (T) devant le bras poussoir (P).
- La poussée de la bouteille sous l'injecteur (I) de gaz par le bras poussoir (P).
- Le serrage de la bouteille réalisé grâce au vérin (C2).
- L'injection du gaz dans la bouteille par l'injecteur (I) donc la bouteille devient pleine.
- Desserrage de la bouteille.

II- TRAVAIL DEMANDE :

- 1- Identifier la partie commande de ce système (P.C).
- 2- Identifier les éléments de sa partie opérative (P.O).
- 3- Identifier les éléments d'interfaces de ce système (Pré-actionneurs, Capteurs).
- 4- Compléter la structure d'un système technique automatisé.
- 5- Tracer les GRAFCETs point de vue P.O, P.C et point de vue système.

TRAVAUX DIRIGÉS N° 5

« POSTE AUTOMATIQUE DE SCIAGE DE BARRE EN ACIER »



I- FONCTIONNEMENT :

L'appui sur le bouton (m) de mise en marche provoque le départ du cycle de la façon suivante

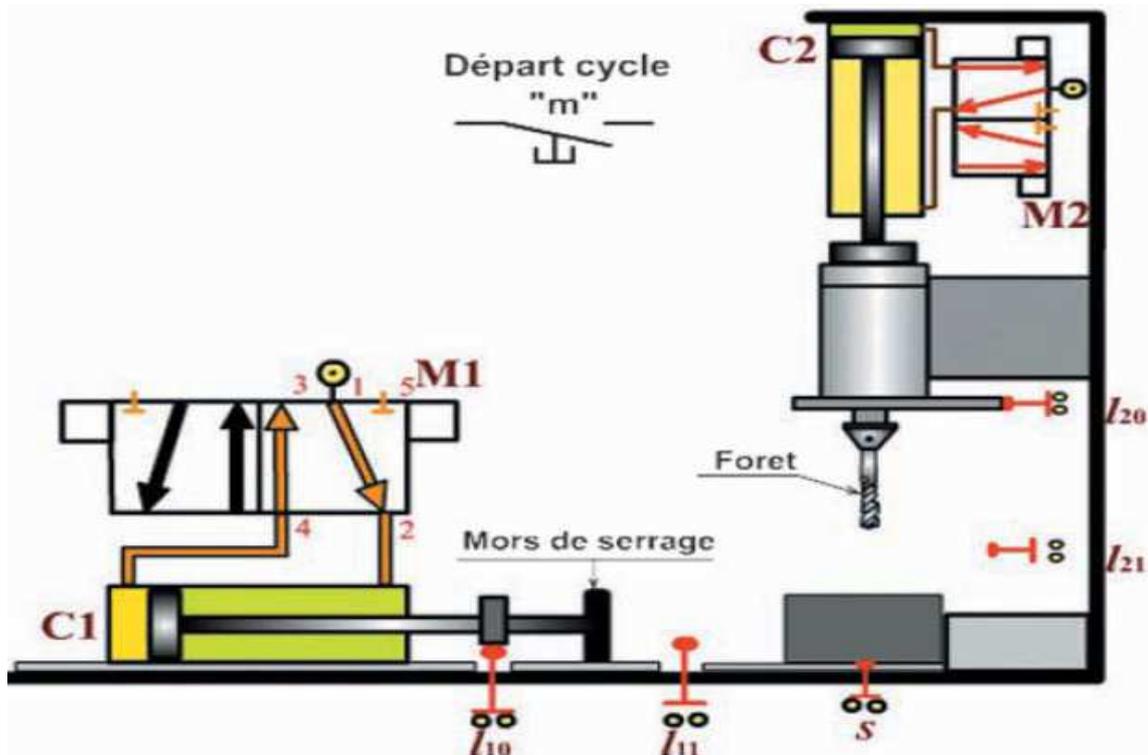
- Déplacement de la barre en Acier assuré par le tapis roulant (T) entraîné par le moteur (Mt1) jusqu'au capteur de proximité « S ».
- Serrage de la barre au moyen de l'étau de serrage (E) actionné par le vérin « C1 ».
- Descente lente du sous-système de découpage (moteur « Mt2 » + Scie « SC »).
- L'action du capteur L20 active l'opération de la remontée rapide du sous-système de découpage.
- Le desserrage de l'étau (E).

II- TRAVAIL DEMANDE :

- 1- Identifier la partie commande de ce système (P.C).
- 2- Identifier les éléments de sa partie opérative (P.O).
- 3- Identifier les éléments d'interfaces de ce système (Pré-actionneurs, Capteurs).
- 4- Compléter la structure d'un système technique automatisé.
- 5- Tracer les GRAFCETs point de vue P.O, P.C et point de vue système.

TRAVAUX DIRIGÉS N° 6

« Système technique : POSTE AUTOMATIQUE DE PERÇAGE »



I- FONCTIONNEMENT :

L'appui sur le bouton de départ cycle "m" provoque le serrage de la pièce puis son perçage. Une fois percée, la pièce est desserrée et le système revient à son état initial.

Ce système permet de percer des pièces.

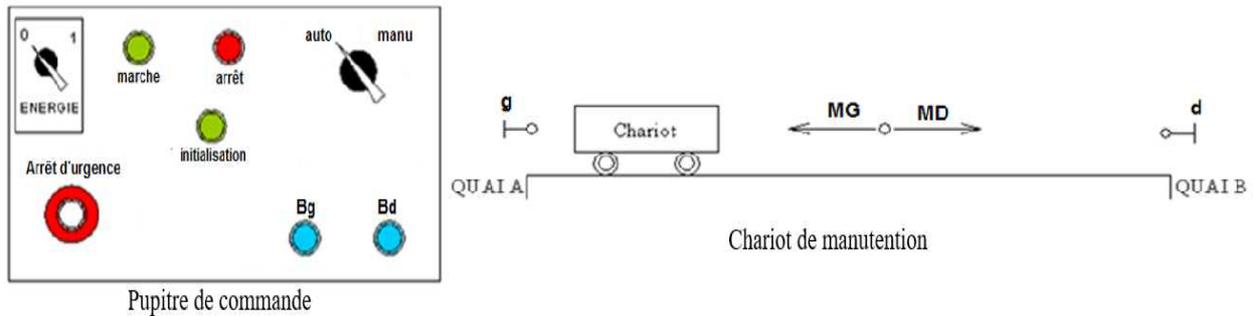
II- TRAVAIL DEMANDE :

- 1- Identifier la partie commande de ce système (P.C).
- 2- Identifier les éléments de sa partie opérative (P.O).
- 3- Identifier les éléments d'interfaces de ce système (Pré-actionneurs, Capteurs).
- 4- Compléter la structure d'un système technique automatisé.
- 5- Tracer les GRAFCETs point de vue P.O, P.C et point de vue système.

TRAVAUX DIRIGÉS N° 7

« POSTE DE MANUTENTION DE MARCHANDISES »

Fonctionnement :



Marche automatique : En choisissant le sélecteur **auto** : Comme conditions initiales, le chariot doit être à gauche. Ce dernier se déplace vers la droite au moyen de l'action **MD** suite à l'appui sur le bouton Marche. Il s'arrête automatiquement au **quai B** lorsqu'il actionne le capteur fin de course **d**. Il repart automatiquement au bout de **20s** vers le **quai A** par l'action **MG**. L'arrêt du chariot à gauche est obtenu lorsqu'il actionne le capteur fin de course **g**. Il attend **10s** et repart vers le **quai B** et ainsi de suite, et l'opération ne peut s'arrêter qu'en fin du cycle suite à l'appui sur le bouton Arrêt ou si on commute vers le mode manu.

