

Faculté de technologie
Département
d'Ingénierie des
Systèmes Electriques



كلية التكنولوجيا
قسم هندسة الأنظمة
الكهربائية

Polycopié de cours
&
Exercices corrigés

Filière : Automatique

Niveau : 3^{ème} Année Licence

Module : Automates Programmables Industriels

Semestre : 02

Préparé par : IKHLEF Boualem

Année universitaire : 2023 - 2024

Table des matières

Chapitre I	Généralités sur les systèmes automatisés	1
I.1	Introduction	1
I.2	Description des différentes Parties d'un système automatisé	2
I.2.1	Définitions :	2
I.2.1.1	Système :	2
I.2.1.2	Système automatisé :	2
I.2.1.4	Automatique :	2
I.2.1.5	Automatisation :	2
I.2.2	Pourquoi l'automatisation :	2
I.3	Architecture d'un système automatisé	3
I.3.1	Partie Commande (PC).....	3
I.3.2	Partie Opérative (PO).....	3
I.3.3	Partie Relation (PR)	3
I.4	Différents équipements d'un système automatisé :	4
I.4.1	Pré-actionneurs :	4
I.4.1.1	Distributeurs pneumatiques	4
I.4.1.2	Distributeurs Hydraulique	6
I.4.2	Actionneurs.....	7
I.4.2.1	Actionneurs électriques	7
I.4.2.2	Relais électrique	8
I.4.2.3	Actionneurs Pneumatiques :	8
I.4.3	Capteurs :	10
I.4.3.1	Types de capteurs	10
I.4.4	Logiciels dans l'automatisation	11
I.4.4.1	Programmation de l'automate	11
I.4.4.2	Supervision	11
I.5	Avantages et Inconvénients de l'automatisation :	12
I.5.1	Avantages	12
I.5.2	Inconvénients	12
I.6	Types de commandes.....	12
I.6.1	Commande en boucle ouverte	12
I.6.2	Commande en boucle fermée	12

I.7 Domaines d'application des systèmes automatisés :	13
I.8 Exemples de systèmes automatisés.....	13
I.9 De la logique câblée à la logique programmée.....	14
I.9.1 Inconvénients :	15
I.9.1.1 Logique câblée	15
I.9.1.2 Logique Programmée	15
I.9.2 Avantages	15
I.9.2.1 Logique câblée	15
I.9.2.2 Logique Programmée	15
I.10 Conclusion.....	15
• Exercices du chapitre	16
• Solution des exercices	20
Chapitre II	
Modélisation des automatismes (GRAFCET)	24
II.1 Introduction.....	24
II.2 Introduction à la logique combinatoire	25
II.3 Les équations logiques et portes logiques.....	25
II.4 Introduction au grafcet.....	26
II.4.1 Description du GRAFCET	27
II.4.1.1 Le point de vue :	27
II.4.1.2 Spécification :	27
II.4.1.3 Finesse :	27
II.4.2. Pourquoi le grafcet ?	27
II.4.3 Transcription d'un cahier des charges en grafcet	28
II.4.4 Concepts de base du GRAFCET	28
II.4.4.1 Etape.....	28
II.4.4.2 Transition.....	29
II.4.4.3 Liaisons orientées.....	29
II.4.5 Classification des actions associées aux étapes	30
II.4.5.1 Actions continues :.....	30
II.4.5.2 Actions conditionnelles :	30
II.4.5.3 Action mémorisée :	31
II.4.5 Règles d'évolution d'un GRAFCET	32
II.4.5.1 Règle N°1 : Condition initiale	32
II.4.5.2 Règle N°2 : Franchissement d'une transition.....	32

II.4.5.3 Règle N°3 : Evolution des étapes actives	32
II.4.5.4 Règle N°4 : Franchissement simultané	32
II.4.5.5 Règle N°5 : Conflit d'activation.....	32
II.4.6 Structures de base	32
II.4.6.1 Notion de Séquence :	32
II.4.6.2 Saut d'étapes et reprise de séquence	33
II.4.6.3 Divergence en OU.....	33
II.4.6.4 Divergence convergence en ET :	34
II.4.6.5 Liaison entre grafjets :	35
II.4.7. Exemple :	35
II.5. GEMMA.....	38
II.5.1. Procédures du GEMMA.....	38
II.5.1.1 Famille F « Procédure de fonctionnement » :.....	38
II.5.1.2 Famille A« Procédure d'arrêt » :.....	39
II.5.1.3 Famille D « Les procédures en défaillances » :	40
II.5.2 Boucles opérationnelles de GEMMA :	41
II.5.2.1 Boucle de réglage :.....	41
II.5.2.2 Boucle de Fonctionnement	42
II.5.2.3 Boucle de sécurité	43
II.5.3 Exemple :	43
II.6 Conclusion :	46
• Exercices de chapitre sur le GRAFCET	47
• Solution	52
• Exercices sur le GRAFCET et le GEMMA.....	57
• Solution	59
Chapitre III	
Automates Programmables Industriels	62
III.1 Introduction	62
III.2 Qu'est-ce qu'un automate programmable.....	63
III.3 Les différents types d'automates	63
III.3.1 Automates compacts.....	63
III.3.2 Automates modulaires	64
III.4 Les éléments constitutifs des automates	65
III.4.1 Rack.....	65
III.4.2 Alimentation (PS).....	65

III.4.3 Unité de contrôle (CPU)	66
III.4.4 Module d'interface (IM)	66
III.4.5 Modules Entrées/Sorties (SM)	67
III.4.6 Modules de fonctions (FM).....	67
III.4.7 Modules de couplage (CP).....	68
III.5 Les critères de choix d'un automate	68
III.6 Différents types de données API.....	69
III.5.1 Booléen (Bool) :	69
III.6.2 Integer (INT):.....	69
III.6.3 Double Integer (DINT)	69
III.6.4 Real :	69
III.6.5 Temporisation :	69
III.6.6 Chaîne :	69
III.7 Câblage de l'API aux différentes E/S.....	69
III.7.1 Carte d'entrée	69
III.7.2 Carte de sortie	70
III. 8 Cartes d'entrées / sorties analogiques	71
III.8.1 Type de signaux analogiques :	71
III.8.2 Cartes d'entrée analogique.....	71
III.8.3 Cartes de sortie analogique	72
III.12 Conclusion	72
• Exercices du chapitre	73
• Solution :	75
Chapitre IV Programmation des Automates Programmables Industriels	77
IV. Introduction	77
IV.1 Cycle d'un automate programmable.....	78
IV.2 Différents langages de programmation	78
IV.2.1 Langage LADDER.....	79
IV.2.2 Langage List (IL) :	79
IV.2.3 Langage FBD :	80
IV.2.4 Langage SFC :	80
IV.2.5 Langage ST :	81
IV.3 Traduction d'un grafct en ladder	81
IV.3.1 Mise en équation d'un grafct :	81

IV.4 Programmation de grafcet à séquence unique	82
IV.4.1 Situation de deux étapes consécutives.....	82
IV.5 Programmation de grafkets à séquences multiples.....	85
IV.5.1 Situation de convergence en « OU »	85
IV.5.2 Situation de divergence en « OU »	86
IV.5.3 Situation de convergence en « ET ».....	86
IV.5.4 Situation de divergence en « ET »	86
IV.6 Conclusion :.....	91
• Exercices du chapitre :	92
• Solution.....	93
1. Grafcet de conduite :.....	94
2. Grafcet de sécurité :.....	99
3. Grafcet de production	100
4. Programme des Actions	104

Figure 01	Structure d'un système automatisé	4
Figure 02	Distributeur 5/2 Monostable	5
Figure 03	Distributeur 5/2 Bistable.....	5
Figure 04	Distributeur 3/2 Bistable.....	5
Figure 05	Distributeurs Hydrauliques monostable	7
Figure 06	Contacteur électrique.....	7
Figure 07	Symbole d'un contacteur électrique	8
Figure 08	Relais de commande	8
Figure 09	Vérin double effet	9
Figure 10	Vérin simple effet.....	9
Figure 11	Moteur électrique triphasé	9
Figure 12	Capteur fin de course.....	10
Figure 13	Détecteur de proximité	10
Figure 14	Exemple de programme Step 7	11
Figure 15	Exemple d'une interface de supervision.....	11
Figure 16	Commande en boucle ouverte	12
Figure 17	Commande en boucle fermée	12
Figure 18	Distributeur de boissons	13
Figure 19	Porte de grande surface	13
Figure 20	Pilotage automatique	14
Figure 21	Constructeur automobile	14
Figure 22	Composants de commande électriques.....	18
Figure 23	schéma de puissance d'un moteur triphasé dans les deux sens de rotation	19
Figure 24	Commande de vérin avec séquenceur pneumatique.....	21
Figure 25	schéma de commande.....	22
Figure 26	schéma de puissance	22
Figure 27	commande directe d'un moteur triphasé	23
Figure 28	commande d'un moteur dans les deux sens de rotation	23
Figure 29	Exemple d'une fonction logique.....	25
Figure 30	Equations et portes logiques	26
Figure 31	Etapes d'un Grafcet	28
Figure 32	Transition dans un grafcet	29
Figure 33	Liaisons orientées.....	29
Figure 34	Actions associées à une étape	30
Figure 35	Actions continues	30
Figure 36	Action conditionnée simple	30
Figure 37	Action retardée.....	31
Figure 38	Action limitée	31
Figure 39	Action mémorisée	32
Figure 40	Grafcet à plusieurs séquences	33
Figure 41	Saut d'étape et reprise de séquence	33
Figure 42	Divergence / Convergence en "ou"	34
Figure 43	Divergence / Convergence en "ET"	34
Figure 44	synchronisation des grafjets	35
Figure 45	commande de chariot	35

Figure 46	grafcet de commande d'un chariot.....	36
Figure 47	Station de remplissage et de bouchage.....	36
Figure 48	grafcet de la station de remplissage et de bouchage.....	37
Figure 49	GEMMA.....	41
Figure 50	Boucle de réglage.....	42
Figure 51	Boucle de fonctionnement	42
Figure 52	Boucle de défaillance	43
Figure 53	Chariot avec son panneau de commande du	44
Figure 54	grafcet de sécurité du chariot	44
Figure 55	grafcet de production du chariot	45
Figure 56	grafcet de conduite	45
Figure 57	commande d'un portail	47
Figure 58	Commande de deux vérins	48
Figure 59	Remplissage d'une caisse de 10 pièces.....	48
Figure 60	Système de traitement de surface.....	49
Figure 61	Station de remplissage et de bouchage.....	49
Figure 62	Station de perçage de pièces	50
Figure 63	grafcet à corriger	51
Figure 64	chargement - déchargement par chariots	52
Figure 65	grafcet de portail	52
Figure 66	grafcet de commande de vérins.....	53
Figure 67	grafcet de remplissage de caisse de 10 pièces	53
Figure 68	grafcet de la station de traitement de surface.....	54
Figure 69	grafcet de la station de remplissage et de bouchage.....	54
Figure 70	Station de perçage de pièces	55
Figure 71	grafcet corrigé.....	56
Figure 72	chargement - déchargement de deux chariots	56
Figure 73	chargement et déchargement de chariot avec pupitre de commande	57
Figure 74	Four avec son pupitre de commande	58
Figure 75	grafcet de sécurité	59
Figure 76	grafcet de production.....	59
Figure 77	grafcet de conduite	60
Figure 78	grafcet de sécurité	60
Figure 79	Grafcet de Production.....	61
Figure 80	Grafcet de conduite	61
Figure 81	Automate programmable de type S7-300	63
Figure 82	Automates programmables Compact	64
Figure 83	Automate modulaire Rockwell	64
Figure 84	Automate programmable modulaire SIEMENS	64
Figure 85	Différents modules de l'automate modulaire.....	65
Figure 86	Rack	65
Figure 87	Module d'alimentation pour la gamme SIEMENS	66
Figure 88	CPU 1500 / 1200 pour la gamme SIEMENS.....	66
Figure 89	Module d'interface	67
Figure 90	Module Entrées/Sortie (SIEMENS)	67

Figure 91	Module FM de commande de position	68
Figure 92	Module de communication	68
Figure 93	Exemple câblage de la carte d'entre (TOR)	70
Figure 94	Exemple du câblage de la carte sortie	71
Figure 95	Câblage entrée analogique en courant	72
Figure 96	Câblage de la carte de sortie analogique	72
Figure 97	Perceuse industrielle.....	73
Figure 98	Composants d'automates programmables (partie 01).....	73
Figure 99	Composants d'automates Programmables (partie 02)	74
Figure 100	Automate programmable S7-300 de SIEMENS	74
Figure 101	Etapes d'un cycle de fonctionnement	78
Figure 102	Exemple d'un réseau Ladder	79
Figure 103	Exemple d'un réseau List.....	80
Figure 104	Exemple d'un réseau FBD	80
Figure 105	Exemple d'un grafcet en step 7	80
Figure 106	Exemple d'un réseau du langage structuré.....	81
Figure 107	Exemple d'un grafcet	83
Figure 108	Exemple de grafcet avec divergence en "ou".....	88
Figure 109	Grafcet de Conduite	92
Figure 110	Grafcet de sécurité.....	93
Figure 111	Grafcet de production.....	93

Chapitre I

Généralités sur les systèmes automatisés

I.1 Introduction

De nos jours le monde industriel utilise de plus en plus des systèmes automatisés dans le but d'améliorer la productivité, la sécurité du personnel, minimiser les couts de maintenance, notamment en installant de la maintenance préventive, faciliter les procédures de maintenance, améliorer la qualité de la production, bien gérer le stock de produit ou de pièces de rechanges...etc.

Ce chapitre va donner un aperçu général sur l'automatisation et les systèmes de production automatisés.

I.2 Description des différentes parties d'un système automatisé

I.2.1 Définitions :

I.2.1.1 Système :

Un système est un ensemble d'éléments interconnectés d'une manière logique qui réalise une tâche bien définie.

I.2.1.2 Système automatisé :

Un système automatisé est un système qui réalise une tâche pour laquelle il a été conçu et ce, sans l'intervention humaine.

I.2.1.4 Automatique :

Est la science qui traite l'ensemble de moyens et de techniques pour l'analyse et la commande des systèmes physiques, dans le but de contraindre un système à se comporter d'une manière donnée.

I.2.1.5 Automatisation :

Est une branche de l'automatique qui est orientée vers l'industrie, l'automate Programmable industriel est l'élément clé de l'automatisation, c'est le cerveau du Système automatisé.

I.2.2 Pourquoi l'automatisation :

L'automatisation est devenue indispensable de nos jours, elle permet de donner :

- **Une meilleure qualité de produit** : ce qui va permettre à l'entreprise de se positionner dans la liste des entreprises qui fournissent des produits de qualité. Ce résultat est rendu possible grâce à un bon réglage des paramètres de la machine.
- **Une bonne gestion de la sécurité** : avec l'automatisation le pourcentage des risques est diminué grâce à des capteurs qui signalent la moindre anomalie et une bonne gestion des procédures post-incident.
- **Une bonne gestion de la maintenance** : avec l'automatisation, les procédures de maintenances sont optimisées, cela permet de réduire le cout de réparation et éliminer les arrêts inattendus avec la maintenance préventive.
- **Une meilleure productivité** : avec l'automatisation, la quantité de produit réalisée est multipliée, la machine travaille de façon répétitive et sans arrêt.

I.3 Architecture d'un système automatisé

Un système automatisé est composé en général de deux parties, la première partie s'agit de la partie opérative (PO) qui constitue le système à automatiser qui est équipé de pré-actionneurs, d'actionneurs et de capteurs, la deuxième partie s'agit de la partie commande (PC) qui est le cerveau du système automatisé.

On retrouve une troisième partie dans quelques systèmes automatisés qui est la partie relation (PR), cette partie gère le dialogue entre l'homme et la machine.

I.3.1 Partie Commande (PC)

Cette partie a pour rôle d'interpréter et de générer des ordres selon les informations qui arrivent de la machine, à partir des consignes données par les opérateurs et également du programme implémenté dans sa mémoire.

Cette partie est composée d'un organe de commande qui est en général un automate programmable ou des cartes de commandes.

I.3.2 Partie Opérative (PO)

Cette partie est constitué du système à commander à qui on ajoute des pré-actionneurs, d'actionneurs et des capteurs. Les actionneurs ont pour tâche d'exécuter les ordres de la partie commande et les capteurs ont la tâche d'informer la partie commande de l'évolution de la machine.

I.3.3 Partie Relation (PR)

Cette partie a pour but d'informer l'opérateur de l'état de partie opérative et permet également à l'opérateur de donner ces consignes.

Exemples :

- Pupitre de commande et de signalisation.
- Poste de contrôle.
- Interface Homme machine (HMI)

La figure (I.1) suivante donne une représentation de la schématique de la structure d'un système automatisé.

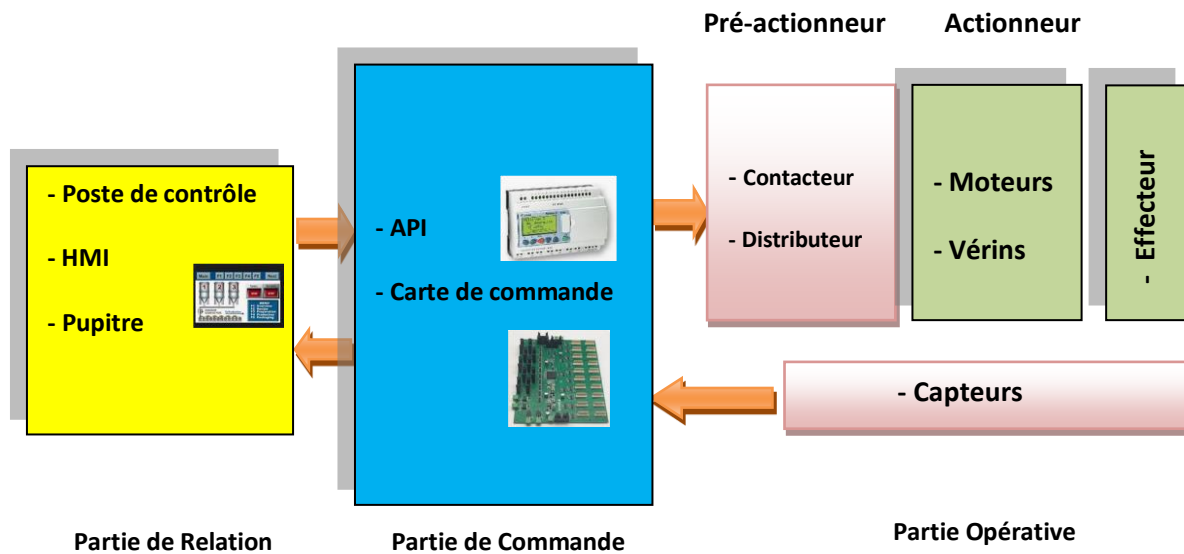


Figure 1 : Structure d'un système automatisé

La partie commande coordonne et gère la succession des ordres donnés par cette dernière et les informations issues des capteurs.

I.4 Différents équipements d'un système automatisé :

L'automatisation a besoin de certains équipements industriels qui sont :

I.4.1 Pré-actionneurs :

Sont des organes qui distribuent de l'énergie vers l'actionneur sous l'ordre de la partie commande, il représente l'interface entre la partie commande et la partie opérative, on retrouve deux types :

I.4.1.1 Distributeurs pneumatiques

Ce type de distributeurs est utilisé pour les petites puissances, ils supportent des pressions à l'entour de 6 bars.

- 1. Distributeur monostable :** c'est un distributeur qui a un seul état stable, lorsque la commande est donnée, le distributeur change d'état, et lorsqu'elle est levée, le distributeur revient à sa position de départ.
- 2. Distributeur Bistable :** c'est un distributeur qui a deux états stables, lorsqu'il est commandé, il change d'état et reste dans cette position même si la commande est

levée, il faut actionner la deuxième commande pour qu'il revienne à sa position de départ.