Marche manuelle : En choisissant le sélecteur **manu** : Le chariot peut aller vers la droite par appui sur le bouton poussoir **Bd** (et s'il n'est pas à droite bien sûr) et peut retourner vers la gauche suite à l'appui sur le bouton poussoir **Bg** (s'il n'est pas à gauche). L'appui sur l'un de ces deux boutons doit être maintenu sinon le chariot s'arrête même s'il n'est pas parvenu à l'extrémité gauche ou droite.

Les modes de marche à considérer sont les suivants :

- Marche automatique
- Initialisation automatique de la partie opérative qui ramène le chariot vers la position extrême gauche. Cette initialisation doit être effectuée avant la marche automatique, après la marche manuelle et après traitement de l'arrêt d'urgence.
- Marche manuelle : le sélecteur de commande manuelle et le maintien de l'appui sur les boutons **Bd** ou **Bg** peut ramener le chariot vers la gauche ou la droite.
- Arrêt d'urgence : on ne prévoit pas la phase de diagnostic et/ou traitement de la défaillance.

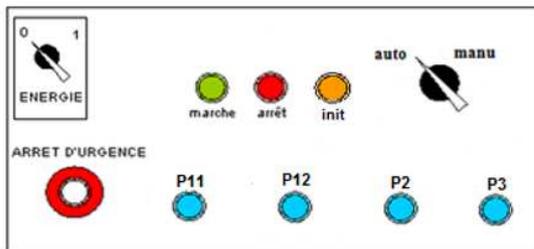
Travail demandé :

1. Compléter le guide GEMMA relatif au poste de manutention.
2. Tracer les GRAFCETs de Conduite GC, de production normale GPN, de marche manuelle GMM et de Sécurité GS.

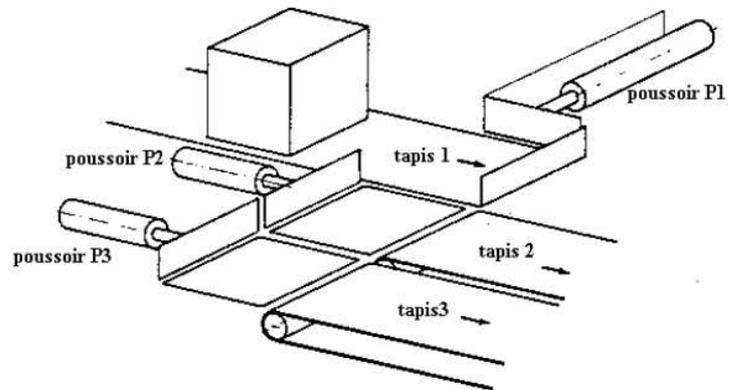
TRAVAUX DIRIGÉS N° 8

« POSTE DE MANUTENTION DEMMARCHANDISES »

Fonctionnement :



Pupitre de commande



Poste de tri de caisses

Un dispositif automatique destiné à trier des caisses de deux tailles différentes se compose d'un tapis amenant les caisses, de trois poussoirs à double effet et de deux tapis d'évacuation.

Le poussoir **P1** pousse les petites caisses devant le poussoir **P2** qui à son tour les transfère sur le tapis 2, alors que les grandes caisses sont poussées devant le poussoir **P3**, ce dernier les évacuant sur le tapis 3.

Pour effectuer la sélection des caisses, un dispositif de détection placé devant le poussoir **P1** permet de reconnaître sans ambiguïté le type de caisse qui se présente.

Les modes de marche à considérer sont les suivants :

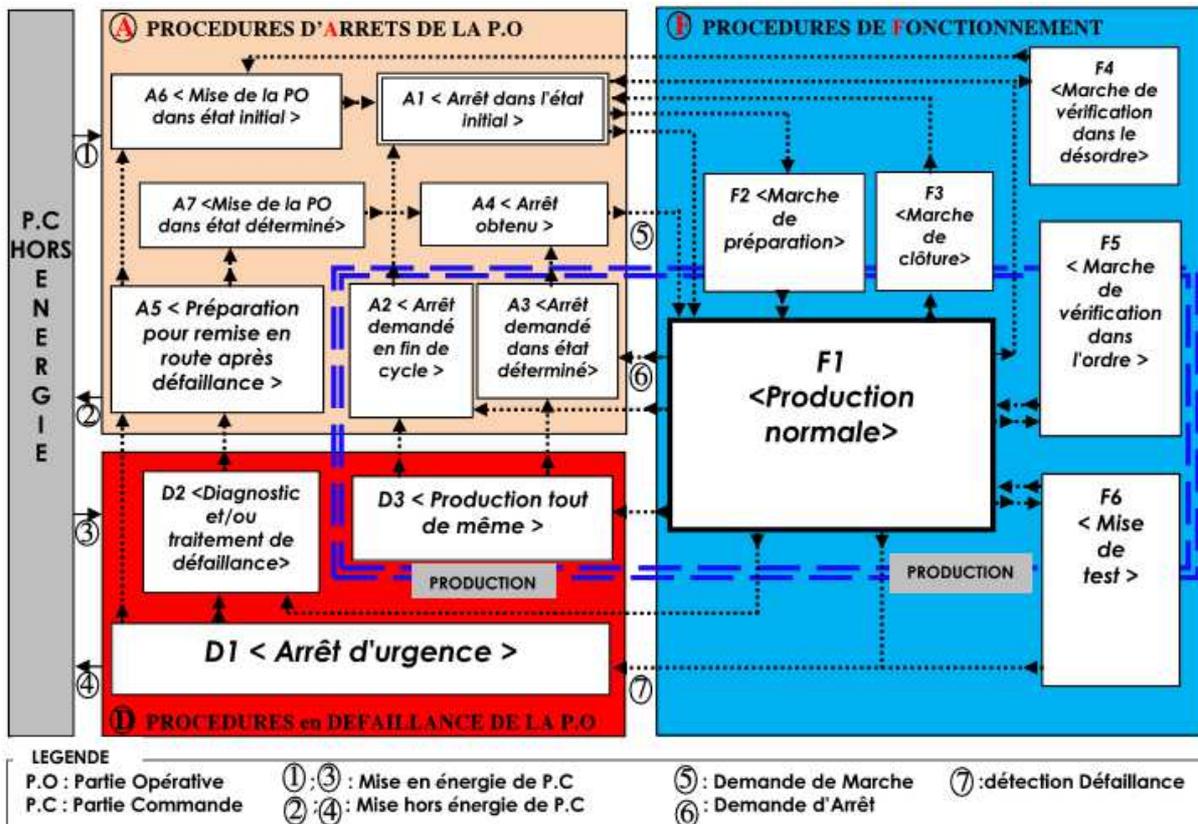
- marche automatique à l'aide du sélecteur **auto**. Le cycle débute suite à l'appui sur le bouton **marche**. Le système est conçu pour fonctionner jusqu'à l'appui sur le bouton **arrêt**, le système termine alors le cycle puis s'arrête.
- initialisation automatique de la partie opérative. Si un des poussoirs est en extension, le ramener à l'état initial.
- marche manuelle : le sélecteur en position **manu**, l'opérateur peut commander un poussoir en appuyant sur le bouton correspondant. Si le bouton est relâché, le poussoir retourne à la position initiale.
- traitement de l'arrêt d'urgence : toutes les actions en cours doivent être désactivées.

Travail demandé :

1. Compléter le guide GEMMA.
2. Réaliser le GRAFCET de Sécurité GS, le GRAFCET de Conduite GC, le GRAFCET de production normale GPN et le GRAFCET de Marche Manuelle GMM en se référant à la nomenclature et au pupitre de commande (sans prendre en considération les tapis roulants)

Table des mnémoniques de Système Technique : « Poste de tri de caisses » :

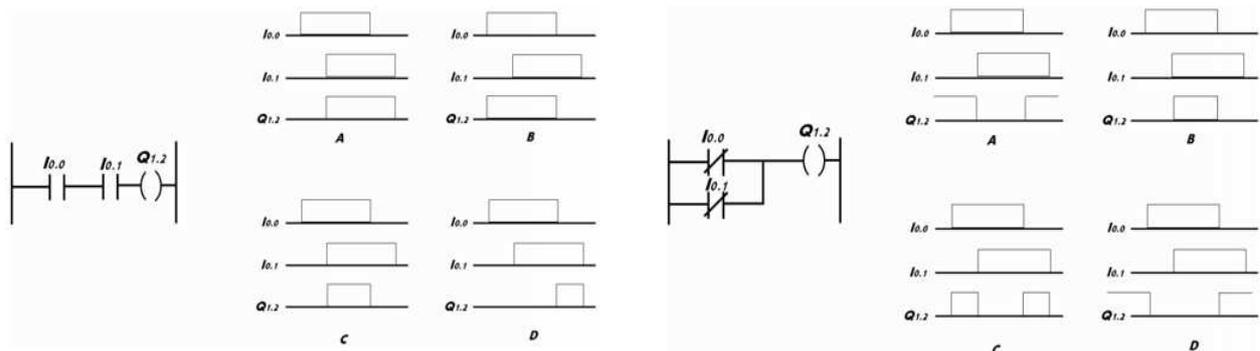
Symbole	Description
SP11	Sortie du poussoir P ₁ à la position intermédiaire
SP12	Sortie du poussoir P ₁ à la position maximale
RP1	Rentrée du poussoir P ₁
SP2	Sortie du poussoir P ₂
RP2	Rentrée du poussoir P ₂
SP3	Sortie du poussoir P ₃
RP3	Rentrée du poussoir P ₃
L10	Capteur de la position en rétraction du poussoir P ₁
L11	Capteur de position intermédiaire du poussoir P ₁
L12	Capteur de position en extension du poussoir P ₁
L20	Capteur de la position en rétraction du poussoir P ₂
L21	Capteur de position en extension du poussoir P ₂
L30	Capteur de position en rétraction du poussoir P ₃
L31	Capteur de position en extension du poussoir P ₃
Pc	Capteur détectant les petites caisses
Gc	Capteur détectant les grandes caisses



TRAVAUX DIRIGÉS N° 9

Exercice 1 :

Les figures ci-dessous montrent un programme Ladder, le lequel des diagrammes montrant les entrées et les signaux de sortie se produirait avec ce programme ?



Exercice 2 :

Décidez si chacune de ces instructions est Vrai (V) ou fausse (F). Les figures ci-dessous montrent un programme en langage LIST pour lequel il y a une sortie quand :

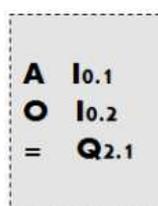
LIST 1 :

- (i) L'entrée I0.1 est activée mais I0.2 est désactivée.
- (ii) L'entrée I0.2 est activée mais I0.1 est désactivée.

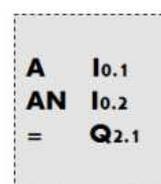
LIST 2 / LIST 3 :

- (i) L'entrée I0.1 est activée mais I0.2 est désactivée.
- (ii) L'entrée I0.2 et I0.1 sont activées.

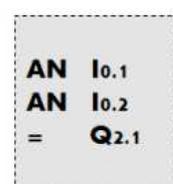
- A: (i)F (ii)F
- B: (i)V (ii)F
- C: (i)F (ii)V
- D: (i)V (ii)V



LIST 1



LIST 2



LIST 3

CHAPITRE VIII

TRAVAUX PRATIQUES

« La mécanique n'est qu'un schéma général ;
elle devient une théorie seulement
par l'indication explicite des lois de force,
comme l'a fait Newton avec tant de
succès pour la mécanique céleste »

Albert Einstein

TRAVAUX PRATIQUES N° 1

GRAFSET Allumage d'une lampe / Temporisation

Partie A : GRAFCET 2D Allumage d'une lampe



Cahier des charges :

On désire allumer une lampe à la première impulsion donnée par un bouton poussoir et l'éteindre à la deuxième. Les deux actions obtenues sont contradictoires et sont obtenues à partir de la même information donnée par l'opérateur.

Travail à réaliser :

1. Identifiez les variables d'entrées/sorties du système sous forme d'un tableau.
2. Tracez le grafset qui décrit le fonctionnement du système.
3. Simulez le grafset sous le logiciel Automgen.
4. A l'aide du module IRIS 2D, créez un pupitre de visualisation comprenant :
 - Un bouton poussoir Marche ou bien un interrupteur ;

Parie B : GRAFCET 2D Temporisation



Cahier des charges :

On désire allumer deux lampes successivement :

Le premier reste allumé pendant 15 secondes et la deuxième 10 secondes.

NB : la commande d'allumage pour les deux lampes s'effectue par un bouton poussoir.

Travail à réaliser :

1. Identifiez les variables d'entrées/sorties du système sous forme d'un tableau.
2. Tracez le grafcet qui décrit le fonctionnement du système.
3. Simulez le grafcet sous le logiciel Automgen.
4. A l'aide du module IRIS 2D, créez un pupitre de visualisation comprenant :
 - Un bouton poussoir.

TRAVAUX PRATIQUES N° 2

Utilisation des Fonctions OU (OR) / ET (AND)

Partie A : Utilisation de la Fonction OU (OR)

Cahier des charges : *ice cream machine*

On désire réaliser un grafcet pour une machine d'*Ice CREAM*, Sachant que la machine ne prépare qu'une seule couleur à la fois :

- *ice cream* en couleur rouge "**R**" ;
- *ice cream* en couleur bleu "**B**" ;
- *ice cream* en couleur vert "**V**".

Travail à réaliser :

1. Identifiez les variables d'entrées/sorties du système sous forme d'un tableau.
2. Tracez le grafcet qui décrit le fonctionnement du système.
3. Simulez le grafcet sous le logiciel Automgen.
4. A l'aide du module IRIS 2D, créez un pupitre de visualisation comprenant :
 - Un bouton poussoir Marche "**M**" ;
 - Un bouton poussoir pour l'action B "**BB**" ;
 - Un bouton poussoir pour l'action B "**BR**" ;
 - Un bouton poussoir pour l'action B "**BV**".

Parie B : Utilisation de la Fonction ET (AND)

Cahier des charges :

On désire réalise un grafcet pour un système de trois moteurs : Moteur A, B et C. Sachant que les trois moteurs fonctionne simultanément et que la durée de fonctionnement de chaque moteur :

- Moteur A : 5 Secondes ;
- Moteur B : 10 Secondes ;
- Moteur C : 15 Secondes.