Exemples :

- Distributeur 5/2 monostable

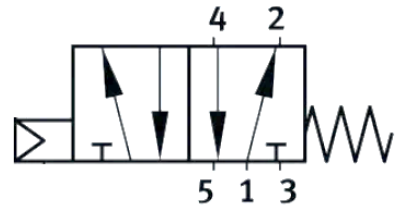


Figure 2 Distributeur 5/2 Monostable

- Distributeur 5/2 Bistable

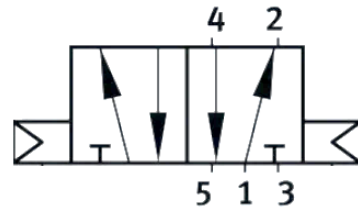


Figure 3 Distributeur 5/2 Bistable

- Distributeur 3/2 Bistable

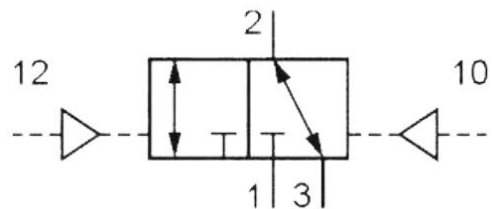
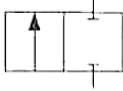
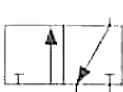
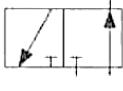
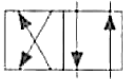
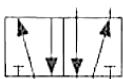
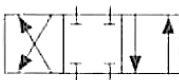


Figure 4 Distributeur 3/2 Bistable

Dans le tableau ci-dessous de « **Philippe HOARAU** », on retrouve les principaux distributeurs avec les principaux dispositifs de commande.

Tableau 01 : Principaux distributeurs et principaux dispositifs de commande

	symbole	orifices	positions	symboles de pilotages	
2/2	 N.F.	2	2	 général	manuel
2/2	 N.O.	2	2	 bouton poussoir	
				 levier	
3/2	 N.F.	3	2	 pédale	mécanique
				 poussoir	
3/2	 N.O.	3	2	 ressort	
				 galet	
4/2		4	2	 1 enroulement	électro-aimant
5/2		5	2	 2 enroulements inversés	
				 hydraulique	distributeur pilote
4/3	 centre fermé	4	3	 pneumatique	
				 par détente	
5/3	 centre ouvert	5	3	 électro-aimant + distributeur pilote	
				 électro-aimant ou distributeur pilote	
N.F. : normalement fermé N.O. : normalement ouvert					

I.4.1.2 Distributeurs Hydraulique

Ce type de distributeurs est utilisé pour les grandes puissances, ils supportent des pressions à l'alentour de 400 bars.

Le fonctionnement de ces distributeurs reste le même que les distributeurs pneumatiques.

Exemples :



Figure 5 Distributeurs Hydrauliques monostable

I.4.2 Actionneurs

I.4.2.1 Actionneurs électriques

Un contacteur électrique est un organe qui distribue l'énergie électrique pour les actionneurs électriques, il est commandé par la partie commande, son rôle est de couper ou de distribuer l'énergie électrique pour le moteur voire la figure suivante :



Figure 6 Contacteur électrique

I.4.2.1.1 Principe de fonctionnement

Le contacteur contient une bobine et un mécanisme composé de deux parties, une fixe et l'autre mobile et un ressort de rappel, si la bobine est excitée la partie mobile est attirée et se met en contact avec la partie fixe réalisant ainsi un contact électrique, si la bobine n'est plus excitée le ressort de rappel remet la partie mobile à sa position initiale coupant ainsi le contact électrique.

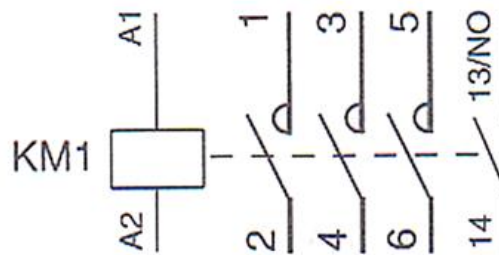


Figure 7 Symbole d'un contacteur électrique

I.4.2.2 Relais électrique

Un relais thermique fonctionne de la même manière qu'un contacteur à la seule différence les relais sont utilisés pour des petites puissances notamment dans le circuit de commande.



Figure 8 Relais de commande

I.4.2.3 Actionneurs Pneumatiques :

Les actionneurs pneumatiques sont les vérins, on retrouve principalement deux types ; simple effet et double effet.

I.4.2.3.1 Vérin double effet :

Ce type de vérin contient deux prises de commande, une pour faire sortir la tige et l'autre pour la faire rentrer. Ce vérin est généralement associé aux distributeur 4/2, 4/3, 5/2, 5/3...

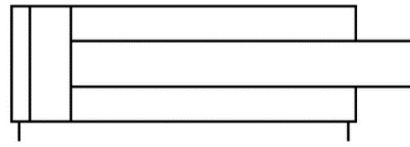


Figure 9 Vérin double effet

I.4.2.3.2 Vérin simple effet :

Ce type de vérin contient une prise de commande pour faire sortir la tige et le rappel se fait grâce à un ressort. Ce vérin est généralement associé aux distributeur 3/2.

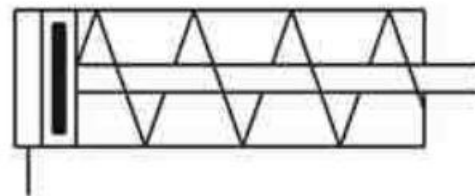


Figure 10 Vérin simple effet

I.4.2.3.3 Actionneurs hydraulique :

Les actionneurs hydrauliques sont des vérins hydrauliques et fonctionnent de la même manière que les vérins pneumatiques.

I.4.2.3.4 Actionneurs électromécaniques :

Les actionneurs électromécaniques sont des moteurs, et il existe plusieurs types de moteurs, mais dans l'industrie on trouve généralement des moteurs asynchrones triphasés et monophasés.



Figure 11 Moteur électrique triphasé

I.4.3 Capteurs :

Sont des composants qui font partie de la chaîne de retour de l'automatisation, leur rôle est de détecter un phénomène physique et de le convertir en un signal exploitable (électrique, pneumatique...etc).

I.4.3.1 Types de capteurs

On peut classer les capteurs de deux façons :

- **Mode d'utilisation :**

Si on classe les capteurs selon leur mode d'utilisation, on trouve des capteurs qui fonctionnent lorsqu'il y a un contact direct avec l'objet et d'autres capteurs fonctionnent sans contact directe avec l'objet, ils exploitent le son, la lumière le champ infrarouge ou électromagnétique.

Exemple :

- Capteur fin de course fonctionne avec contact direct avec l'objet



Figure 12 Capteur fin de course

- Capteur de proximité fonctionne s'il y a un objet suffisamment proche.



Figure 13 Détecteur de proximité

I.4.4 Logiciels dans l'automatisation

On retrouve deux principaux logiciels dans l'automatisation, le premier dans la programmation des automates et le deuxième dans la supervision de l'installation.

I.4.4.1 Programmation de l'automate

Pour la programmation des automates, un logiciel dédié est indispensable, par exemple : pour les automates SIEMENS le « STEP 7 » est incontournable, pour les automates Schneider le « PL7 junior ou premium » sont indispensables.

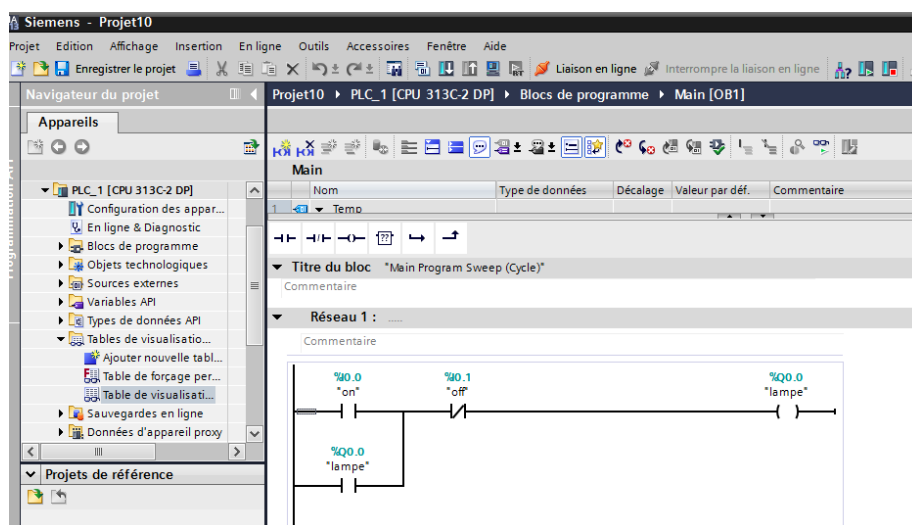


Figure 14 Exemple de programme Step 7

I.4.4.2 Supervision

Pour réaliser une supervision d'une installation, des logiciels spéciaux sont requis, par exemples : WinCC, FoxView ...

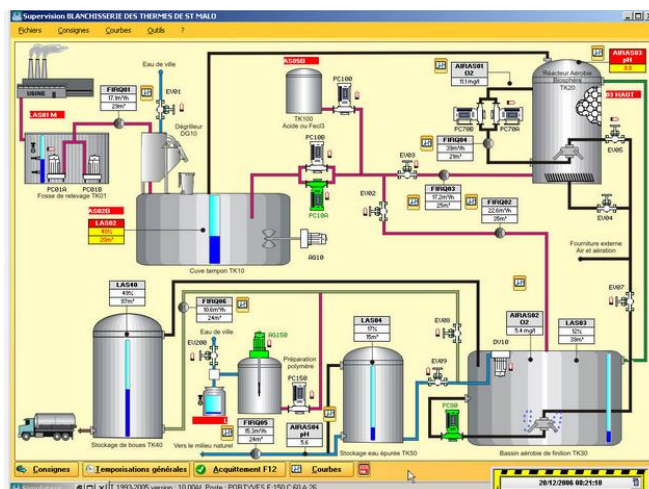


Figure 15 Exemple d'une interface de supervision

I.5 Avantages et Inconvénients de l'automatisation :

I.5.1 Avantages

- Améliorer la productivité.
- Travailler dans des zones dangereuses.
- Une grande souplesse pour une extension.
- Evite des tâches pénibles à l'homme.

I.5.2 Inconvénients

- Le coût élevé du matériel.
- La maintenance doit être bien organisée.
- La suppression d'emplois

I.6 Types de commandes

I.6.1 Commande en boucle ouverte

La commande en boucle ouverte est une commande dans laquelle la partie opérative n'envoie pas de compte rendu sur son état à la partie commande, elle reçoit seulement les commandes, les ordres générés par la partie commande dépendent seulement des consignes de l'opérateur et du programme de la partie commande.

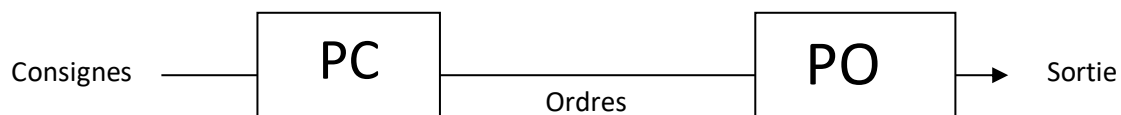


Figure 16 Commande en boucle ouverte

I.6.2 Commande en boucle fermée

La commande en boucle fermée est une commande dans laquelle la partie commande reçoit des informations sur l'état de la partie opérative, à partir de ces informations, du programme installé et des consignes de l'opérateur, elle génère des ordres vers la partie opérative.

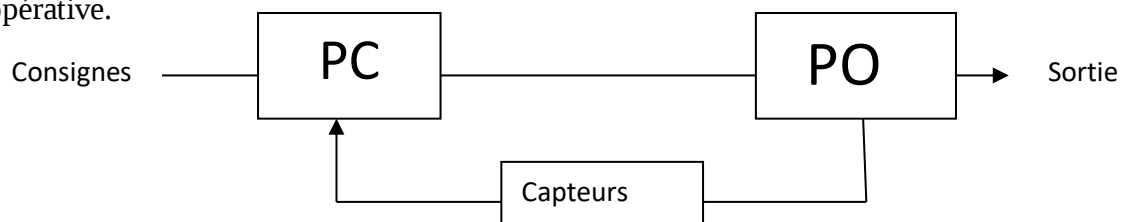


Figure 17 Commande en boucle fermée

I.7 Domaines d'application des systèmes automatisés :

Aujourd'hui, presque tous les domaines industriels et non industriels comportent des automatismes tels que :

- Constructeurs Automobile
- Domaine d'aviation
- Industrie manufacturière
- Industrie Médicale
- Transport avec ses différents types
- Electroménager (machine à laver vaisselle ...)

I.8 Exemples de systèmes automatisés

- **Distributeur de boissons**



Figure 18 Distributeur de boissons

- **Port de grande surface**



Figure 19 Porte de grande surface

- Aviation (pilote automatique)



Figure 20 Pilotage automatique

- Constructeur automobile

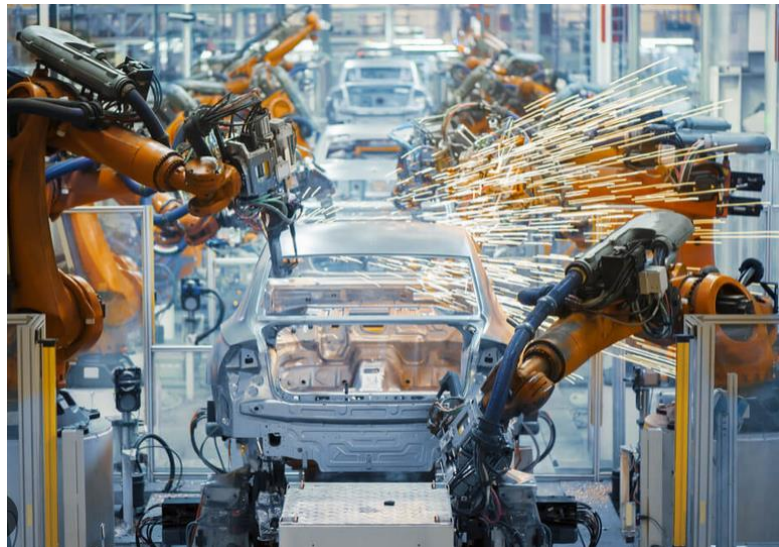


Figure 21 Constructeur automobile

I.9 De la logique câblée à la logique programmée

Avant l'arrivée de l'informatique dans les automatismes, la commande des systèmes industriels était réalisée avec de la logique câblée, c'est-à-dire avec le câblage des composants électriques tels que des boutons, des relais temporisateurs, des contacteurs... afin d'obtenir des fonctions logiques avec lesquelles le système est commandé.

Après l'arrivée de l'informatique dans les automatismes, on parle de la logique programmée, car on utilise des programmes pour commander les systèmes. Dans cette technique quelques instructions de programme remplacent un grand schéma de câblage.

I.9.1 Inconvénients :

I.9.1.1 Logique câblée

- Encombrante
- En cas de modification, il faut refaire tout le câblage
- Beaucoup de composants signifie moins de fiabilité

I.9.1.2 Logique Programmée

- Matériels très coûteux
- Il faut une maintenance structurée
- Vitesse inversement proportionnelle à la complexité du problème.

I.9.2 Avantages

I.9.2.1 Logique câblée

- Rapidité, l'exécution est simultanée
- Composants pas chers

I.9.2.2 Logique Programmée

- Moins encombrant
- Fiabilité
- Souplesse pour une extension
- Facile à programmer

I.10 Conclusion

L'automatisation est devenue une partie indispensable de l'industrie, telle que sans l'automatisation on ne peut pas avoir de meilleure qualité ni une grande quantité de produit, ce qui va rendre le marché concurrentiel de l'entreprise une chose très difficile.

L'automatisation offre une meilleure fiabilité et surtout elle garantit la sécurité des opérateurs et de l'environnement.

• Exercices du chapitre

Exercice 01 :

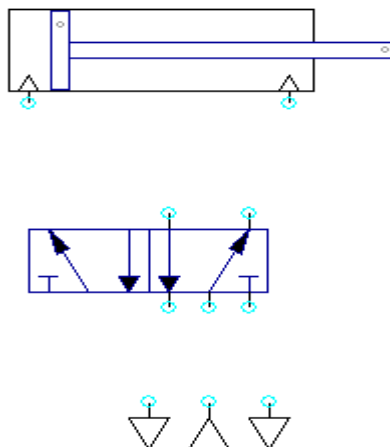
Veillez classer les composants suivants selon leurs fonctions :

- Vérin
- Distributeur pneumatique
- Contacteur
- Lampe
- Moteur
- Buzzer
- Résistance chauffante
- Interrupteur
- Bouton poussoir
- Compresseur
- Fin de course

Pré-actionneur	Actionneur	Capteur
-		

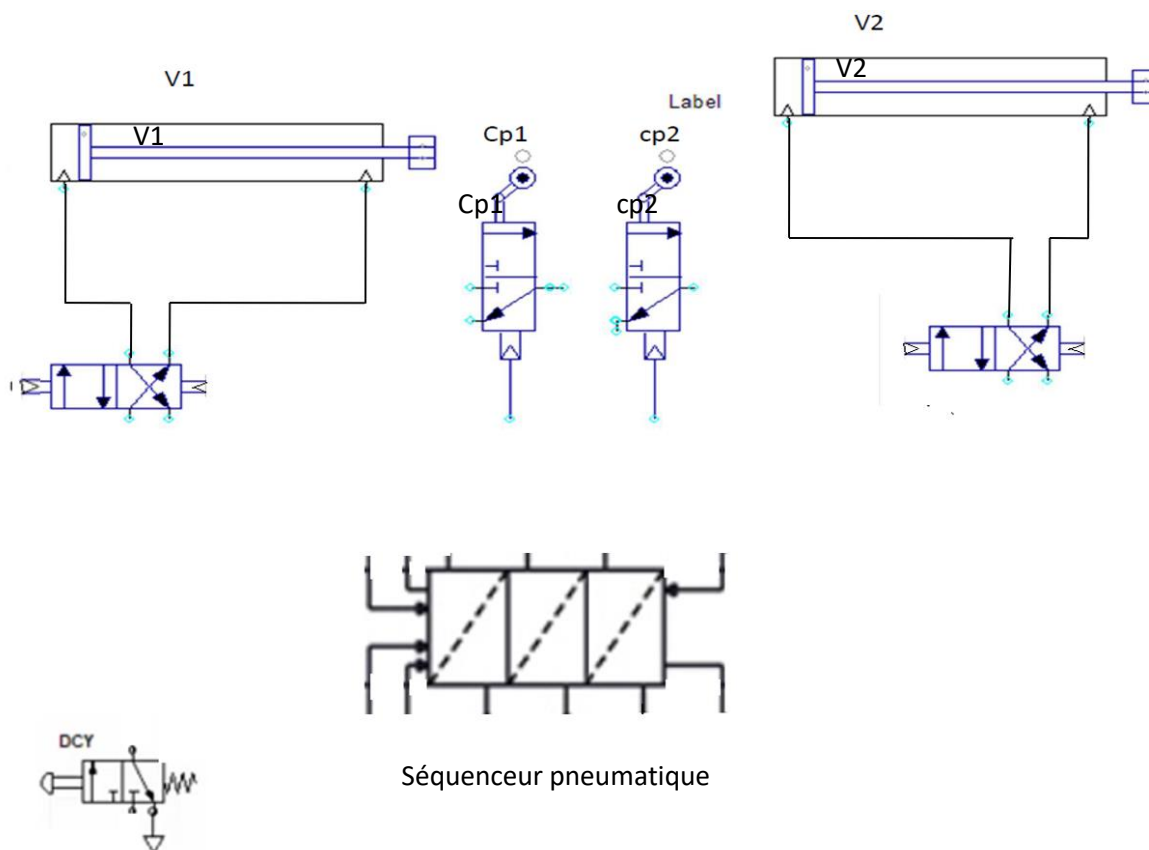
Exercice 02

Réaliser la commande manuelle et électrique d'un vérin pneumatique double effets

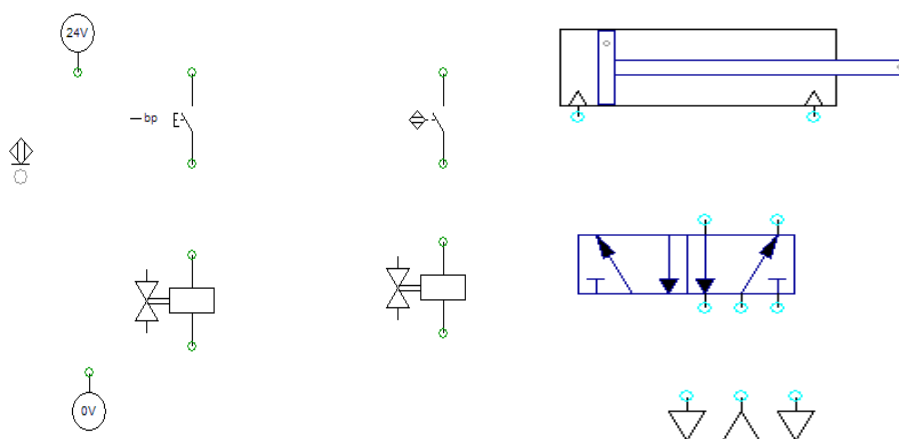


Exercice 03

Réaliser la commande pneumatique de deux vérins pneumatiques double effets ; Compléter le schéma de sorte que le capteur Cp1 fait sortir le vérin V2 et le capteur Cp2 le fait rentrer.