Travail à réaliser :

1. Identifiez les variables d'entrées/sorties du système sous forme d'un tableau.
2. Tracez le grafcet qui décrit le fonctionnement du système.
3. Simulez le grafcet sous le logiciel Automgen.
4. A l'aide du module IRIS 2D, créez un pupitre de visualisation comprenant :
 - Un bouton poussoir Marche "M".

TRAVAUX PRATIQUES N° 3

Réalisation d'un Compteur

Cahier des charges :

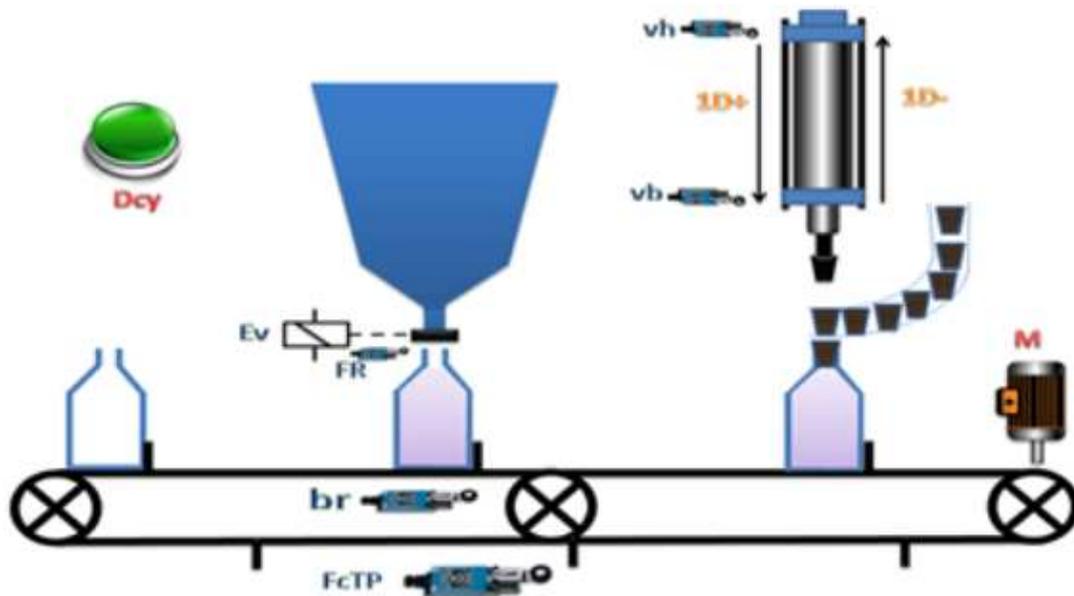
On désire réaliser un grafcet pour l'allumage de deux lampes A et B à l'impulsion donnée par un interrupteur, **BA** pour lampe A et **BB** pour lampe B. Sachant qu'il faut ajouter un compteur pour les deux lampes pour compter les nombres de fois que lampe A ou B s'allume.

Travail à réaliser :

1. Identifiez les variables d'entrées/sorties du système sous forme d'un tableau.
2. Tracez le grafcet qui décrit le fonctionnement du système.
3. Simulez le grafcet sous le logiciel Automgen.
4. A l'aide du module IRIS 2D, créez un pupitre de visualisation comprenant :
 - un interrupteur **BB** et un afficheur pour lampe B ;
 - un interrupteur **BA** et un afficheur pour lampe A.

TRAVAUX PRATIQUES N° 4

Modes : Urgent, Arrêt et Marche



Cahier des charges :

Il s'agit d'un système utilisé dans les usines de production des boissons liquides. Il décrit une partie du processus assurant les fonctions de remplissage et de bouchage des bouteilles.

Le système est réalisé autour de :

- Un tapis roulant permettant le déplacement des bouteilles.
- Un poste de remplissage P_1 commandé par l'électrovanne EV .
- Un poste de bouchage P_2 commandé par un vérin presseur iD à double effet.
- Le déclenchement de la chaîne d'embouteillage se fait par action sur l'interrupteur Dcy .
- Le moteur "Avance Tapis : M " tourne d'un pas jusqu'à l'action du capteur "Tapis en position : $FcTP$ ". Une bouteille est alors présente à chacun des postes P_1 (détecter par pbv) et P_2 (détecter par pbp).

- Les opérations de remplissage et de bouchage s'effectueront simultanément sur les deux bouteilles :

Le remplissage se fera en deux étapes :

- Ouverture de l'électrovanne *EV* ;
- Fermeture de l'*EV* après le remplissage de la bouteille. Le capteur "Bouteille remplie : *br*" permettra de contrôler le niveau de remplissage des bouteilles.

Le bouchage se fera en deux étapes :

- Descente du vérin presseur *ID+* ;
- Remonte du vérin *ID-* après l'enfoncement du bouchon.

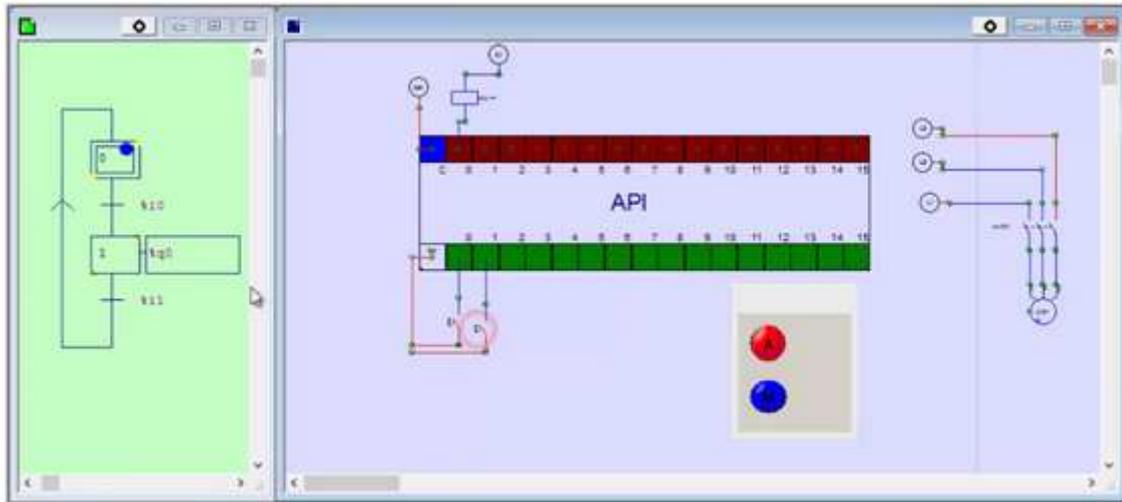
Il est à noter que le cycle ne recommencera que si les deux opérations de remplissage et de bouchage sont achevées.

Travail à réaliser :

1. Identifiez les variables d'entrées/sorties du système sous forme d'un tableau.
2. Tracez le grafcet qui décrit le fonctionnement du système.
3. Simulez le grafcet sous le logiciel Automgen.
4. A l'aide du module IRIS 2D, créez un pupitre de visualisation comprenant :
 - Un bouton poussoir *Marche* et *Arrêt*.
 - Un bouton poussoir arrêt d'*urgence*.
 - Un voyant signalant l'état du moteur (rotation).
 - Modifier le GRAFCET précédent afin de gérer l'arrêt d'urgence (AU).

TRAVAUX PRATIQUES N° 5

Automatisation d'un Démarrage direct d'un moteur triphasé



Cahier des charges :

On veut réaliser une simulation en deux dimensions, pour un démarrage direct d'un moteur triphasé au moyen de logiciel Automgen, à l'aide d'un :

- Automate programmable industriel (API) ;
- Moteur triphasé ;
- Contact de puissance ;
- Bouton marche et arrêt.

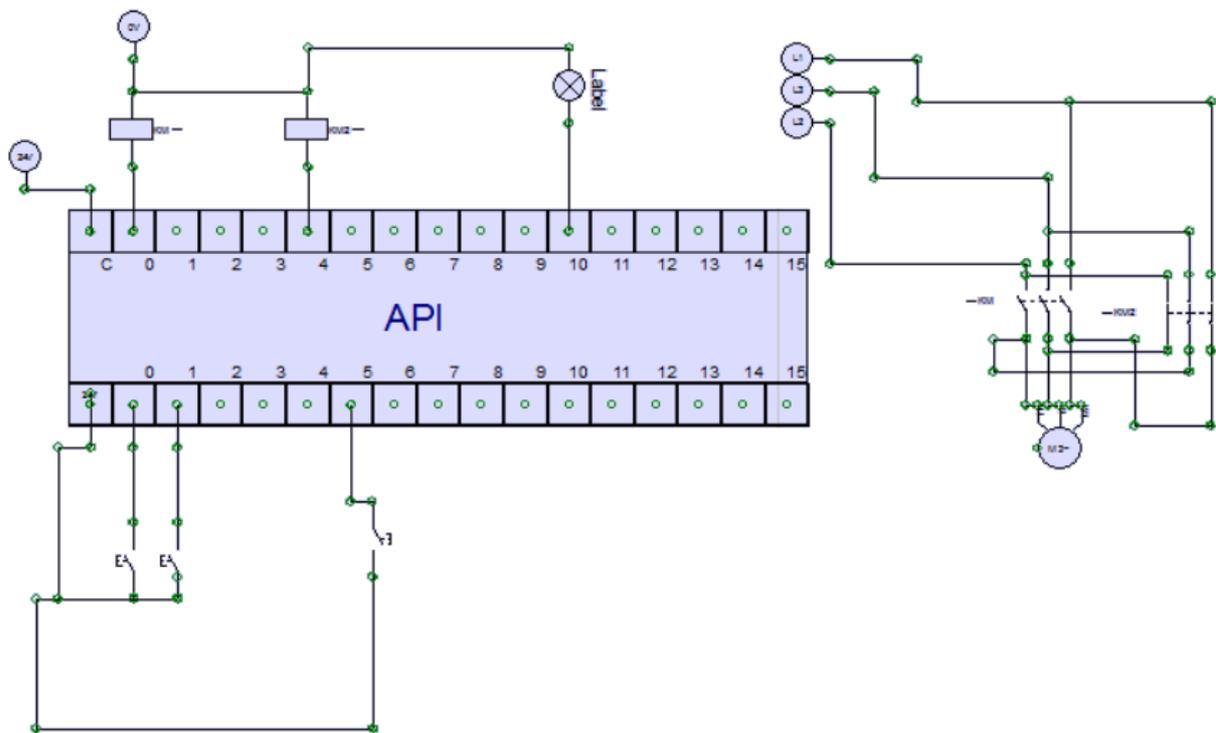
Remarque : Vous trouverez ces matériels (API, Moteur) dans l'atelier Automsim dans Logicielle Automgen.

Travail à réaliser :

1. Identifiez les variables d'entrées/sorties du système sous forme d'un tableau.
2. Tracez le grafcet qui décrit le fonctionnement du système.
3. Simulez le grafcet sous le logiciel Automgen.
4. A l'aide du module IRIS 2D, créez un pupitre de visualisation comprenant :
 - Un bouton poussoir Marche et Arrêt.

TRAVAUX PRATIQUES N° 6

Automatisation d'un Démarrage Direct à Deux Sens de Rotation d'un Moteur Triphasé



Cahier des charges :

On veut réaliser une simulation en deux dimensions, pour un démarrage direct à deux sens de rotation d'un moteur triphasé au moyen de logiciel Automgen, À l'aide d'un :

- Automate programmable industriel (API) ;
- Moteur triphasé ;
- Deux contact de puissance KM1 et KM2 ;
- Bouton marche 1 et 2 et arrêt.

Remarque : Vous trouverez ces matériels (API, Moteur....) dans l'atelier Automsim dans Logicielle Automgen.

Travail à réaliser :

1. Identifiez les variables d'entrées/sorties du système sous forme d'un tableau.
2. Tracez le grafcet qui décrit le fonctionnement du système.
3. Simulez le grafcet sous le logiciel Automgen.
4. A l'aide du module IRIS 2D, créez un pupitre de visualisation comprenant :
 - Un bouton poussoir Marche et Arrêt.

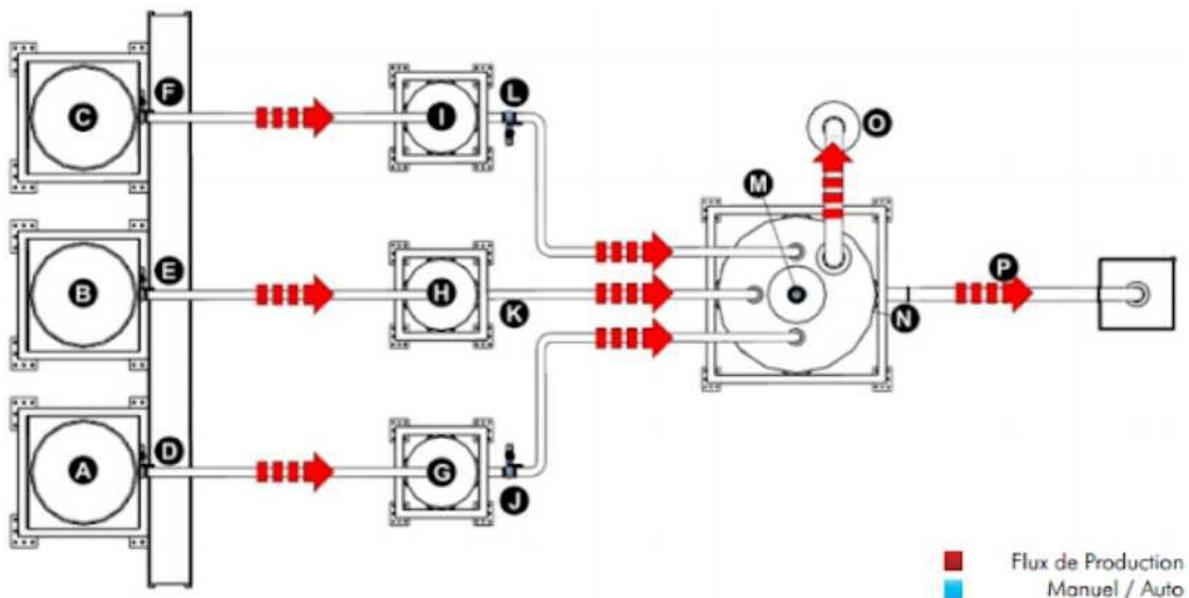
TRAVAUX PRATIQUES N° 7

Automatisation d'un Système de Production Industriel Mélangeur de Couleur



Cahier des charges :

Cette partie opérative simule un processus de mélange de peintures. L'objectif est de mélanger les trois couleurs primaires (*rouge*, *vert* et *bleu*) pour obtenir la couleur souhaitée.