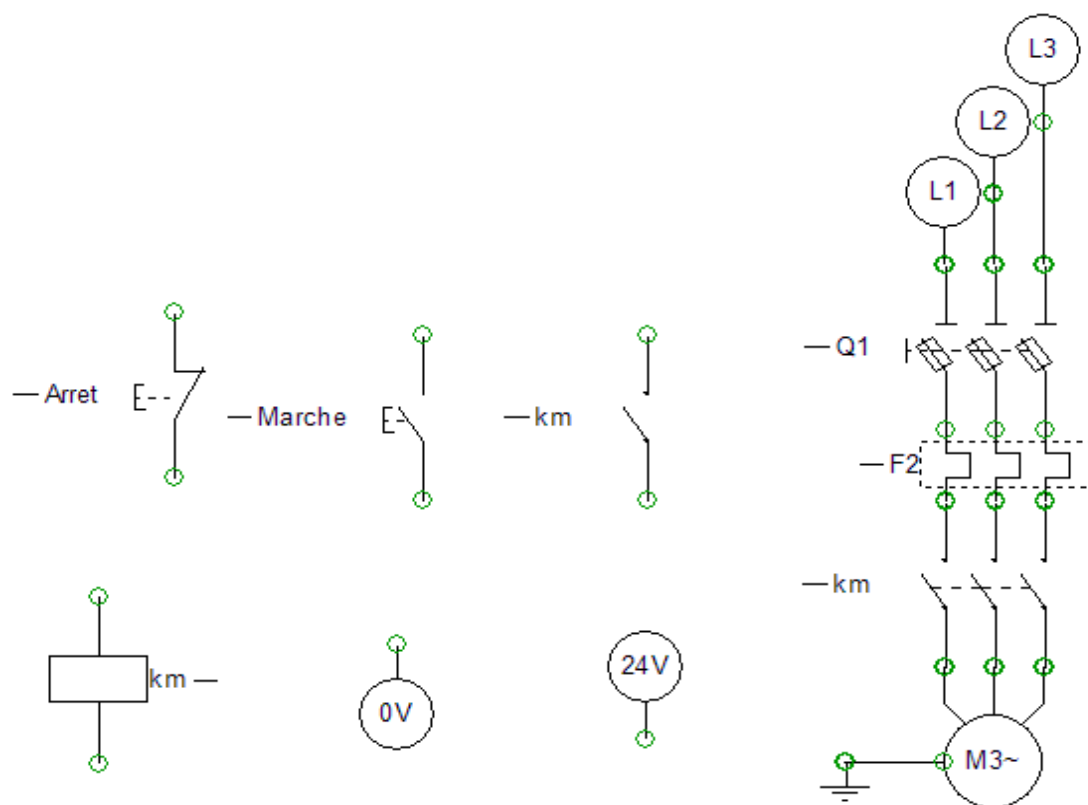


Exercice 04 : Réaliser le circuit de commande d'un vérin double effets en utilisant les composants suivants :



Exercice :05 Commande d'un Moteur Electrique

Réaliser le schéma de commande en utilisant les composants suivants :



Exercice 06 : Commande d'un Moteur Electrique dans les deux sens de rotation

Réaliser le schéma de commande d'un moteur asynchrone triphasé dont le schéma de puissance est donné ci-dessous, et ce en utilisant les composants suivants :

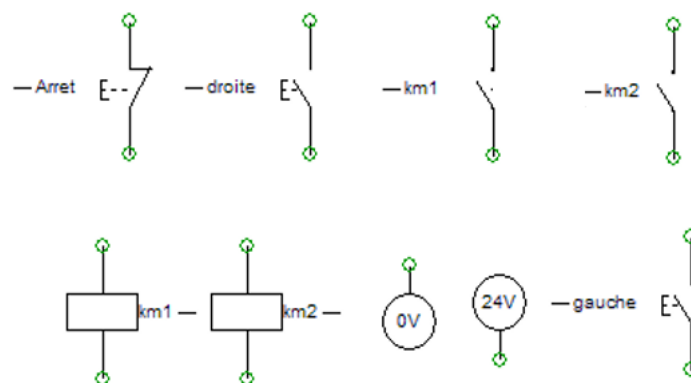


Figure 22 Composants de commande électriques

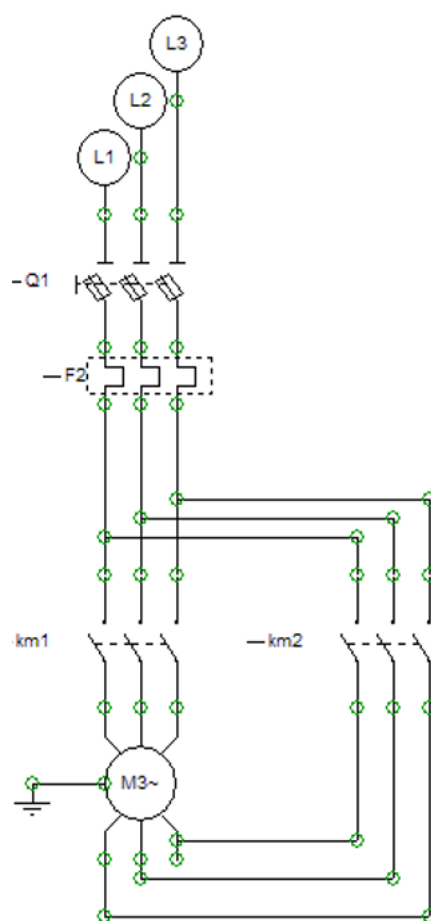


Figure 23 schéma de puissance d'un moteur triphasé dans les deux sens de rotation

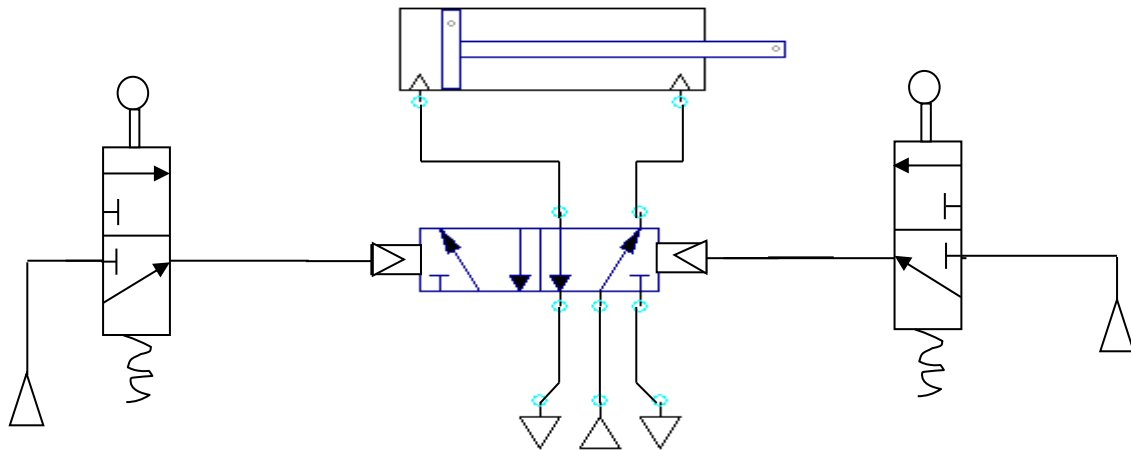
• Solution des exercices

Exercice 01 :

Pré-actionneur	Actionneur	Capteur
- Distributeur - Contacteur	- Vérin - Moteur - Résistance chauffante - Lampe - Buzzer - Compresseur	- Fin de course - Interrupteur - Bouton poussoir

Exercice 02 :

1. Commande pneumatique



On utilise deux distributeurs 3/2 pour réaliser des bouton poussoir pneumatique

2. Commande électrique

Pour la commande électrique, il faut un schéma de commande pour commander les électrovannes de distributeur.

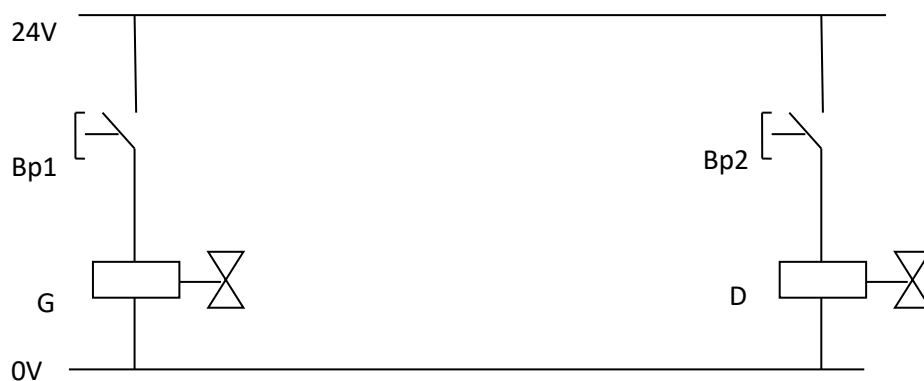
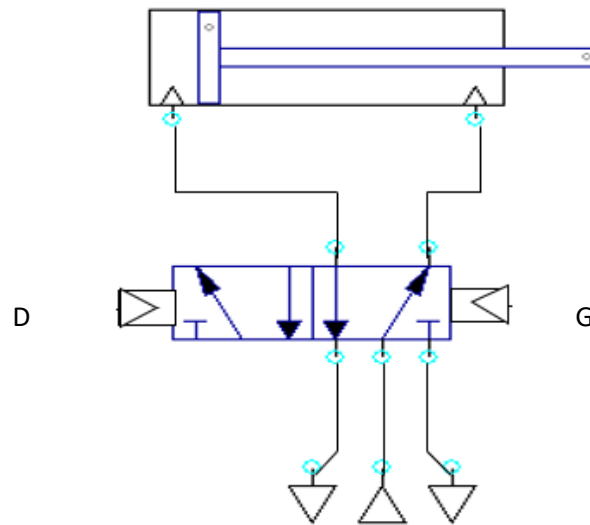


Schéma de puissanceExercice 03

Dans ce cas on utilisera le signal de premier distributeur 3/2 pour faire sortir le deuxième vérin, et le signal de deuxième distributeur 3/2 pour le faire rentrer.

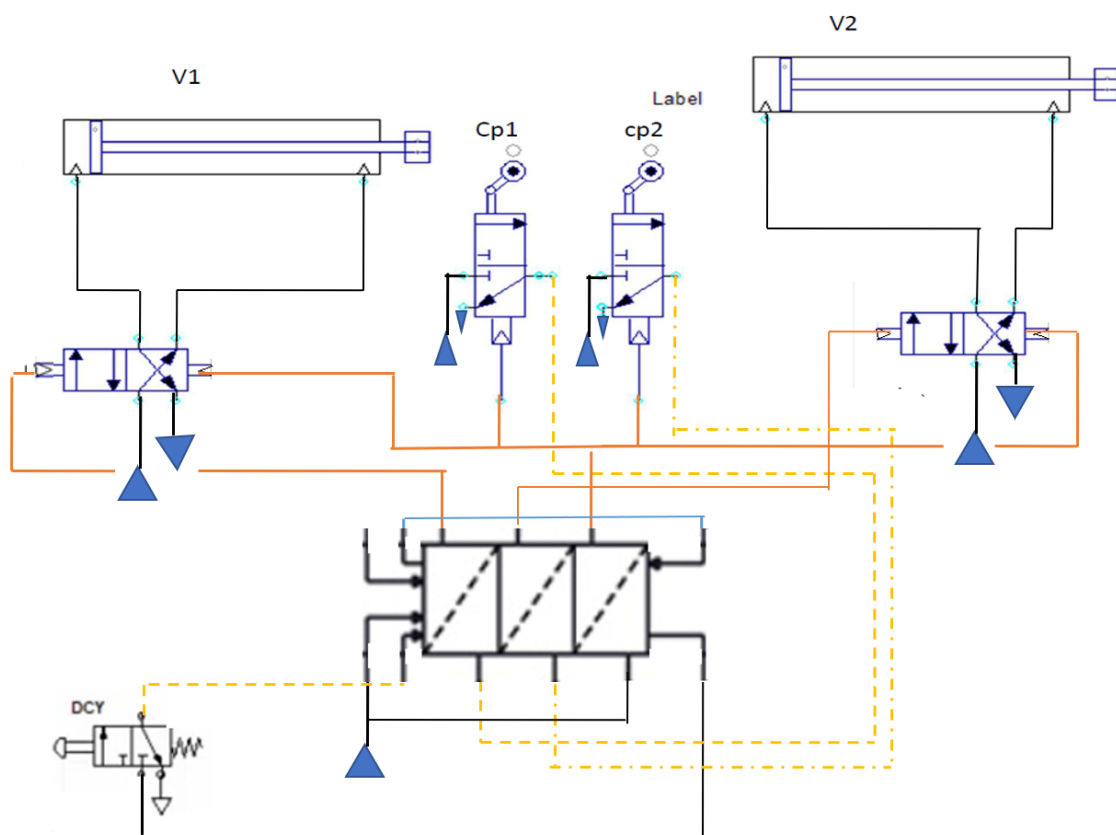


Figure 24 Commande de vérin avec séquenceur pneumatique

Exercice 04 :

Commande électrique d'un vérin pneumatique avec capteur de fin de course

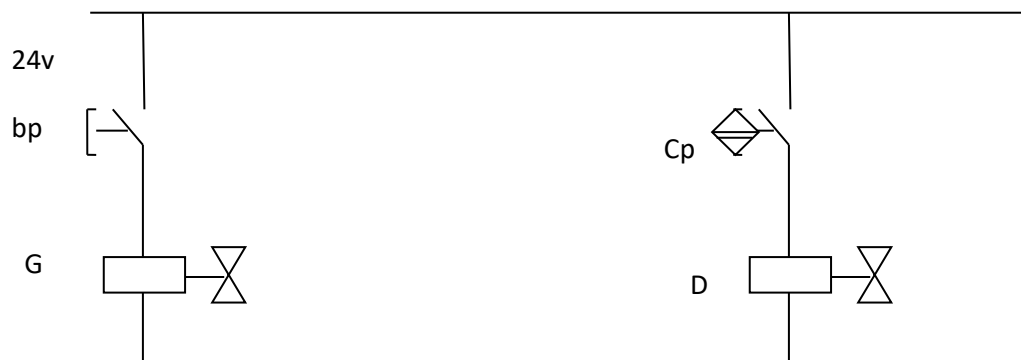


Figure 25 schéma de commande

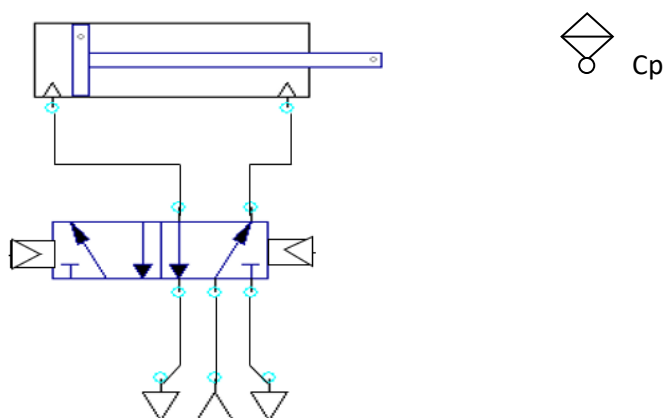


Figure 26 schéma de puissance

Exercice 05

Commande directe d'un moteur asynchrone

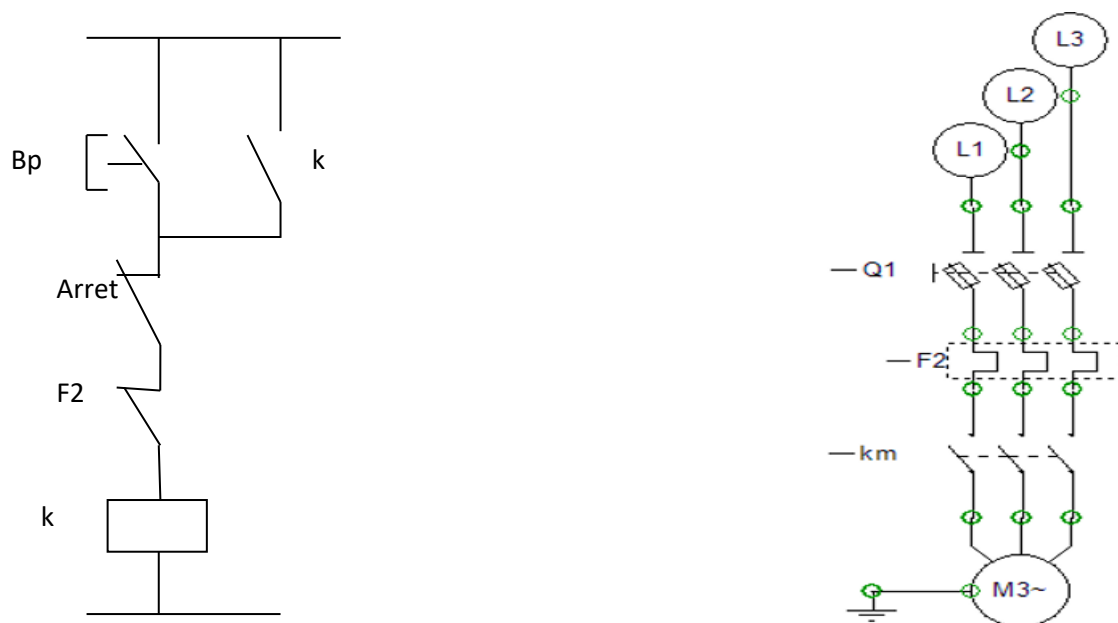


Figure 27 commande directe d'un moteur triphasé

Exercice 06 :

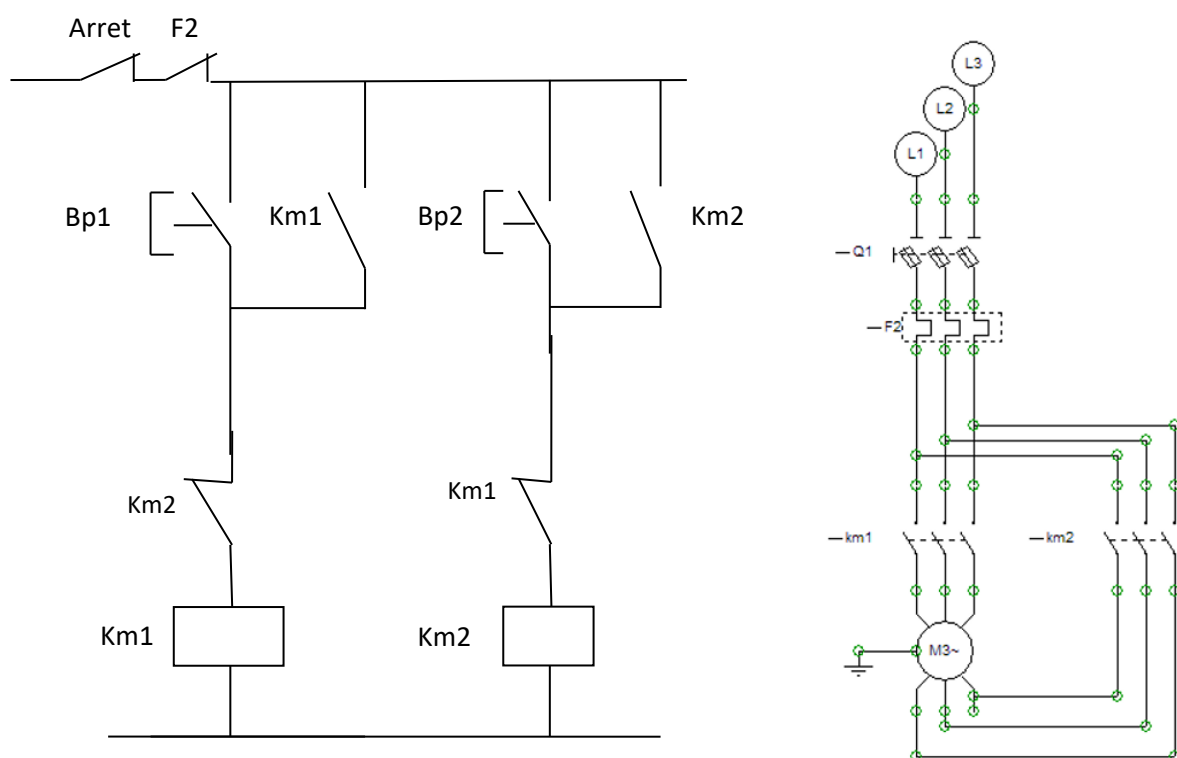


Figure 28 commande d'un moteur dans les deux sens de rotation

Chapitre II Modélisation des automatismes (GRAFCET)

II.1 Introduction

Pour commander des systèmes, l'automates doit traiter plusieurs données à savoir les consignes de l'opérateur et les informations issues de capteurs, selon un programme qui est implémenté dans sa mémoire par un automaticien, ce programme dicte à l'automate comment les données seront traitées et comment les ordres seront générés. Pour réaliser cette fonction l'automate utilise de la logique combinatoire.