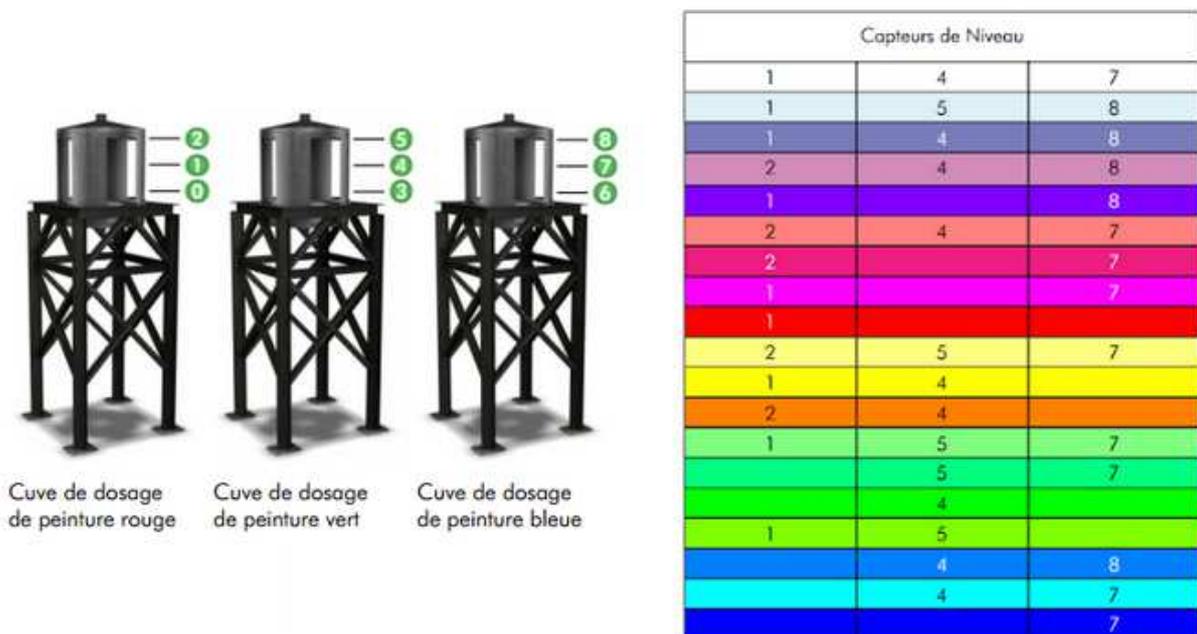
Ce système de mélange est composé de trois réservoirs de peinture, trois cuves de dosage et d'une cuve pour faire le mélange.

Les réservoirs de peinture (A, B et C) contiennent respectivement de la peinture rouge, verte et bleue. Les vannes (D, E, F) permettent de remplir les cuves de dosage (G, H, I) à partir des réservoirs. Chaque cuve de dosage a trois niveaux de mesure.

La peinture des cuves de dosage est envoyée à la cuve de mélange (M) via les vannes (J, K, L). Si le volume de peinture ainsi envoyé est plus important que la capacité de la cuve de mélange, le surplus passe par le tuyau de trop plein (O).

Le mélange doit durer un minimum de cinq secondes. La peinture obtenue est déchargée par la vanne (N) dans le tuyau de sortie (P).

Les différentes couleurs obtenues en fonction des capteurs de niveaux :



Note : Il faut 5 secondes pour obtenir un mélange homogène.

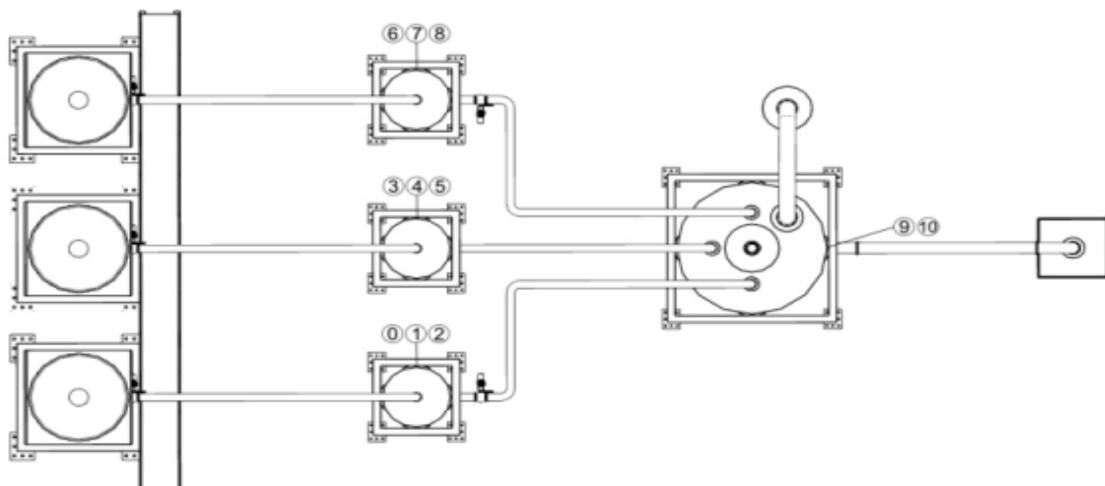
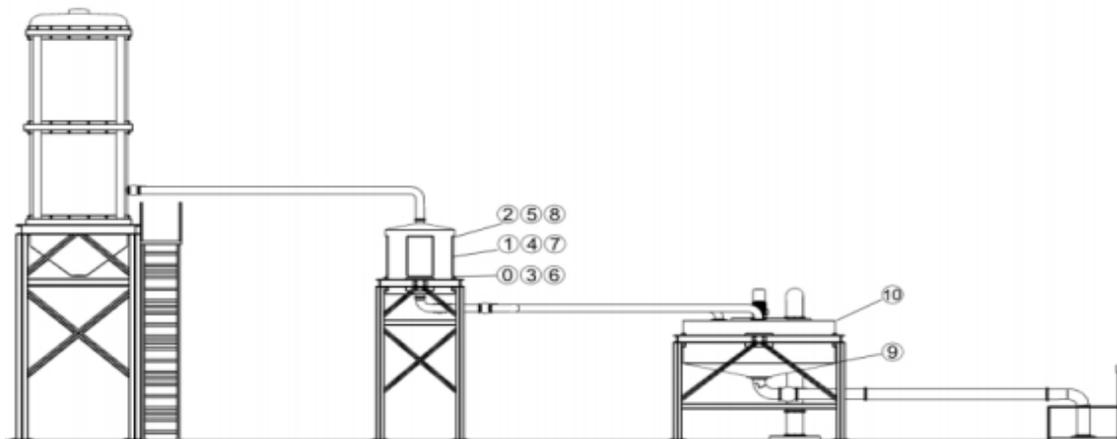
Suggestions :

- ☞ Commencer par un programme qui produit une couleur de façon répétitive.

☞ Utiliser des temporisations pour doser chaque couleur et obtenir ainsi des couleurs différentes de celles décrites dans le tableau ci-dessous.

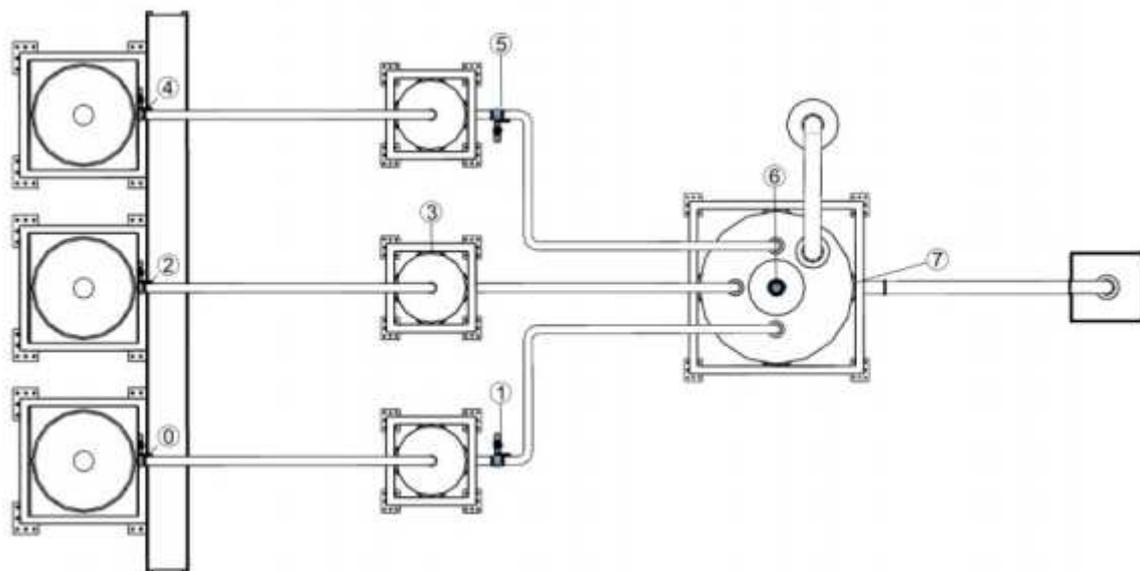
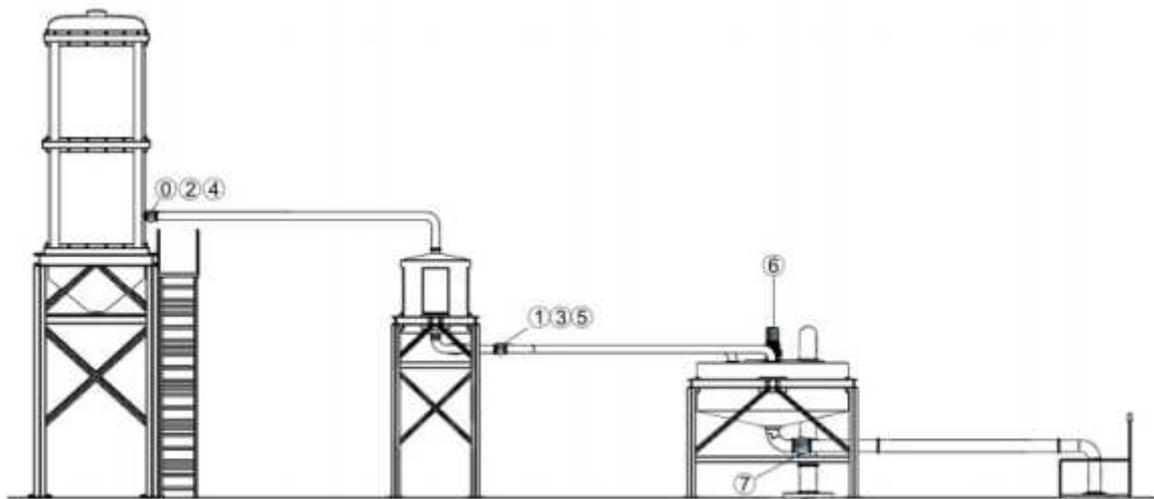
Les entrées de système (les capteurs) :

Capteurs	Description
0	Capteur niveau bas de la cuve de dosage de peinture rouge (cuve vide)
1	Capteur niveau milieu de la cuve de dosage de peinture rouge
2	Capteur niveau haut de la cuve de dosage de peinture rouge (cuve pleine)
3	Capteur niveau bas de la cuve de dosage de peinture verte (cuve vide)
4	Capteur niveau milieu de la cuve de dosage de peinture verte
5	Capteur niveau haut de la cuve de dosage de peinture verte (cuve pleine)
6	Capteur niveau bas de la cuve de dosage de peinture bleue (cuve vide)
7	Capteur niveau milieu de la cuve de dosage de peinture bleue
8	Capteur niveau haut de la cuve de dosage de peinture bleue (cuve pleine)
9	Capteur niveau bas de la cuve de mélange
10	Capteur niveau haut de la cuve de mélange



Les sorties de système (Les actionneurs) :

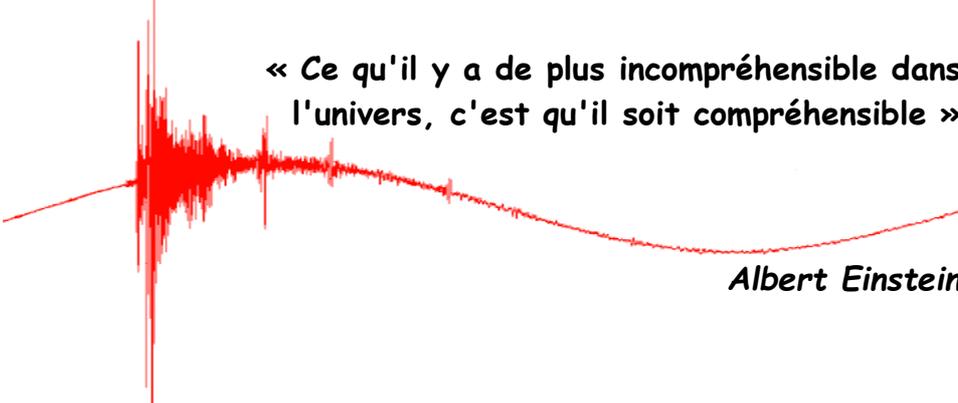
Actionneur	Description
0	Vanne de vidange du réservoir de peinture rouge
1	Vanne de vidange de la cuve de dosage de peinture rouge
2	Vanne de vidange du réservoir de peinture verte
3	Vanne de vidange de la cuve de dosage de peinture verte
4	Vanne de vidange du réservoir de peinture bleue
5	Vanne de vidange de la cuve de dosage de peinture bleue
6	Mélangeur
7	Vanne de vidange de la cuve de mélange



Relation entre les entrées et les sorties de ITS PLC et l'API simulé :

ITS PLC	AUTOMGEN
Capteur 0	I0
Capteur 1	I1
Capteur 2	I2
Capteur 3	I3
Capteur 4	I4
Capteur 5	I5
Capteur 6	I6
Capteur 7	I7
Capteur 8	I8
Capteur 9	I9
Capteur 10	I10
Sélecteur Manuel/Auto	I11
Bouton Start	I12
Bouton Stop	I13
Bouton Reset	I14
Bouton d'Arrêt d'Urgence	I15
Actionneur 0	O0
Actionneur 1	O1
Actionneur 2	O2
Actionneur 3	O3
Actionneur 4	O4
Actionneur 5	O5
Actionneur 6	O6
Actionneur 7	O7
Voyant du bouton Marche	O8
Voyant du bouton Reset	O9

CONCLUSION GÉNÉRALE



« Ce qu'il y a de plus incompréhensible dans
l'univers, c'est qu'il soit compréhensible »

Albert Einstein

CONCLUSION GÉNÉRALE

Répondant à la demande incessante du marché local et international, il est indispensable de suivre l'évolution des techniques d'automatisation industrielle. Actuellement, les industriels sont passés du stade de la machine à fonctionnement manuel à celui d'une chaîne utilisant des machines automatisées.