II.2 Introduction à la logique combinatoire

Les variables de la logique combinatoire sont soit vraies ou fausses qui correspondent respectivement à « 0 » ou à « 1 ». Une fonction combinatoire est le résultat de la combinaison de plusieurs variables logiques, parmi les caractéristiques de la fonction combinatoire c'est que le résultat est instantané, et qu'elle ne possède pas de fonction mémoire c'est-à-dire, elle dépend seulement des entrées, et il suffit juste d'avoir des variables en entrée pour avoir le résultat de la fonction. Les outils utilisés pour analyser ces fonctions sont l'algèbre de Boole, tables de vérités et tableaux de Karnaugh.

II.3 Les équations logiques et portes logiques

Toute fonction logique est exprimée par une relation entre deux ou plusieurs portes logiques qui sont à leurs tour en fonction des variables d'entrées, il existe plusieurs portes logiques, dans l'exemple suivant, on trouve une fonction logique qui est en fonction de plusieurs portes logiques.

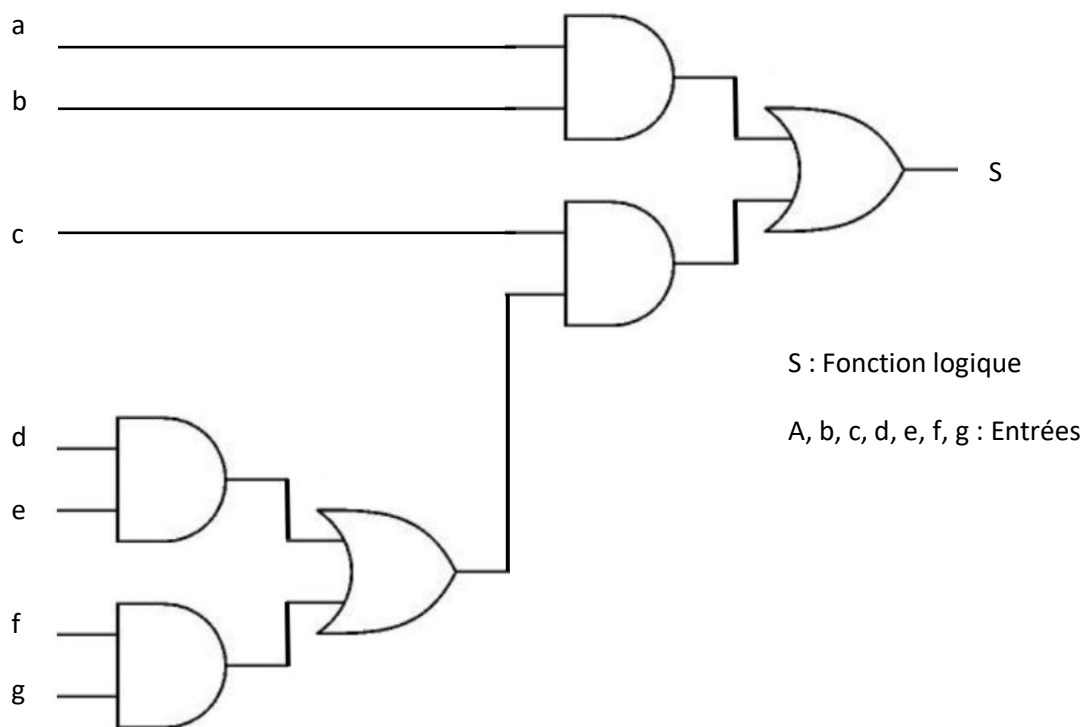


Figure 29 Exemple d'une fonction logique

Dans cet exemple le résultat de la fonction est égale : $S = a.b + c.(d.e + f.g)$

Le tableau suivant montre les équations et les portes logiques les plus utilisées

Fonction logique	Table de vérité	Equation logique	Opérateur logique (symbole)																
			Norme française	Norme américaine															
NON	<table><tr><td>e_1</td><td>S</td></tr><tr><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td></tr></table>	e_1	S	0	1	0	1	1	0	1	0	$S = \overline{e_1}$							
e_1	S																		
0	1																		
0	1																		
1	0																		
1	0																		
OU	<table><tr><td>e_1</td><td>e_2</td><td>S</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	e_1	e_2	S	0	0	0	0	1	1	1	0	1	1	1	1	$S = e_1 + e_2$		
e_1	e_2	S																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	1																	
ET	<table><tr><td>e_1</td><td>e_2</td><td>S</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	e_1	e_2	S	0	0	0	0	1	0	1	0	0	1	1	1	$S = e_1 \cdot e_2$		
e_1	e_2	S																	
0	0	0																	
0	1	0																	
1	0	0																	
1	1	1																	
NON OU (NOR)	<table><tr><td>e_1</td><td>e_2</td><td>S</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	e_1	e_2	S	0	0	1	0	1	0	1	0	0	1	1	0	$S = \overline{e_1 + e_2}$		
e_1	e_2	S																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	0																	
NON ET (NAND)	<table><tr><td>e_1</td><td>e_2</td><td>S</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	e_1	e_2	S	0	0	1	0	1	1	1	0	1	1	1	0	$S = \overline{e_1 \cdot e_2}$		
e_1	e_2	S																	
0	0	1																	
0	1	1																	
1	0	1																	
1	1	0																	
OU EXCLUSIF (EXOR)	<table><tr><td>e_1</td><td>e_2</td><td>S</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	e_1	e_2	S	0	0	0	0	1	1	1	0	1	1	1	0	$S = e_1 \oplus e_2$		
e_1	e_2	S																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	0																	
ET EXCLUSIF (EXAND)	<table><tr><td>e_1</td><td>e_2</td><td>S</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	e_1	e_2	S	0	0	1	0	1	0	1	0	0	1	1	1	$S = e_1 \odot e_2$		
e_1	e_2	S																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	1																	

Figure 30 Equations et portes logiques

II.4 Introduction au grafcet

(Graphe Fonctionnel de Commande des Étapes et Transitions) est un mode de représentation et d'analyse d'un automate, particulièrement bien adapté aux systèmes à évolution séquentielle, c'est-à-dire décomposable en étapes.

Le Grafcet est donc un langage graphique représentant le fonctionnement d'un automate par un ensemble de :

- Étapes auxquelles sont associées des actions ;
- Transitions entre étapes auxquelles sont associées des conditions (réceptivités)
- Liaisons orientées entre les étapes et les transitions.

II.4.1 Description du GRAFCET

La description du comportement attendu d'un automate peut être représentée par un **GRAFCET** d'un certain « **niveau** », la caractérisation du « **niveau** » du GRAFCET nécessite de prendre en compte trois dimensions :

II.4.1.1 Le point de vue :

Caractérisant le point de vue selon lequel un observateur s'implique dans le fonctionnement du système pour en donner une description, on distingue trois points de vue :

- Un point de vue système.
- Un point de vue Partie Opérative.
- Un point de vue Partie Commande.

II.4.1.2 Spécification :

Caractérisant la nature des spécifications techniques auxquelles doit satisfaire la Partie Commande, on distingue trois groupes de spécifications :

- Spécifications fonctionnelles,
- Spécifications technologiques,
- Spécifications opérationnelles.

II.4.1.3 Finesse :

Caractérisant le niveau de détail dans la description du fonctionnement, d'un niveau global (ou macro-représentation) jusqu'au niveau de détail complet où toutes les actions et informations élémentaires sont prises en compte.

II.4.2. Pourquoi le grafcet ?

Lorsque certaines spécifications sont exprimées en langage courant, il y a un risque permanent d'incompréhension. Certains mots sont peu précis, mal définis ou possèdent plusieurs sens.

Le langage courant est mal adapté pour décrire précisément les systèmes séquentiels. Le GRAFCET fut donc créé pour représenter de façon symbolique et graphique le fonctionnement d'un automate.

Cela permet une meilleure compréhension de l'automatisme par tous les intervenants. Un GRAFCET est établi pour chaque machine lors de sa conception, puis utilisé tout au long de sa vie : réalisation, mise au point, maintenance, modifications, réglages. Le langage GRAFCET

doit donc être connu de toutes les personnes concernées par les automatismes, depuis leur conception jusqu'à leur exploitation.

II.4.3 Transcription d'un cahier des charges en grafcet

Un cahier des charges est généralement donné sous forme d'un texte contenant toutes les instructions et recommandations sur le fonctionnement et sur la sécurité que doit fournir l'utilisateur pour le concepteur de l'automatisme afin de les respecter lors de la phase de conception.

Le grafcet de sécurité doit respecter toutes les consignes de sécurité, c'est-à-dire il doit prendre en charge toutes les conditions liées à la sécurité telles que :

- Arrêt d'urgence
- Capteur de flame, de gaz, de fumer...

Le grafcet de conduite doit gérer tous les modes de l'automatisme, le mode manuel et le mode automatique, et également la mise et la remise en route en l'automatisme.

Le grafcet de production normal doit gérer seulement la production, c'est-à-dire le fonctionnement normal de l'automatisme.

II.4.4 Concepts de base du GRAFCET

II.4.4.1 Etape

Une **étape** symbolise un état ou une partie de l'état du système automatisé, l'étape possède deux états possibles : **active** représentée par un jeton à l'intérieur de l'étape ou **inactive**. L'étape i représentée par un carré repéré numériquement, possède ainsi une variable d'état, appelée variable d'étape X_i , cette variable est une variable booléenne valant **1** si l'étape est active, **0** sinon.

La situation initiale d'un système automatisé est indiquée par une étape dite **étape initiale** et représentée par un carré double.

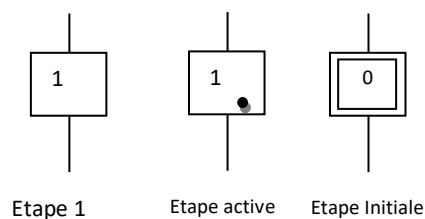


Figure 31 Etapes d'un Grafcet

II.4.4.2 Transition

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

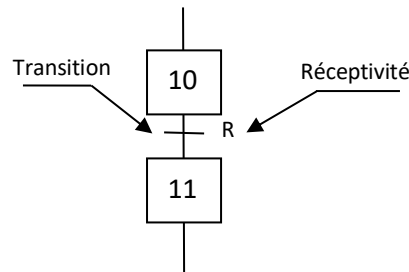


Figure 32 Transition dans un grafcet

La réceptivité qui est une information d'entrée qui est fournie par :

- L'opérateur : pupitre de commande,
- La partie opérative : états des capteurs,
- Temporisateur, compteur ou toute opération logique, arithmétique...
- Etape d'un grafcet : une étape d'un grafcet peut être utilisée comme réceptivité
- D'autres systèmes : dialogue entre systèmes,

Remarque :

Si la réceptivité n'est pas précisée, alors cela signifie qu'elle est toujours vraie. (=1)

II.4.4.3 Liaisons orientées

Elles sont de simples traits verticaux qui relient les étapes aux transitions et les transitions aux étapes, elles sont normalement orientées de haut vers le bas, une flèche est nécessaire dans le cas contraire.

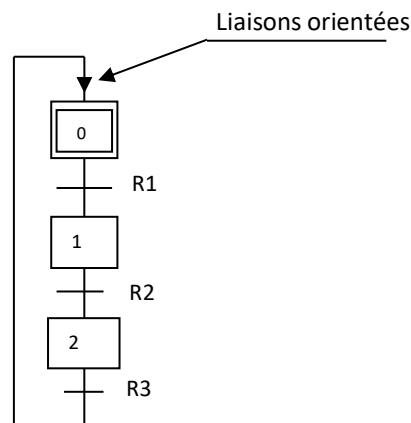


Figure 33 Liaisons orientées

II.4.5 Classification des actions associées aux étapes

A chaque étape est associée une **action** ou plusieurs, c'est à dire un ordre vers la partie opérative ou vers d'autres grafquets, mais on peut rencontrer aussi une même action associée à plusieurs étapes ou une étape **vide** (*sans action*).

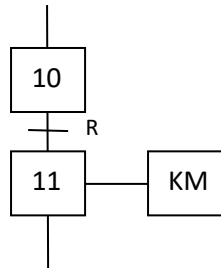


Figure 34 Actions associées à une étape

L'action associée à l'étape peut être de 3 types : **continue**, **conditionnelle** ou **mémorisée**. Les actions peuvent être classées en fonction de leur durée par rapport à celle de l'étape.

II.4.5.1 Actions continues :

L'ordre est émis de façon continue tant que l'étape à laquelle il est associé est active.

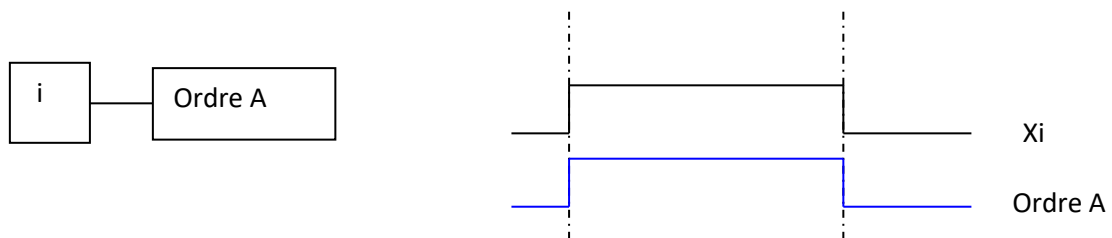


Figure 35 Actions continues

II.4.5.2 Actions conditionnelles :

Une action **conditionnelle** n'est exécutée que si l'étape associée est active et si la condition associée est vraie.

Les actions conditionnelles peuvent être décomposées en 3 cas particuliers :

II.4.5.2.1 Action conditionnelle simple : Type C

L'ordre A est exécutée si l'étape **i** est active et la condition associée à l'action est vraie

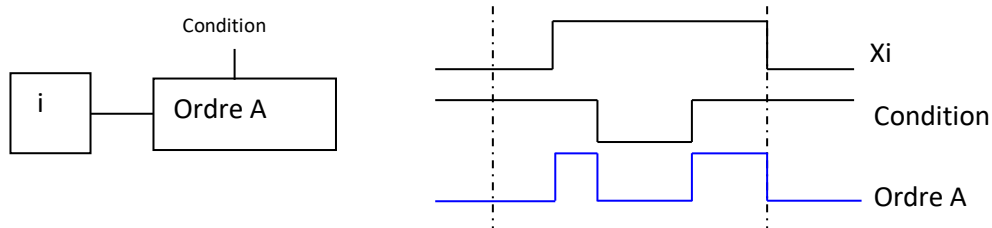


Figure 36 Action conditionnée simple

II.4.5.2.2 Action retardée : Type D (delay)

Le temps intervient dans cet ordre conditionnel comme condition logique. L'indication du temps s'effectue par la notation générale " $t / xi / q$ " dans laquelle " xi " indique l'étape prise comme origine du temps et " q " est la durée du retard.

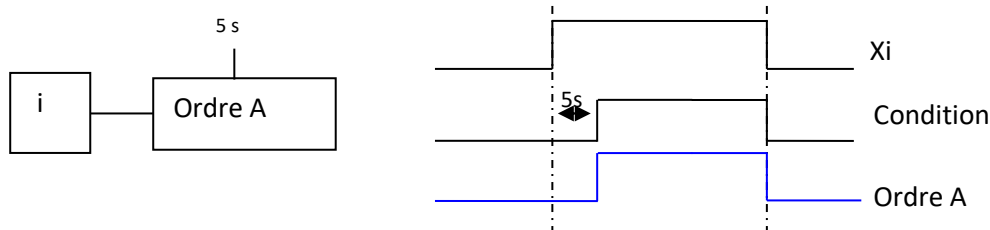


Figure 37 Action retardée

II.4.5.2.3 Action de durée limitée : Type L (limited)

L'ordre est émis dès l'activation de l'étape à laquelle il est associé ; mais la durée de cet ordre sera limitée à une valeur spécifiée.

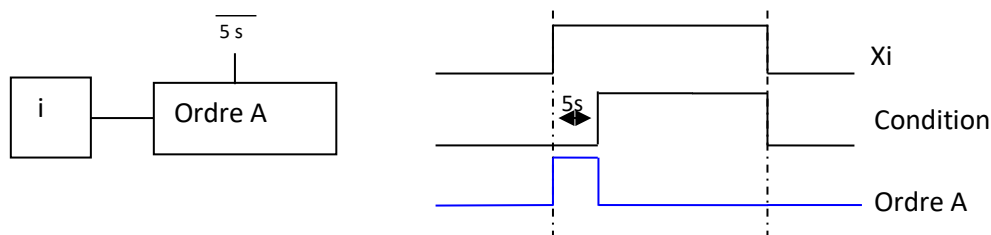


Figure 38 Action limitée

L'ordre "A" est limité à 2s après l'activation de l'étape 4.

II.4.5.3 Action mémorisée :

Le maintien d'un ordre sur la durée d'activation de plusieurs étapes consécutives, peut être obtenu par la mémorisation de l'action, et cela est obtenu par l'utilisation d'une fonction auxiliaire appelée fonction mémoire.

Le **S** pour l'activation et **R** pour la désactivation

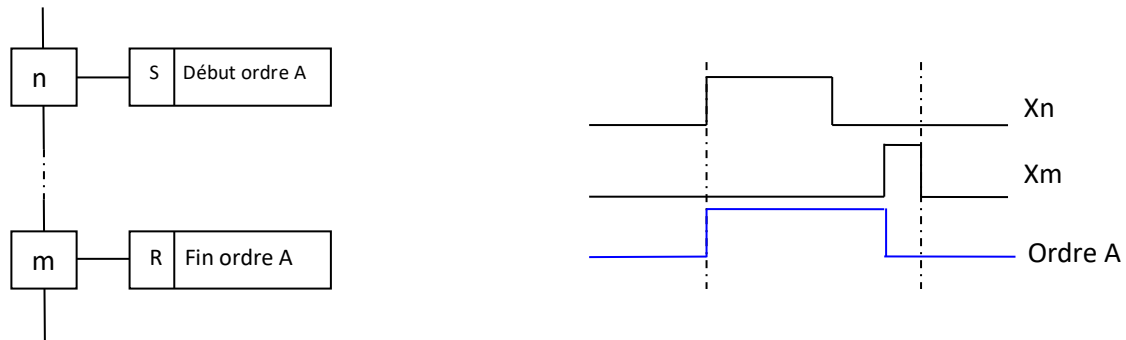


Figure 39 Action mémorisée

II.4.5 Règles d'évolution d'un GRAFCET

II.4.5.1 Règle N°1 : Condition initiale

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

II.4.5.2 Règle N°2 : Franchissement d'une transition

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

II.4.5.3 Règle N°3 : Evolution des étapes actives

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

II.4.5.4 Règle N°4 : Franchissement simultané

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

II.4.5.5 Règle N°5 : Conflit d'activation

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

II.4.6 Structures de base

II.4.6.1 Notion de Séquence :

Une séquence dans un Grafcet est une suite d'étapes à exécuter l'une après l'autre, autrement dit chaque étape ne possède qu'une seule transition aval et une seule transition amont.

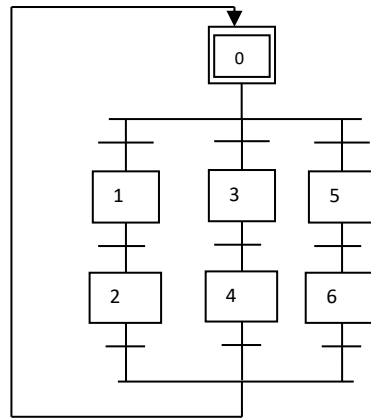


Figure 40 Grafcet à plusieurs séquences

II.4.6.2 Saut d'étapes et reprise de séquence

Le saut d'étapes permet de sauter une ou plusieurs étapes lorsque les actions associées sont inutiles à réaliser. La reprise de séquence (ou boucle) permet de reprendre, une ou plusieurs fois, une séquence tant qu'une condition n'est pas obtenue.