L'exploitation et la programmation des automates programmable industriels ne fait qu'accroître de jour en jour et devient de plus en plus sollicitée dans plusieurs domaines. Elle est basée principalement sur des connaissances préalables, tels que la logique combinatoire, séquentielle et un langage de programmation, elle permet d'automatiser efficacement beaucoup de cas de la productivité. Bien évidemment, ce polycopié traite la programmation des APIs où on a donné quelques exemples et des outils les plus utilisés dans l'industrie des API.

Une partie permet de comprendre la structure d'un Automate Programmable Industriel (API). Ce dernier est un ensemble électronique qui gère et assure la commande d'un système automatisé.

D'autre part, ce polycopié est accompagné par un complément des travaux dirigés et pratiques implémentés sous l'environnement automatisme.

BIBLIOGRAPHIE

BIBLIOGRAPHIE

- [1] Jacques Cuvillier, “*Automatismes Industriels et Tertiaires*”, Manuscrit de Cours, l’Institut Universitaire de Technologie (IUT) de Nantes, la France, Mars 2003.
- [2] Cédric Sindjui, “*Le Grand Guide Des Systèmes de Contrôle-Commande Industriels*”, Edition Dunod, Décembre 2007.
- [3] P. Brard and G. Colombari, “*Outil de Description ses Automatismes Séquentiels : Le grafcet*”, Techniques de l’Ingénieur, 1995.
- [4] J.M. Dutertre, “*Automatique Linéaire*”, Manuscrit de Cours, Mars 2016.
- [5] R. David and H. Alla, “*Du grafcet aux Réseaux de Petri*”, Edition Hermes 2009.
- [6] Jean-Louis Boimond, “*Simulation des Systèmes de Production – Réseaux de Pétri*”, Manuscrit de Cours, Université de Adelaide, Sydney, l’Australie, le 18 Février 2003.
- [7] Bassam Kattan, “*Synthèse structurelle d’un Contrôleur Basée sur le Grafcet*”, Thèse de Doctorat de l’Université de Grenoble, France, 2 Septembre 2004.
- [8] G. Dechenaux, “*API et PC : Solutions Concurrentes ou Complémentaires*”, Techniques de l’Ingénieur, R 8022, 1998.
- [9] D. Dupont & D. Dubois, “*Réalisation Technologique du Grafcet*”, Techniques de l’Ingénieur, S 8032, 2002.
- [10] P. Jargot, “*Langage de Programmation pour API Norme CIE 1131-3*”, Techniques de l’Ingénieur, S 8030, 1999.
- [11] M. Bertrand, “*Automates Programmables Industriels*”, Techniques de l’Ingénieur, S 8015, 2001.
- [12] M. Blanchard, “*Comprendre, Maitriser et Appliquer le GRAFCET*”, Editions Capades, 2000.
- [13] S. Moreno & 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.
- [14] J. Perrin, M. Magniez, S. Sinibaldi, A. Cortial, S. Badeau, F. Fronteri, & F. Agate, “*Automatique Informatique Industrielle*”, Edition Dunod, 2001.
- [15] L. Bergougroux, “*Automates Programmables Industriels*”, Manuscrit de Cours, Université de Marseille, 2005.
- [16] C. Pourcel, “*Systèmes Automatisés de Production*”, Edition Cepadues, 2000.
- [17] Ilhem Hafsaoui, “*Capteurs et Chaines de Mesure*”, Manuscrit de Cours, Université de Jijel, 2020.
- [18] Bacem Jrad, “*Systèmes Automatisés*”, Manuscrit de Cours, Université de Djerba, 2012.
- [19] Bensaid Samir, “*Capteurs et Actionneurs*”, Manuscrit de Cours, Université de Bouira, 2014.
- [20] Miloudi Mohamed, “*Automatismes Industriels*”, Manuscrit de Cours, Université de Ahmed Zabana, 2017.

- [21] Yann Morere, “Réseau de Petri”, Manuscrit de Cours, 2012.
- [22] Ir. H. Lecocq, “Les Automates Programmables”, Manuscrit de Cours, Université de Liege, 2005.
- [23] Olivier Le Gallo, “Automatique des Systèmes Mécaniques”, Edition Dunod, Décembre 2009.
- [24] Ermili Zineb et Zentar Hajar, “Automatisation d’une Machine Coquilleuse par Siemens Simatic Step7”, Mémoire de Master, Université de de Fès, 2013.
- [25] R. Thibaut, “Automatismes Pneumatiques et Hydrauliques”, Edition A. DE BOECK.
- [26] Moez Ayadi, “Automatismes Industriels”, Manuscrit de Cours, Université de Nabeul, 2016.
- [27] Ait Dahmane Kahina et Ait Ziane Meziane, “Conception et Réalisation d’un Bras Manipulateur Commandé par API”, Mémoire de Master, Université de de Khemis Miliana, 2015.
- [28] ———, “Introduction à l’Automatisme”, Manuscrit de Cours, Université de Douala, 2014.
- [29] Tlili Kais, “Automatisme Industriel”, Manuscrit de Cours, Université de Sousse.
- [30] ———, “Le GEMMA”, Manuscrit de Cours, —.
- [31] Mohammed Maatou & Abderahman Bellagh, “Automatisation et Réalisation à Petite Echelle (Maquette) d’une Chaîne Transporteuse de Briques”, Mémoire de Master, Université Hassiba Benbouali de Chlef, 2006.
- [32] Patricia Jargot, “Langages de Programmation pour API Norme IEC 1131-3”, Techniques de l’Ingénieur, traité Informatique industrielle S8030.
- [33] S. Moreno et E. Peulot, “Le GEMMA”, Edition A. Capilez Educactiv 1997.
- [34] S. Moreno et E. Peulot, “Le GRAFCET Conception et Implémentation dans les Automates Programmables Industriels”, Edition A. Capilez. Educactiv 1997.
- [35] Jean-Yves FABERT, “Automatismes et Automatique”, Edition Ellipses 2005.
- [36] Dominique Le Page, Jacques Gillot, “Câblage des Entrées et des Sorties des Automates Programmables Industriels Dédiés à La Sécurité”, Manuscrit de Cours, l’Institut Nationale de Recherche et de Sécurité, Paris, Mai 2003.
- [37] A. Naifi, “Etude de cas Mise en Œuvre des Automates TSX Micro”, Manuscrit de Cours, —.
- [38] Ahmed Louchene, “Automate Programmable Industriel”, Manuscrit de Cours, Université de Batna, 2018.
- [39] Sofiane Doudou, “Automates Programmables Industriels”, Manuscrit de Cours, Université de Jijel, 2020.
- [40] Bugeia Joe, “Fonctions Principales d’un Système Automatisé”, Manuscrit de Cours, —, 2004.
- [41] Houidi Ajmi, “Commande des Systèmes Industriels”, Manuscrit de Cours, Université de Soussa, 2011.
- [42] ———, “La Programmation”, Manuscrit de Cours, ———.
- [43] Philippe Hoarau, “Automate Programmable”, Note de Cours, ———.