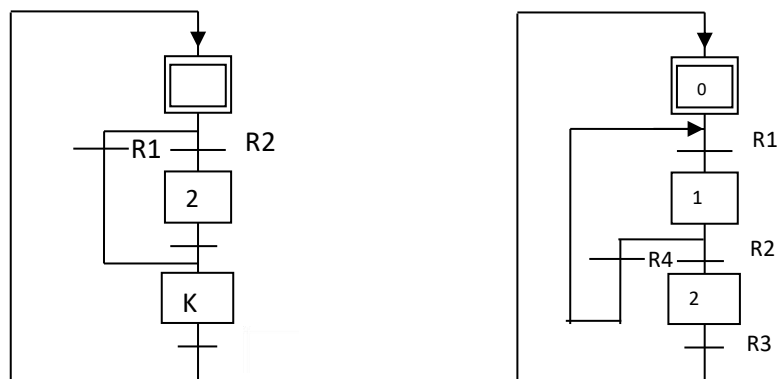


Figure 41 Saut d'étape et reprise de séquence

II.4.6.3 Divergence en OU

On dit qu'il y a **Aiguillage** ou **divergence en OU** lorsque le grafcet se décompose en deux ou plusieurs séquences selon un choix conditionnel. Comme la divergence en **OU** on rencontre aussi la convergence en **OU**, on dit qu'il y a convergence en **OU**, lorsque deux ou plusieurs séquences du grafcet converge vers une seule séquence.

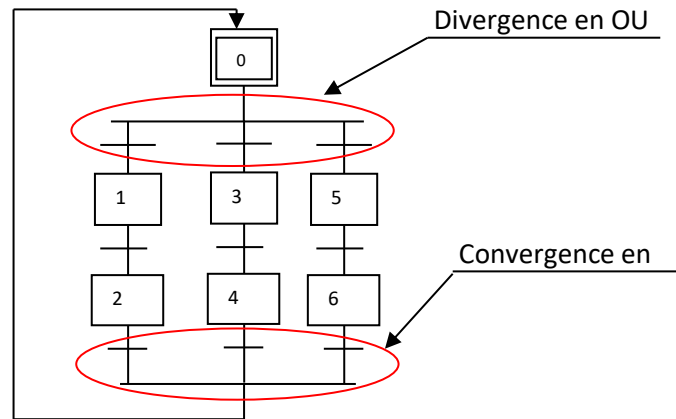


Figure 42 Divergence / Convergence en "ou"

Si les conditions de la convergence sont à **1** simultanément, les étapes 1, 3 et 5 vont devenir actives simultanément, situation non permise, donc ces conditions doivent être en relation exclusives.

II.4.6.4 Divergence convergence en ET :

Au contraire de l'aiguillage où ne peut se dérouler qu'une seule activité à la fois, On dit qu'on se trouve en présence d'un parallélisme structurel, si plusieurs activités indépendantes pouvant se dérouler en parallèle. Le début d'une divergence en **ET** et la fin d'une convergence en **ET** d'un parallélisme structurel sont représentés par deux traits parallèles.

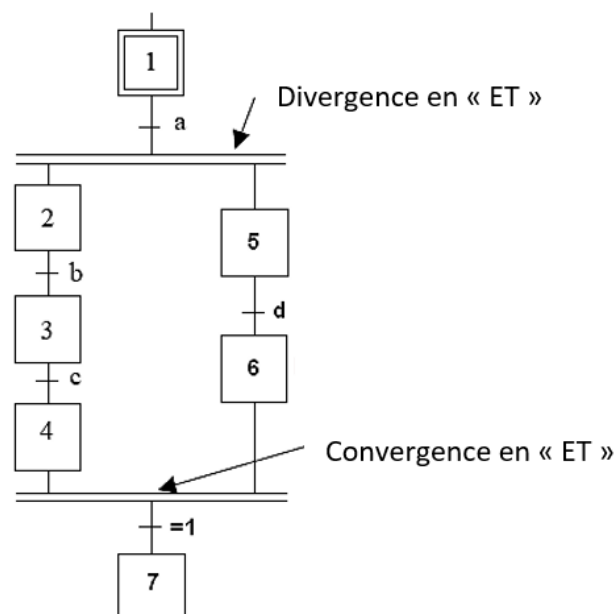


Figure 43 Divergence / Convergence en "ET"

La synchronisation permet d'attendre la fin de plusieurs activités se déroulant en parallèle, pour continuer par une seule séquence.

II.4.6.5 Liaison entre grafkets :

Une étape dans un grafket peut servir comme réceptivité à une autre étape d'un autre grafket. Cette méthode est utilisée aussi pour synchroniser deux grafkets c'est à dire rendre l'évolution de l'un dépendent de l'évolution de l'autre.

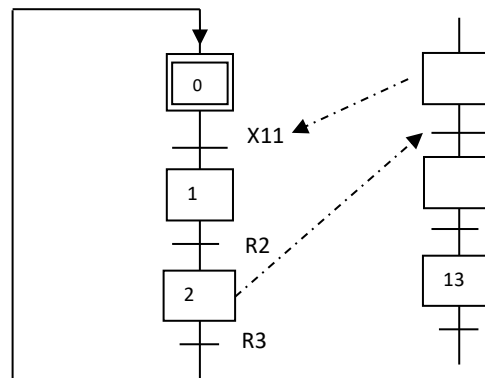


Figure 44 synchronisation des grafkets

II.4.7. Exemple :

Cahier des charges : après avoir donné l'ordre de départ (dcy), le chariot part jusqu'au point B puis revient au point C et repart au point B et delà revient directement au point A.

- Donner le grafket de ce système
- Donner les équations de chaque étape

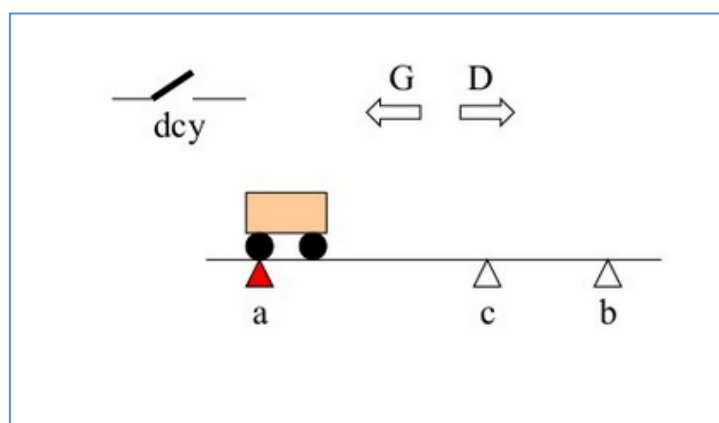


Figure 45 commande de chariot

Solution

- **Le Grafcet**

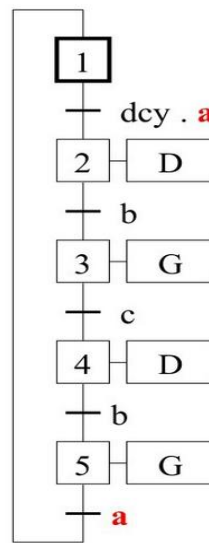


Figure 46 grafcet de commande d'un chariot

Exemple 02 :**Chaîne d'embouteillage**

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 P1 commandé par l'électrovanne EV.
- Un poste de bouchage P2 commandé par un vérin presseur V à double effet.

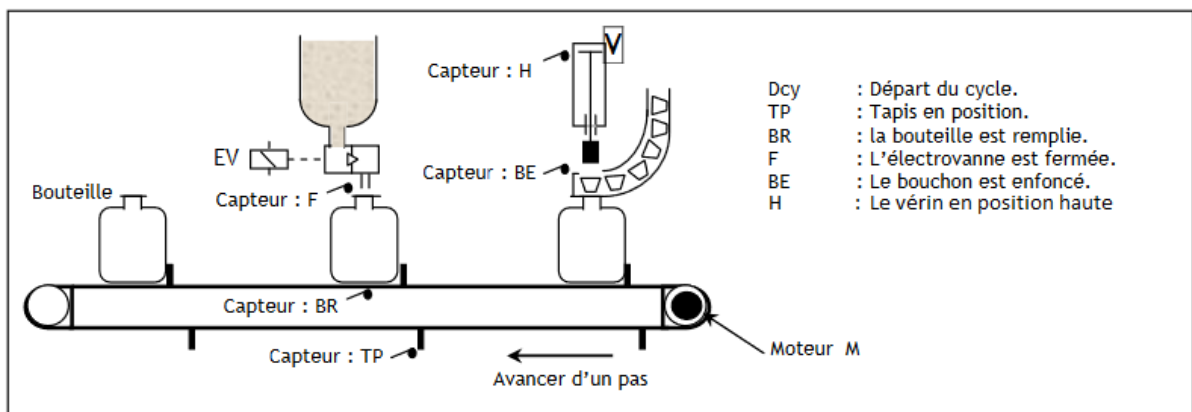


Figure 47 Station de remplissage et de bouchage

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 : TP. Une bouteille est alors présente à chacun des postes P1 et P2. 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 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 V ;
- Remonte du vérin V 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.

Solution :

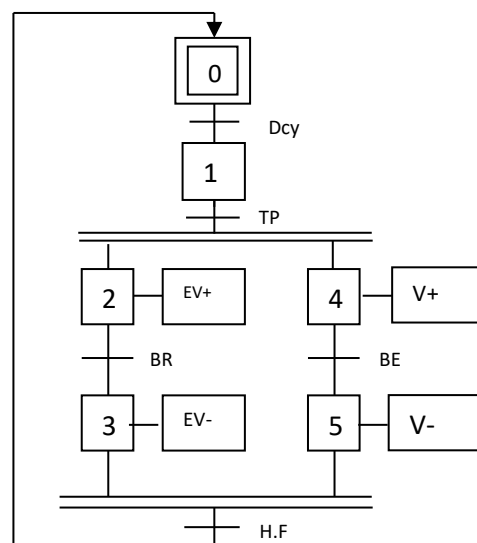


Figure 48 grafcet de la station de remplissage et de bouchage

Remarque :

On remarque que si on veut arrêter le système en fin de cycle, il suffit d'ajouter un bouton stop, mais par contre si on veut l'arrêter au milieu, on se rend compte que c'est impossible avec ce type de grafcet. Dans la partie suivante on présentera un guide qui prend en charge les modes marches et arrêts du système.

II.5. GEMMA

Le guide graphique GEMMA porte les « rectangles-états » dans lesquels seront exprimés les différents états de marches et d'arrêts pris par la machine.

En pratique, pour une machine donnée, on ne choisira parmi les états proposés par le guide que ceux qui sont nécessaires, et on précisera le nom de chacun des états retenus, à l'intérieur du rectangle-état correspondant.

Pour effectuer ce choix, il est nécessaire de bien comprendre la signification de chacun des états de marches et d'arrêts proposés par le guide graphique.

II.5.1. Procédures du GEMMA

II.5.1.1 Famille F « Procédure de fonctionnement » :

On regroupe tous les états du système automatisé qui sont indispensables à l'obtention de la valeur ajoutée (fabrication d'un produit, contrôle, stockage, teste, réglage, etc.).

La sous famille F1, F2, F3 assure la mise en/ou hors service de la production normale. L'autre famille F4, F5, F6 permet les essais et les vérifications avant ou après le passage en production normale.

- **F1 « Production normale »** : dans cet état le système produit normalement. C'est l'état pour lequel elle a été conçue. C'est à ce titre que le rectangle-état a un cadre particulièrement renforcé, On peut souvent faire correspondre à cet état un grafcet que l'on appelle grafcet de base.
- **F2 « Marche de préparation »** : cet état est utilisé pour les machines nécessitant une préparation préalable à la production normale : préchauffage, remplissage de la machine, etc.
- **F3 « Marche de clôture »** : c'est l'état nécessaire pour certaines machines d'atteindre une certaine position avant un arrêt prolongé : d'être vidées, nettoyées, etc., en fin de journée ou enfin de série.
- **F4 « Marche de vérification dans le désordre »** : Cet état permet la vérification dans le désordre des différents actionneurs du système automatisé sans respecter l'ordre de production normale.
- **F5 « Marche de vérification dans l'ordre »** : Cet état permet la vérification dans l'ordre de production normale des différents actionneurs du système automatisé. Cet état permet de faire évoluer le cycle de production normale tâche par tâche. Dans cet état la machine est en production ou hors production.

- **F6 « Marche de test »** : Les machine de contrôle, de mesure, de tri ..., comporte des capteurs qui doivent être réglés ou étalonnés périodiquement.

II.5.1.2 Famille A« Procédure d'arrêt » :

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

- **A1 « Arrêt dans l'état initial »** : C'est l'état dans lequel se trouve la machine avant de passer en production normale (état repos). Il correspond un peu à l'état initial d'un grafcet : c'est pourquoi, comme une étape initiale, ce rectangle-état est entouré d'un double cadre.
- **A2 « Arrêt demandé en fin de cycle »** : Cet état permet de conduire le système à un arrêt en fin d'un cycle de production. Le système va continuer de produire et s'arrêter lorsque le cycle de production sera terminé, cet état est utilisé lorsque l'on souhaite réalimenter en matière première un système par exemple. **A2** est donc un état transitoire vers l'état **A1**.
- **A3 « Arrêt demandé dans un état déterminé »** : Cet état permet de conduire le système à un arrêt en position autre que la fin de cycle, il permet par exemple d'arrêter le système dans un état permettant une intervention sur le système. C'est un état transitoire vers état **A4**.
- **A4 « Arrêt obtenu »** : le système est alors arrêté en une autre position que la fin de cycle.
- **A5 « Préparation pour remise en route après défaillance »** : c'est dans cet état que l'on procède à toutes les opérations (dégagement, nettoyages, ...) nécessaire à une mise en route après un à toutes les opérations nécessaires à une remise en route après défaillance.
Dans cet état l'opérateur intervient en général manuellement pour dégager, nettoyer ou vider le système.
- **A6 « Mise de la PO dans l'état initial »** : La machine étant en **A6**, on remet manuellement ou automatiquement la partie opérative en position pour un redémarrage dans l'état initiale.
- **A7 « Mise de la PO dans un état déterminé »** : La machine étant en **A7**, on remet la partie opérative en position pour un redémarrage dans une position autre que l'état initial.

II.5.1.3 Famille D « Les procédures en défaillances » :

Les procédures en défaillance définissent les états que devra avoir la partie opérative en cas de défaillance, ils sont au nombre de trois, ces rectangles d'état permettent de gérer les défaillances du système tel que par exemple l'arrêt d'urgence. Les états de défaillance sont les états de défaillance de la PO.

- **D1 « Arrêt d'urgence » :**

Cet état permet de gérer le système 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 éviter ou limiter les conséquences dues à la défaillance.

- **D2 « Diagnostic et ou traitement de défaillance » :**

Cet état permet à la maintenance de diagnostiquer l'origine de la défaillance et d'envisager le traitement approprié qui permettra redémarrage du système après traitement de la défaillance.

- **D3 « Production tout de même » :**

Il est parfois nécessaire de continuer la production après défaillance du système, on aura alors « un production dégradée », ou une « production forcée », ou une production aidée par des opérateurs non prévus en « production normale ».

La figure suivante représente tous les différents cas possibles qu'un système peut rencontrer, de la mise en route jusqu'à l'arrêt de fin de cycle, en passant par la vérification dans l'ordre ou dans le désordre (mode manuel) et également le mode réglage.

On retrouve également le comportement de la mise en route à partir de l'état initial, et de la remise en route après défaillance.

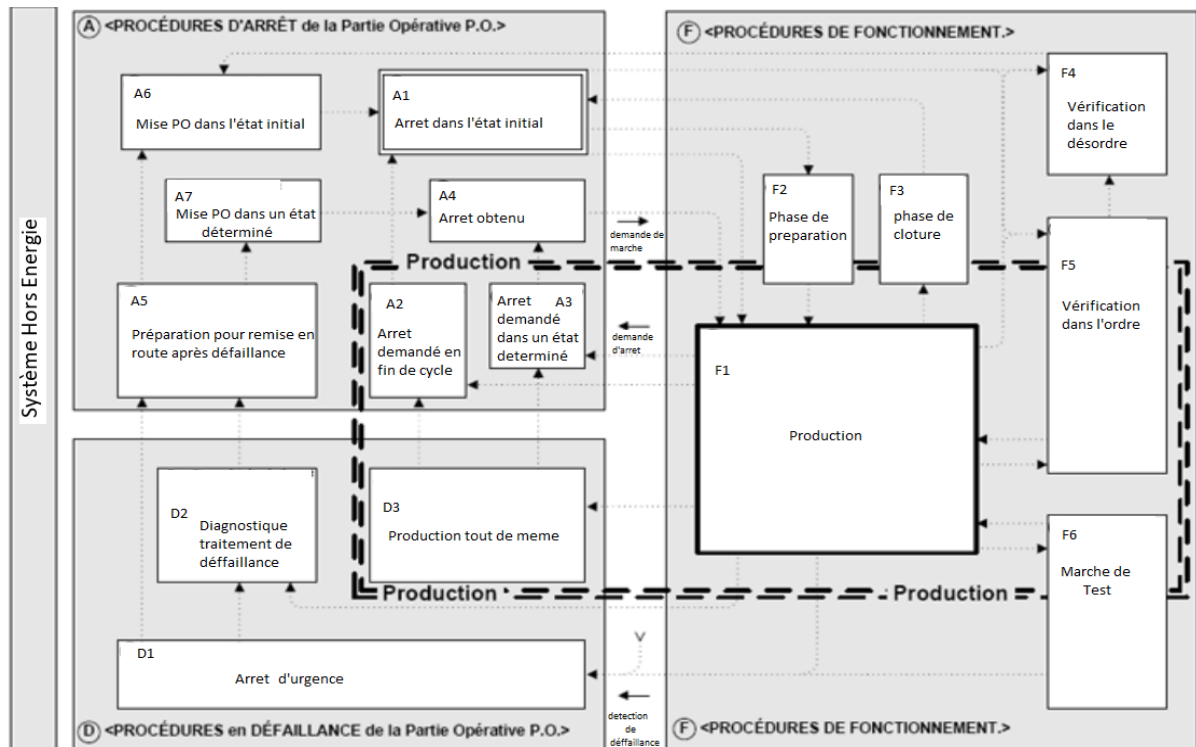


Figure 49 GEMMA

Les différents boutons qui gèrent les modes marche et arrêt :

- **Dcy** : Départ de cycle, c'est pour lancer la production
- **Acy** : Arrêt de fin de cycle, c'est pour arrêter le système après avoir terminé son cycle de production
- **MAN/AUT** : Sélecteur de mode manuel / Automatique, c'est pour sélectionner le mode qui sera utilisé, soit le mode manuel ou mode automatique.
- **AU** : Arrêt d'urgence, c'est pour arrêter le système en urgence, dans cette situation toutes étapes des grafjets seront désactivées uniquement les étapes initiales qui seront activées.
- **AuD** : Arrêt d'urgence doux, dans cette situation seulement les actions qui seront désactivés, les étapes resteront telles qu'elles le sont.
- **Réarm** : Réarmer le système, lorsqu'on a un arrêt d'urgence, avant de relancer la production, on doit réarmer le système.

II.5.2 Boucles opérationnelles de GEMMA :

On distingue au minimum trois boucles :

II.5.2.1 Boucle de réglage :

Cette boucle permet d'activer le mode manuel pour d'éventuelle vérification

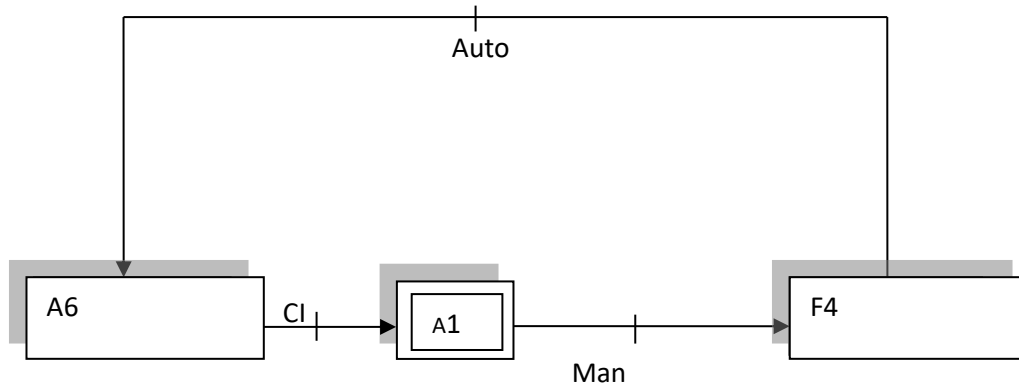


Figure 50 Boucle de réglage

À la fin de la vérification on doit passer par **A6** pour remettre le système à l'état initial.

II.5.2.2 Boucle de Fonctionnement

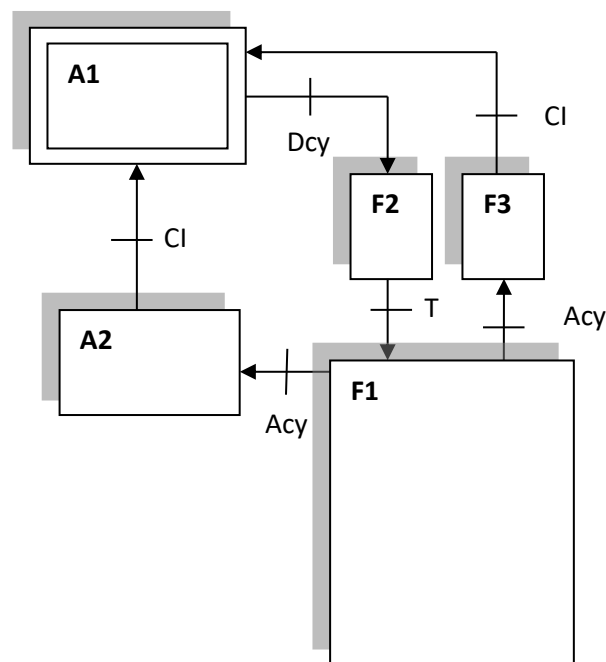


Figure 51 Boucle de fonctionnement

Cette boucle représente la production normale du système, qui prend toutes les situations de démarrage et d'arrêt normales.

II.5.2.3 Boucle de sécurité

Cette boucle gère la sécurité du système, son exécution est prioritaire sur les autres boucles.

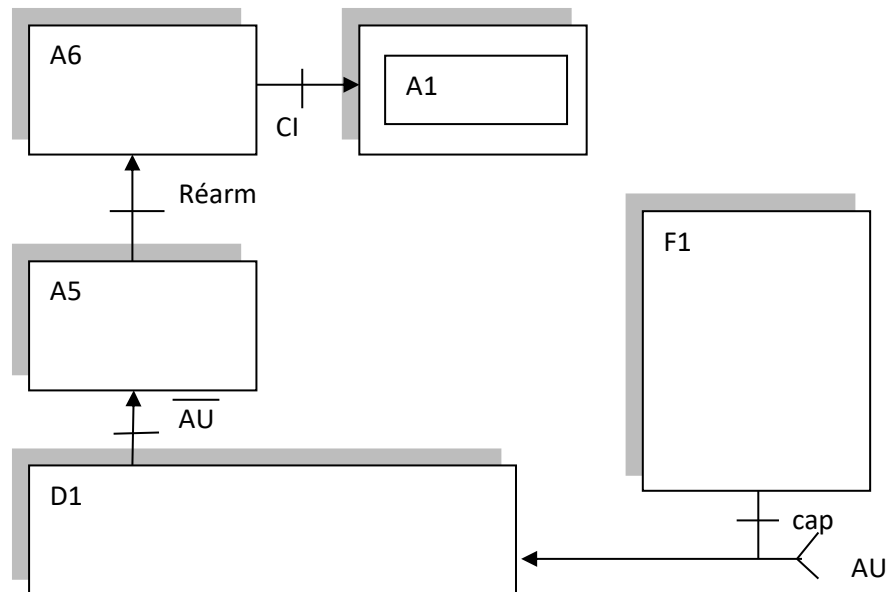


Figure 52 Boucle de défaillance

En convertissant ces trois boucles de GEMMA ; nous obtenons au minimum trois grafcet qu'on appellera : grafcet de sécurité ; grafcet de conduite et le grafcet de production normale.

II.5.3 Exemple :

Cahier des charges : après avoir donné l'ordre de départ (Dcy), le chariot part jusqu'au point **B** puis revient au point **C** et repart au point **B** et delà revient directement au point **A**.

Le pupitre est composé de :

Dcy: Depart de cycle

Acy: Arret de Cycle, contact normalement fermé

Au: Arret d'urgence , contact normalement fermé

Sélecteur: pour commutateur entre le mode manuel et automatique

km: pour tester le moteur du chariot dans le mode manuel

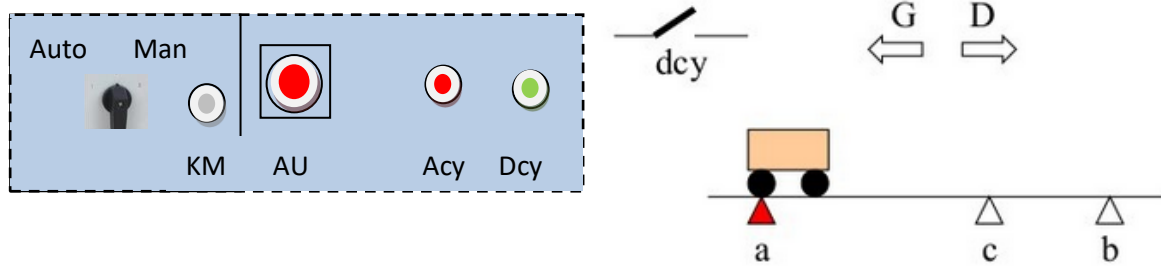


Figure 53 Chariot avec son panneau de commande du

Solution

Grafcet de sécurité

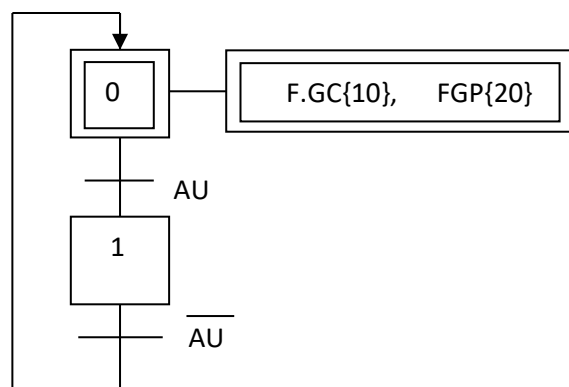


Figure 54 grafcet de sécurité du chariot

Grafcet de production

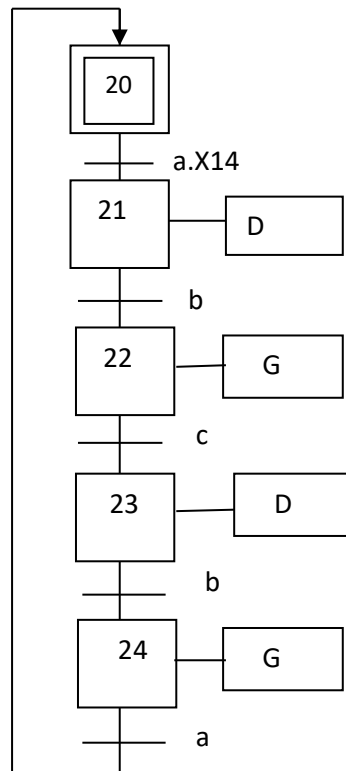


Figure 55 grafcet de production du chariot

Grafcet de conduite

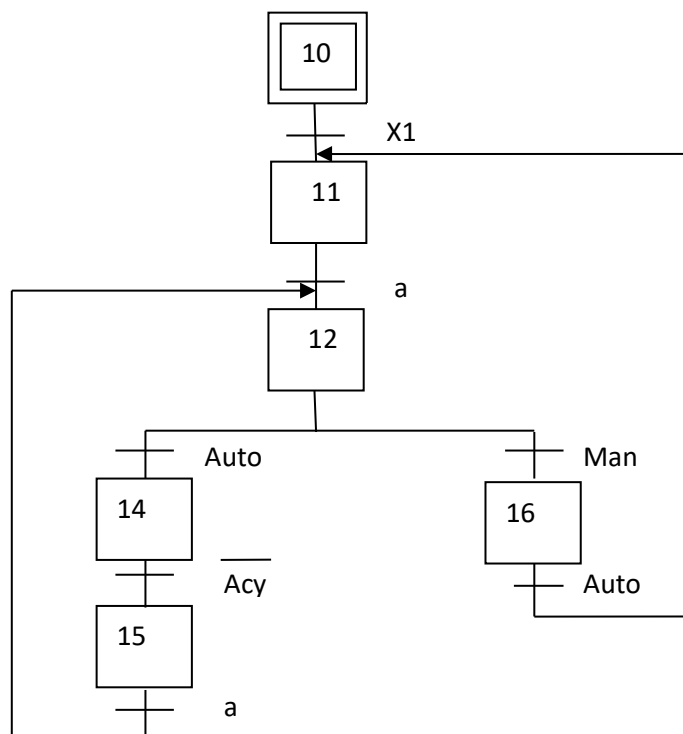


Figure 56 grafcet de conduite

II.6 Conclusion :

Dans ce chapitre nous avons présenté l'outil le plus important de l'automatisation des systèmes, le GRAFCET qui sert à la modélisation des comportements des automatismes.

Dans le chapitre suivant, on présentera l'architecture des automates programmables industriels, dans lequel on verra les constituants des automates, les différents types d'automates et les composants les plus importants de l'automate.

• Exercices de chapitre sur le GRAFCET

Exercice 01 :

Donner le Grafcet du système suivant :

Soit un portail coulissant à commander :

- Le portail étant fermé, le contact fin de course fcFE est actionné ;
- L'appuie sur le bouton-poussoir d'ouverture S3, le moteur actionne le portail et provoque son ouverture ;

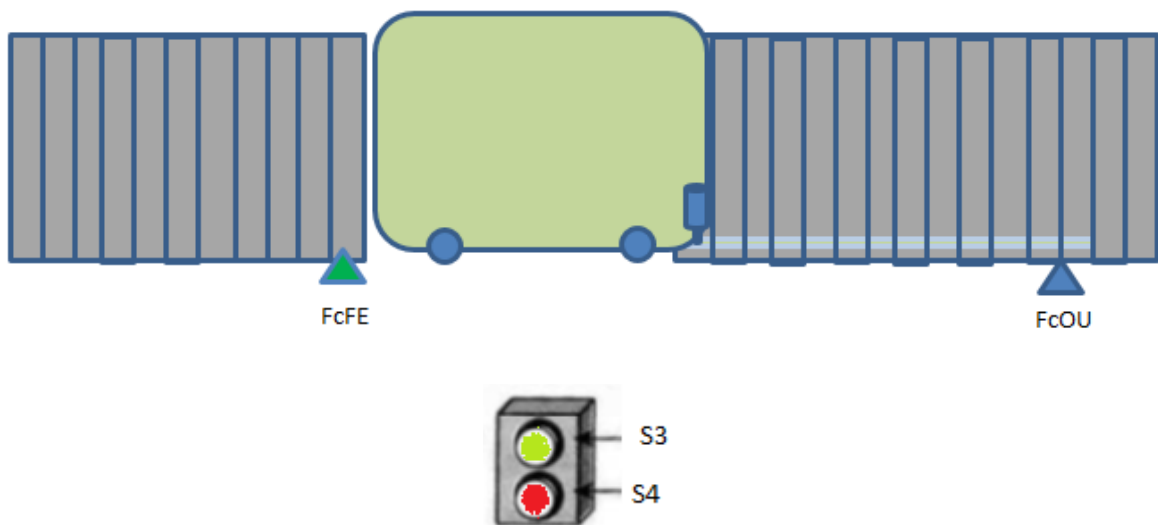


Figure 57 commande d'un portail

- En fin d'ouverture, le contact fin de course fcOU est actionné, il signale l'ouverture du portail, et il coupe l'alimentation du moteur.
- L'action sur le bouton-poussoir de fermeture S4 provoque l'inversion de sens de marche du moteur, et la fermeture du portail.
- Le portail libère le contact fcOU, et se déplace jusqu'à actionner le contact fcFE qui provoque l'arrêt du moteur.

Exercice : 02

Donner le Grafcet du système suivant :

On a deux vérins A, B, un bouton poussoir BP et un capteur cp. Après avoir appuyé sur le Bp et que la pièce est présente, le vérin B sort et ramène la pièce grâce à un aimant et revient à sa position initiale, puis le vérin A prend la pièce et l'évacue et revient à sa position initiale.

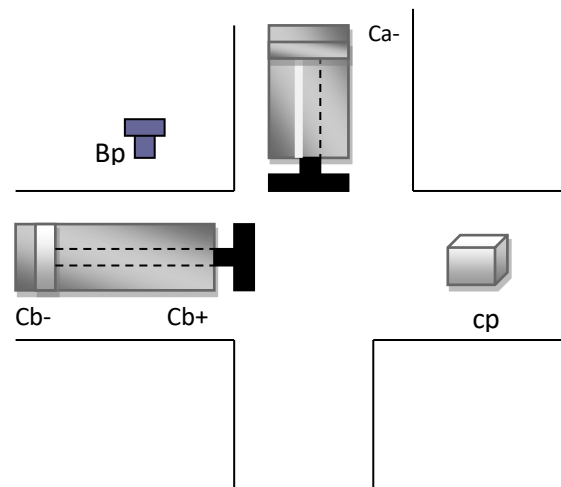


Figure 58 Commande de deux vérins

Exercice 03 :

Donner le Grafcet du système suivant :

On a deux vérins A, B et un bouton poussoir BP.

Après avoir appuyé sur le Bp, le vérin A pousse les pièces pour les mettre dans une boîte, puis revient à sa position, initiale.

Le vérin B évacue la boîte après qu'elle soit remplie de 10 pièces et revient à sa position initiale.

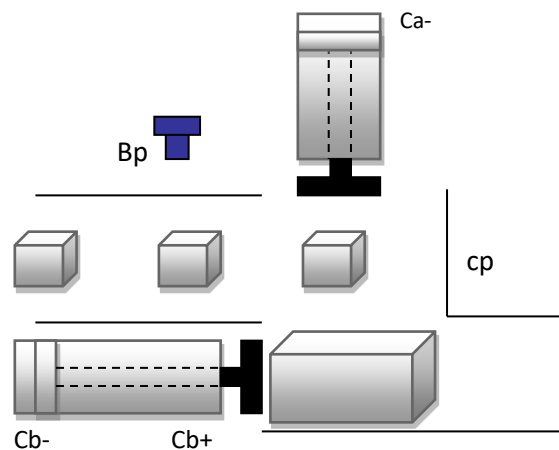


Figure 59 Remplissage d'une caisse de 10 pièces

Exercice 04 :

Donner le Grafcet du système suivant :

Soit une installation de traitement de surface, deux chariots peuvent être actionnés en montée ou en descente de façon à saisir des bacs. Les chariots peuvent être déplacés vers la droite ou vers la gauche. Le but de la manipulation consiste à saisir et à tremper ces bacs dans des containers.

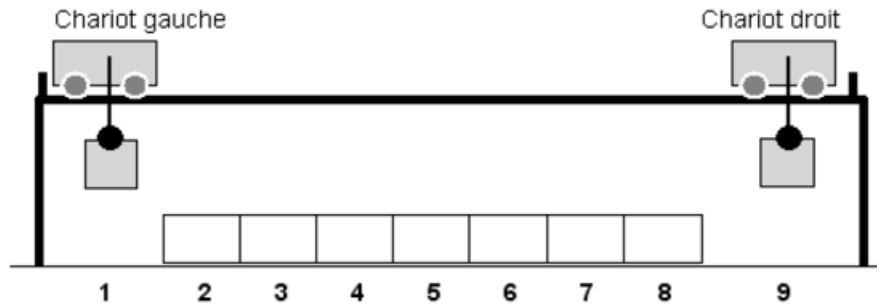


Figure 60 Système de traitement de surface

Donner les grafjets de niveaux 1 et 2 permettant après action sur interrupteur départ de cycle de tremper le bac de gauche dans les containers 4, 3, 2, retour en 1, et simultanément, le bac de droite dans les containers 6, 7, 8, retour en 9. A l'issue de ce traitement, les chariots devront être en position haute.

Exercices 05 :

Donner le Grafjet du système suivant :

Un tapis avance pas à pas et transporte des bidons vides qui seront d'abord remplis puis ensuite bouchés à des postes de travail différents.

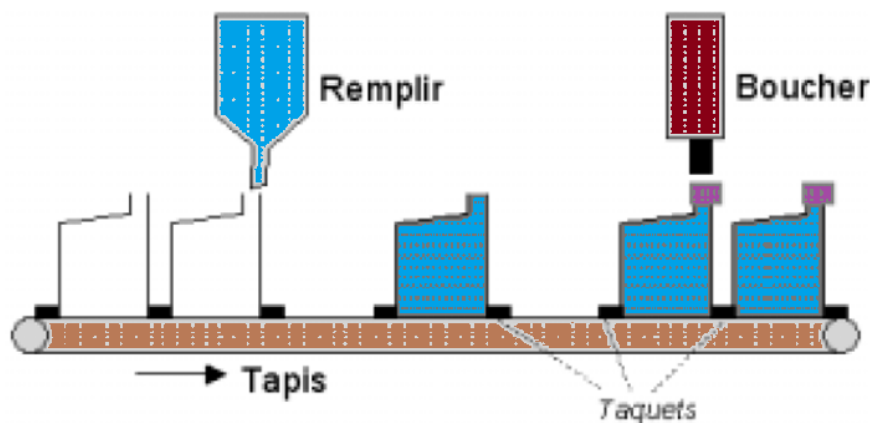


Figure 61 Station de remplissage et de bouchage

L'approvisionnement en bidons n'est pas régulier et certains bidons peuvent manquer de temps à autres. La distance entre bidons présents est fixée par des taquets situés sur un tapis et distants d'un pas.

Un dispositif permet à chacun des postes décrets de détecter la présence ou l'absence d'un bidon.

Capteur :

Cr : capteur de présence en poste remplissage

Cb : capteur de présence en poste bouchage

Cb+: capteur de bouchage

Le remplissage et le bouchage est instantané

Actions :

R : remplissage

B : bouchage

Km : lancer tapis

Exercice 06 :

Un tapis roulant ramène des pièces jusqu'à la perceuse, si une pièce est présente la perceuse effectue le perçage et remonte à sa position. La récupération s'effectue dans une caisse de capacité de 20 pièces qui est évacuée et remplacée par une autre caisse à chaque remplissage.

Remarque :

- L'acheminement de pièces n'est pas régulier.
- Le remplacement de la caisse n'est pas instantané.

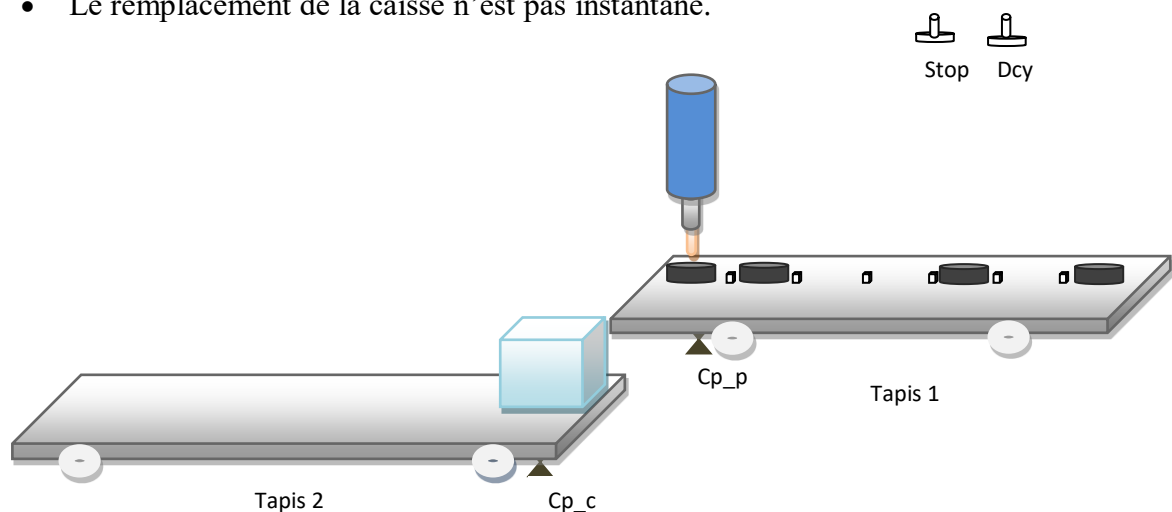


Figure 62 Station de perçage de pièces

Capteurs:

Cp_p : Capteur de présence de pièce ; **Cp_c** : Capteur de présence de caisse ; **Dcy** : Départ de cycle ; **C0** : Compteur ; **b** : niveau bas de la perceuse ; **h** : niveau haut de la perceuse
op : signal donné par l'opérateur après avoir changé la caisse

Actionneurs :

KM1 : Avance tapis 1 ; **KM2** : Avance tapis 2

D : Descente perceuse ; **M** : Montée perceuse

Km : moteur de la mèche

- Donner le **Grafcet** du système,

Exercice 07

Soit le grafcet suivant :

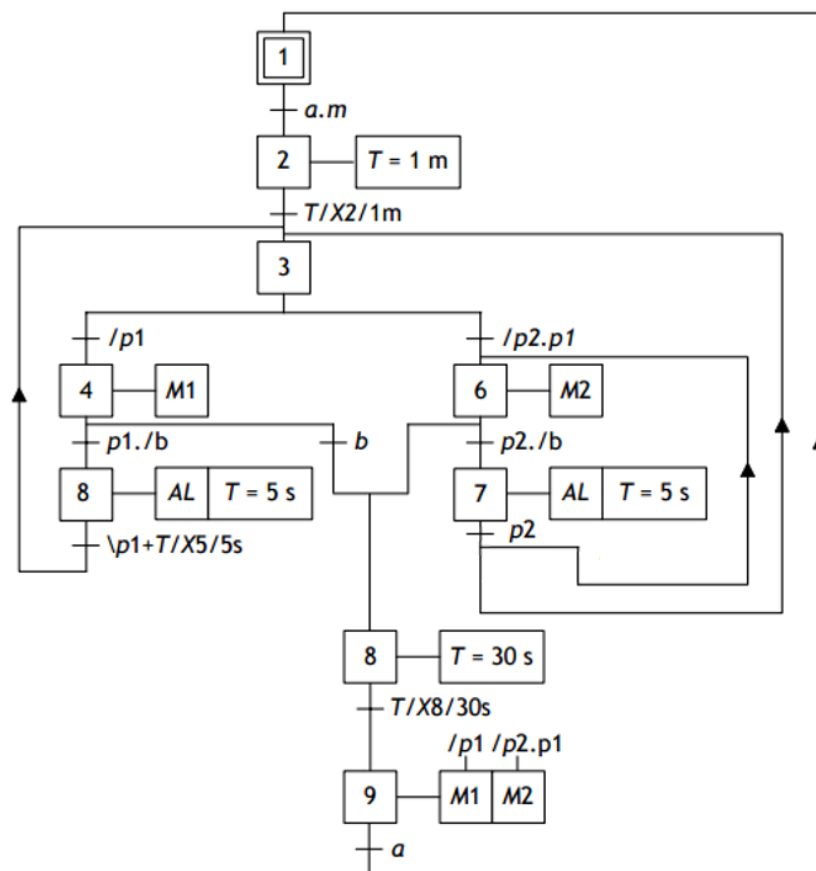


Figure 63 grafcet à corriger

- Trouver les erreurs de ce grafcet

Exercice 08 :

Deux chariots transportent de la marchandise de point **A** vers le point **B**.

Le départ des chariots vers le déchargement est lancé par **Dcy** et le retour vers le chargement est lancé par les boutons **Bp1** et **Bp2**. Le chargement et le déchargement se font manuellement.

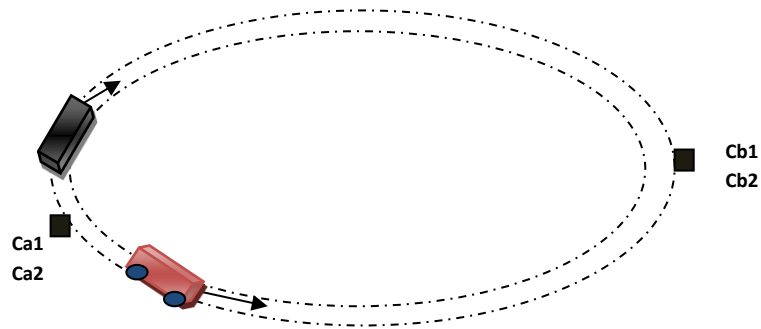


Figure 64 chargement - déchargement par chariots

Capteurs

Ca1/Cb1 : capteur de présence chariot 1 au point A / point B

Ca2/ Cb2: capteur de présence chariot 2 au point A / point B

Actionneurs

Km11/Km12 pour lancer le **chariot 1** respectivement aller / retour

Km21/Km22 pour lancer le **chariot 2** respectivement aller / retour

Question :

-Donner le grafcet de production

• **Solution**

Exercice 01 :

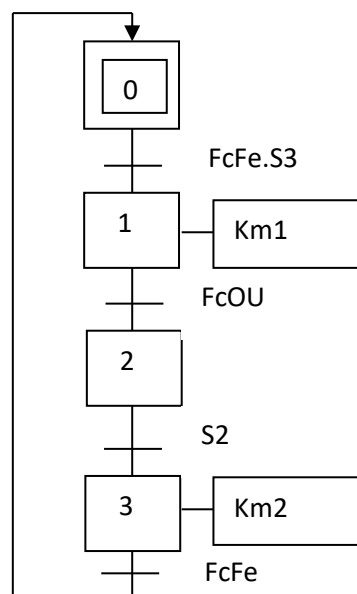
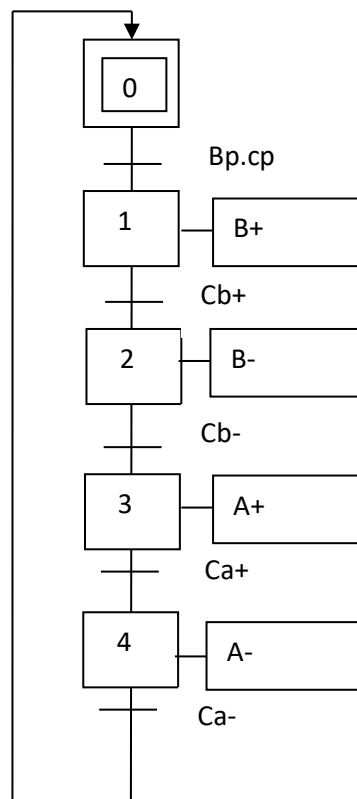
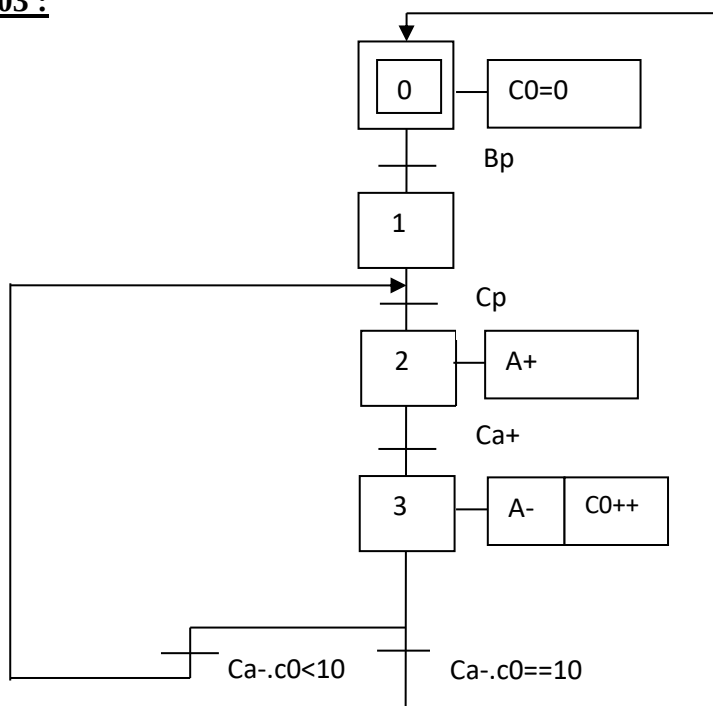
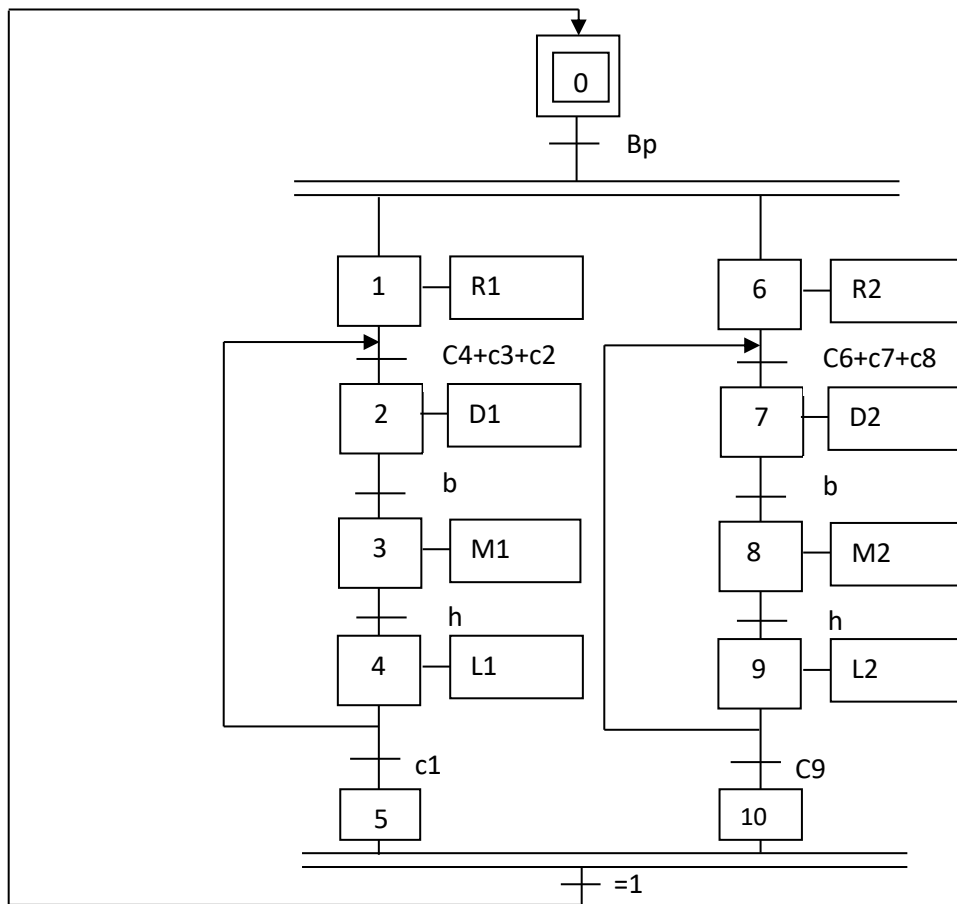
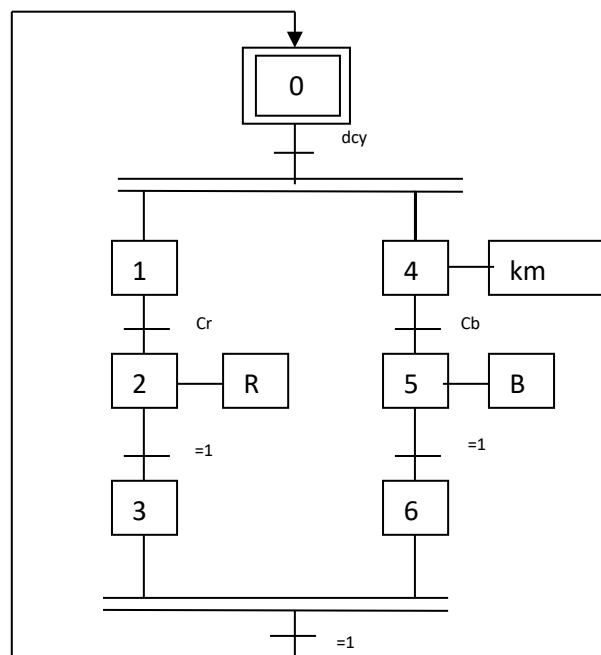


Figure 65 grafcet de portail

Exercice 02 :**Figure 66** grafcet de commande de vérins**Exercice 03 :****Figure 67** grafcet de remplissage de caisse de 10 pièces

Exercice 04 :**Figure 68** grafcet de la station de traitement de surface**Exercice 05 :****Figure 69** grafcet de la station de remplissage et de bouchage

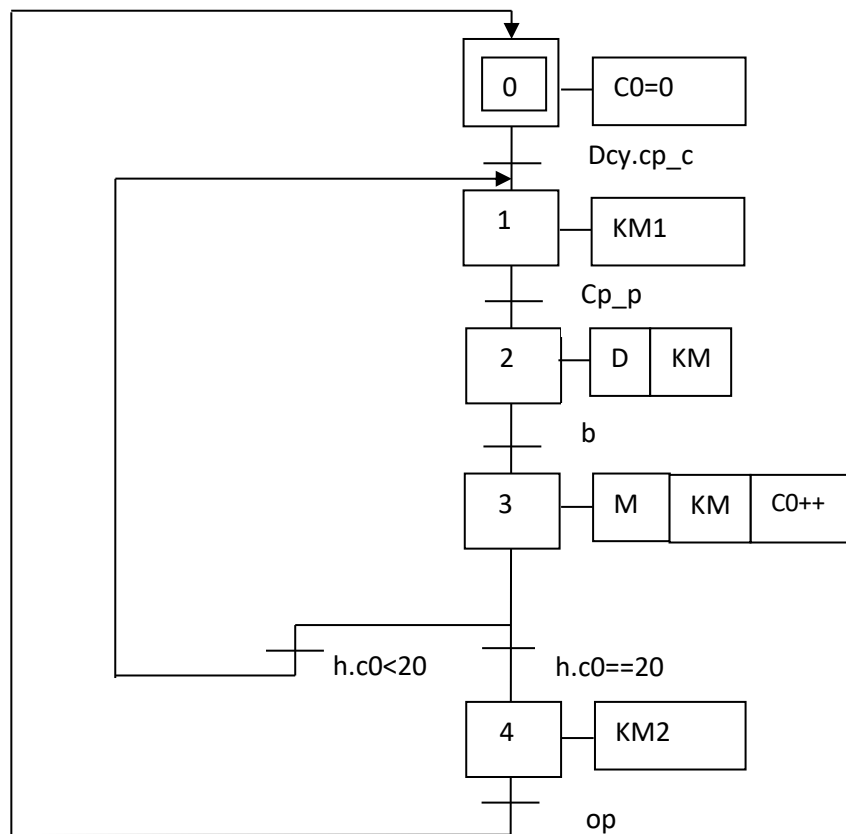
Exercice 06 :

Figure 70 Station de perçage de pièces

Exercice 07 :

Donne ce grafcet on a trois erreurs syntaxiques qui sont encadrées par des cercles rouges :

1. Il manque une transition, car c'est une convergence en « **ou** »
2. Ce numéro d'étape est déjà utilisé ; dans un grafcet on ne doit pas trouver deux ou plusieurs étapes avec un même numéro.
3. Il manque une transition, car c'est une divergence en « **ou** »

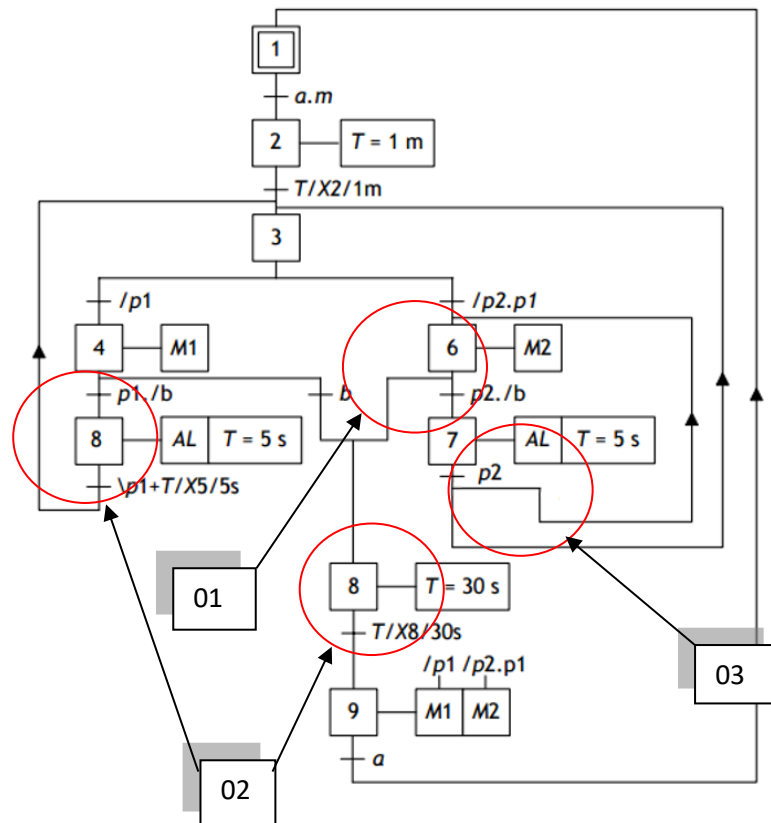


Figure 71 grafcet corrigé

Exercice 08 :

Dans ce grafctet on ajoutera deux étapes pour le déchargement et deux autres à la fin pour la synchronisation.

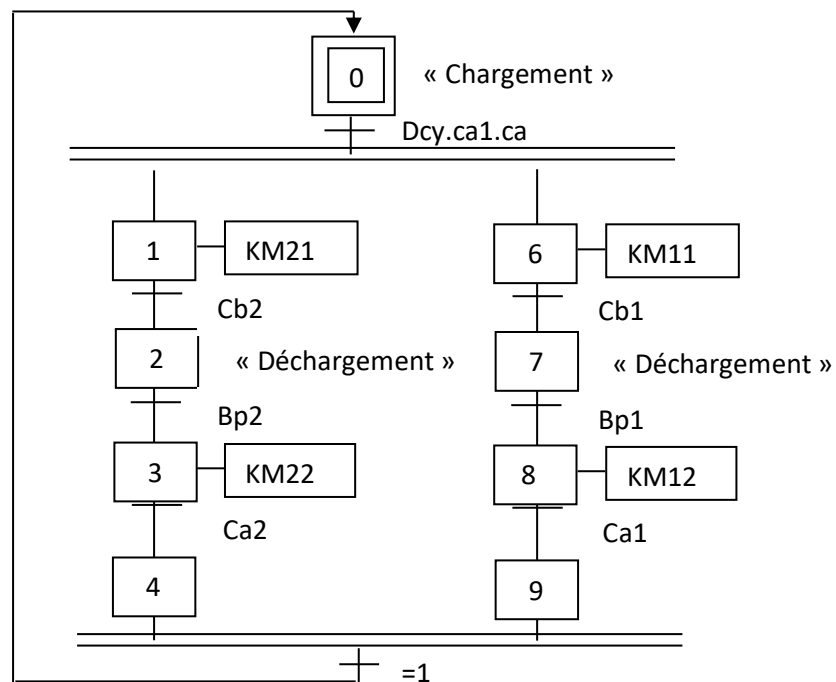


Figure 72 chargement - déchargement de deux chariots

• Exercices sur le GRAFCET et le GEMMA

Exercice 01 :

Après le chargement, on lance le départ de cycle, le chariot prend toujours le sens **(1)**, il prend le sens **(2)** si et seulement si le sens **(1)** est occupé et ce, valable pour l'allée et pour le retour, le retour est lancé par un bouton poussoir Bp. Le sens (2) est toujours libre, le chargement et le déchargement sont manuels.

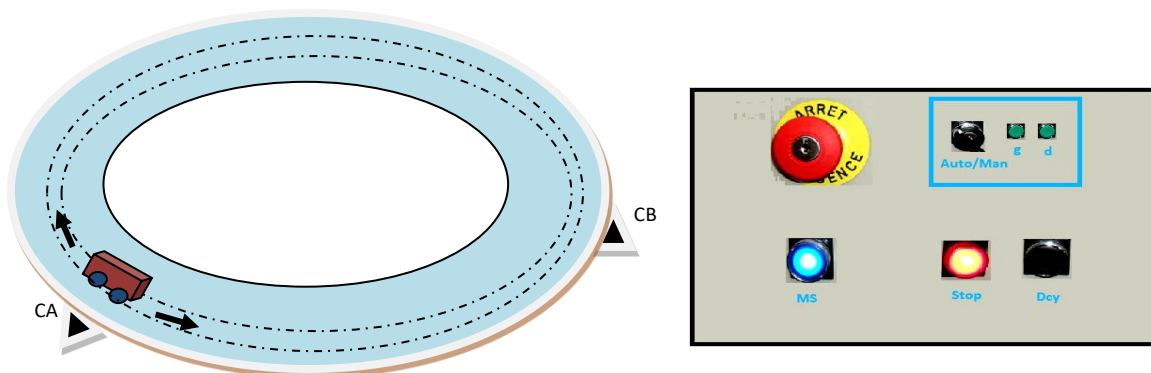


Figure 73 chargement et déchargement de chariot avec pupitre de commande

Capteurs /Actions

Dcy : départ de cycle. **Bp** : lancer le retour

Stop : arrêt de fin de cycle

Km1 : pour lancer chariot sens(1)

Km2 : pour lancer chariot sens(2)

d : lancer le chariot sens (1) dans le mode manuel

g : lancer le chariot sens (2) dans le mode manuel

Au : Arrêt d'urgence ; **MS** : mise sous tension

CR =1: chariot déraillé, **CR =0**: chariot sur les rails

CA/CB : capteur de position point A/B, **S1 =0**: sens occupé, **S1 =1**: sens libre

Auto/Man : sélecteur de mode manuel /automatique, (peut être demandé à l'arrêt ou en production, et qui s'active uniquement si le grafcet de production est à l'étape initiale)

Question :

- Donner les trois grafkets : Sécurité ; Production normale ; Conduite

Exercice 02 :

Lorsque les conditions de sécurité (« **Td** » et « **Au** ») sont vérifiées, et on appuis sur le « **Dcy** », le tapis se mis en route et envoie des pièces dans le Four. Si la température « **Td** » ou l'arrêt d'urgence « **Au** » est activé le système s'arrête immédiatement. La gestion de la température du four est assurée par un autre dispositif. Lors de la mise ou de remise en route le tapis se lance pendant 5s pour évacuer les pièces bloquées.

Si un arrêt de fin de cycle est demandé, le système S'arrêtera 5s après.

Capteurs

Dcy : départ de cycle ; **Stop** : arrêt de fin de cycle ; **d** : lancer le tapis dans le mode manuel ;
Au =0: Arrêt d'urgence ; **MS** : mise sous tension ; **Td** =1: Température dangereuse ;
Auto/Man : sélecteur de mode manuel /automatique.

Actionneurs

Km : pour lancer le tapis (pièce vers le four)

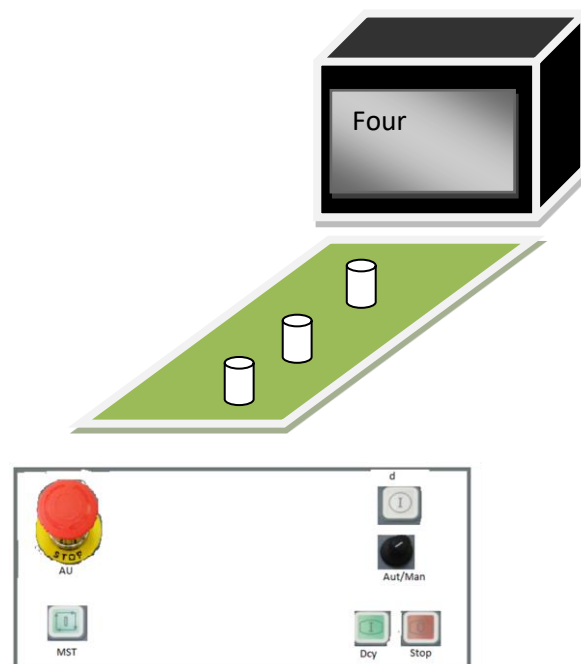
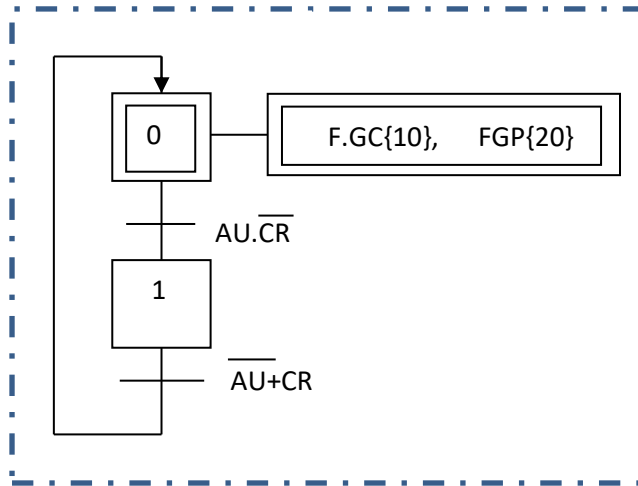
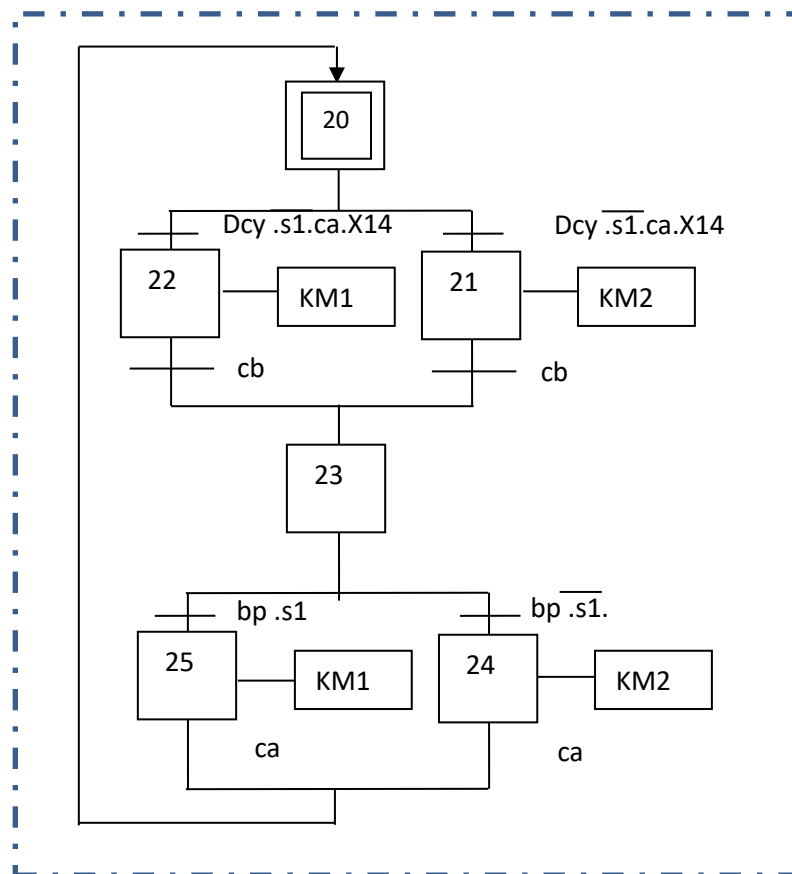


Figure 74 Four avec son pupitre de commande

Questions :

- Donner les trois grafkets : Sécurité, Production normale, Conduite
- Donner le programme Ladder

- **Solution**

Exercice 01 :**Figure 75** grafcet de sécurité**Figure 76** grafcet de production

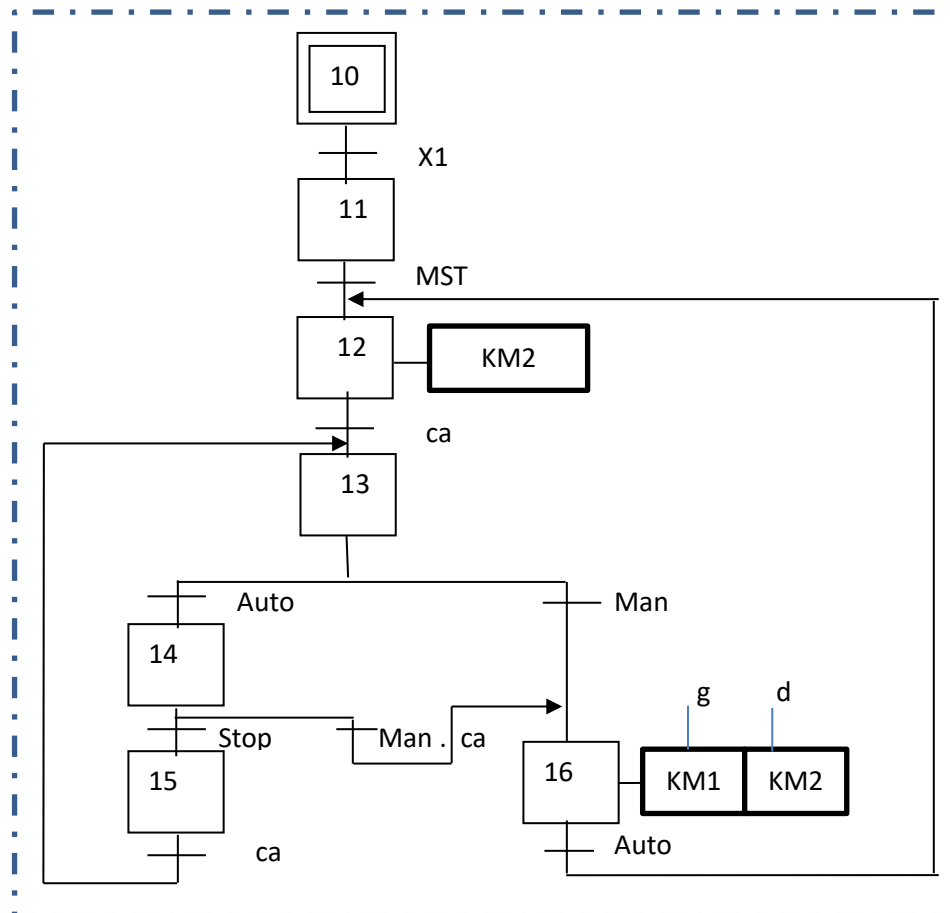


Figure 77 grafcet de conduite

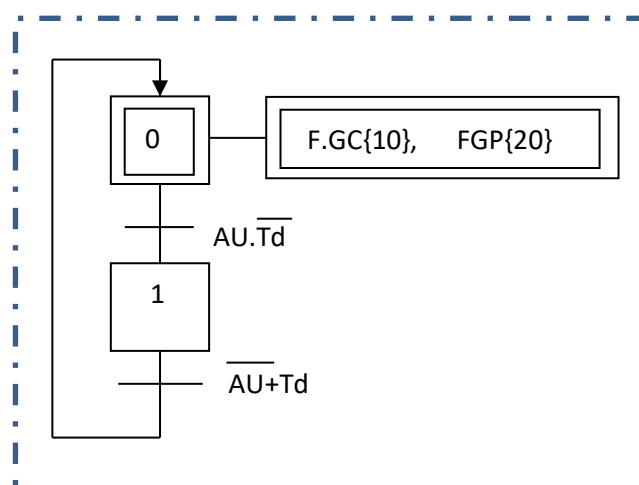
Exercice 02 :

Figure 78 grafcet de sécurité

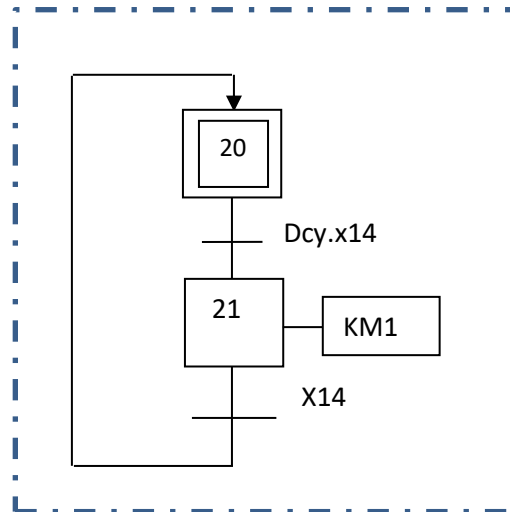


Figure 79 Grafcet de Production

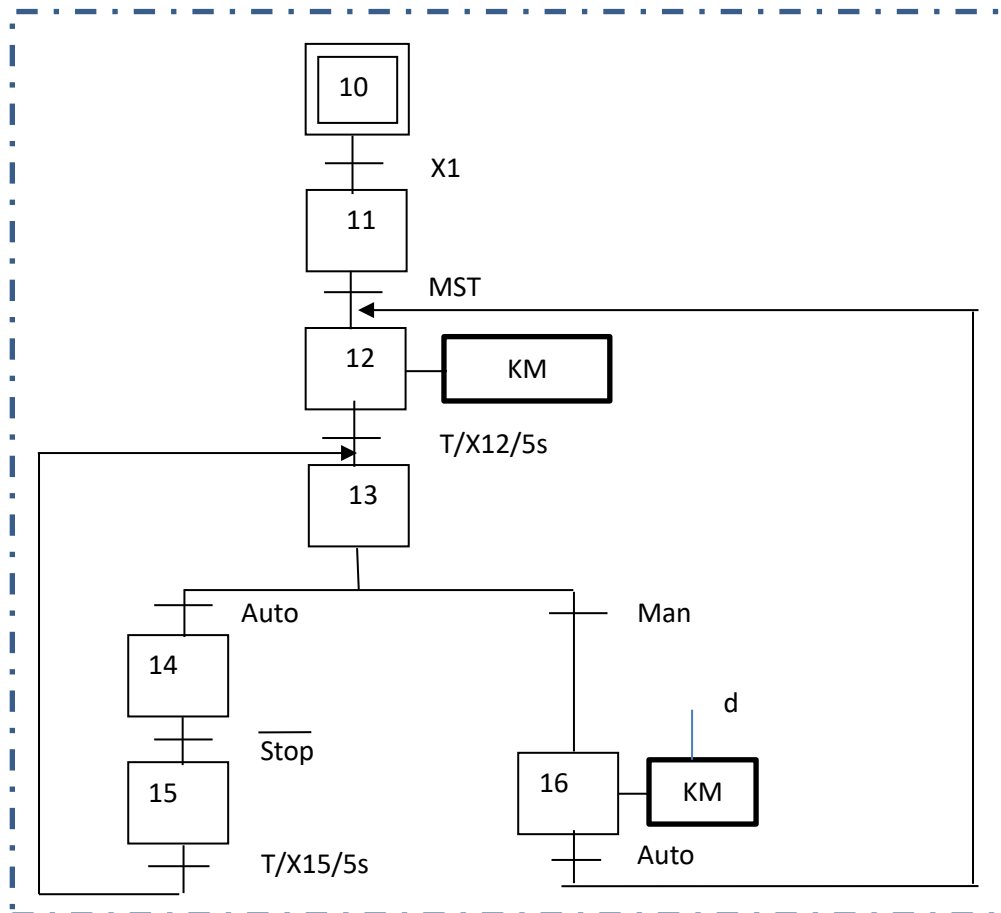


Figure 80 Grafcet de conduite

Chapitre III

Automates Programmables Industriels

III.1 Introduction

Les Automates Programmables Industriels, également connus sous l'acronyme API, sont des dispositifs électroniques largement utilisés dans le domaine de l'automatisation industrielle. Ces systèmes jouent un rôle essentiel dans le contrôle et la supervision des processus de fabrication et d'assemblage dans divers secteurs tels que l'industrie manufacturière, l'automobile, la chimie, l'énergie, et bien d'autres.

Un API est conçu pour effectuer des tâches spécifiques en réponse à des signaux d'entrée, souvent issus de capteurs qui surveillent divers paramètres dans un environnement industriel. Ces signaux d'entrée sont ensuite traités par le programme logique intégré dans l'API, qui prend des décisions en fonction des conditions prédéfinies. Les actions de sortie générées par l'automate peuvent inclure le contrôle de moteurs, de vannes, de robots, ou d'autres équipements industriels.

III.2 Qu'est-ce qu'un automate programmable

Un automate programmable industriel est un appareil électronique programmable qui commande des systèmes industriels grâce à un programme sauvegardé dans sa mémoire et aux informations issues des capteurs sur l'état de ce système. Il réalise la commande grâce aux ordres qu'ils envoient vers les actionneurs à travers des modules de sorties et aux informations récupérées sur l'état du système à travers les modules d'entrées, la figure suivante représente un automate de marque SIEMENS et type S7-300.



Figure 81 Automate programmable de type S7-300

III.3 Les différents types d'automates

On retrouve dans le marché plusieurs types d'automate, ils sont en deux catégories, les automates compacts et modulaires.

III.3.1 Automates compacts

Sont des automates programmables qui ont des caractéristiques fixes et qui n'acceptent pas d'extension, c'est-à-dire, le nombre d'entrées/sorties est fixe et ne peut être augmenter.

Exemple :



Mitsubishi



Emerson



Ascon technologie



Schneider

Siemens

Crouzet

Figure 82 Automates programmables Compact

III.3.2 Automates modulaires

Sont des automates programmables qui ont des caractéristiques d'origines et qui peuvent subir des extensions en ajoutant des modules supplémentaires, en effet dans le cas où on veut faire une extension à notre système, le nombre de capteur et d'actionneur va augmenter, ce qui va créer un manque dans les entrées / sorties de l'automate, alors la solution existe avec ces types d'automates, il suffit juste d'ajouter des modules supplémentaires.

Exemples :



Figure 83 Automate modulaire Rockwell



Figure 84 Automate programmable modulaire SIEMENS

III.4 Les éléments constitutifs des automates

Un automate programmable est constitué de plusieurs éléments qui contribuent à sa fonction, tel qu'une alimentation pour l'alimenter en Energie ... etc., la figure suivante résume presque tous les éléments dont se compose un automate programmable.

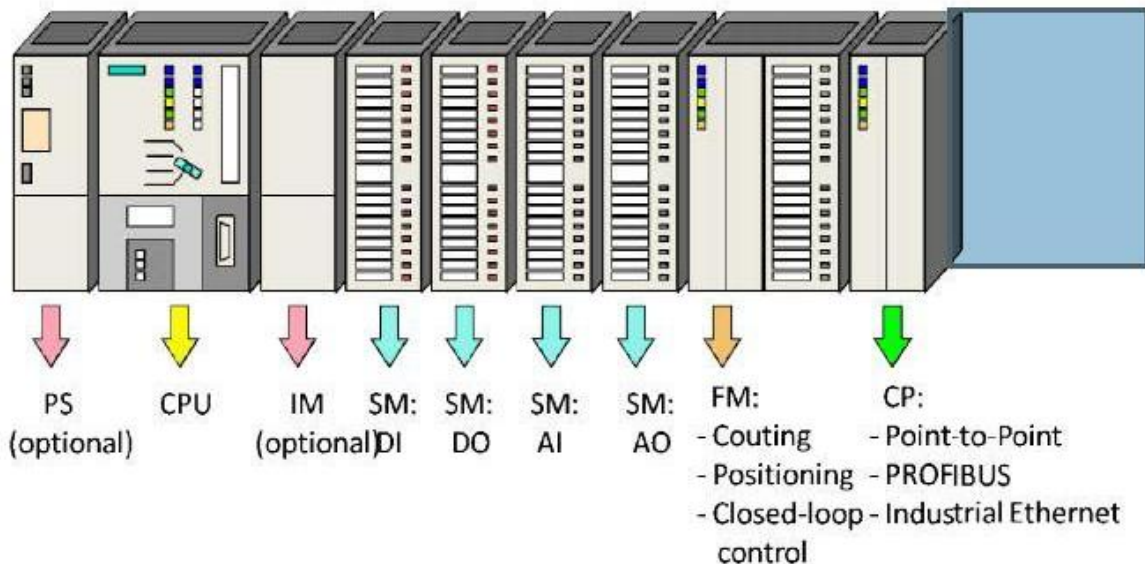


Figure 85 Différents modules de l'automate modulaire

III.4.1 Rack

Le rack sert de colonne vertébrale à l'automate modulaire, il maintient tous les modules ensemble, tels que le CPU, le module d'alimentation, le module de communication, les modules d'entrée-sortie, etc. Le rack synchronise à la fois l'alimentation et le signal de communication avec tous les modules.



Figure 86 Rack

III.4.2 Alimentation (PS)

Cette partie assure l'alimentation en énergie électrique du CPU et tous les autres modules de l'automate.



Figure 87 Module d'alimentation pour la gamme SIEMENS

III.4.3 Unité de contrôle (CPU)

C'est dans partie qu'on sauvegarde le programme et c'est partie qui assure le traitement de toutes les données qui arrivent à l'automate.



Figure 88 CPU 1500 / 1200 pour la gamme SIEMENS

III.4.4 Module d'interface (IM)

C'est un module qui assure la communication entre l'automate avec d'autres modules sur un autre rack.



Figure 89 Module d'interface

III.4.5 Modules Entrées/Sorties (SM)

Sont des modules d'interface avec les pré-actionneurs et les capteurs, en effet à travers les modules sorties, l'automate envoie des ordres vers les pré-actionneurs et à travers les modules entrées, il récupère les informations des capteurs sur l'état du système.



Figure 90 Module Entrées/Sortie (SIEMENS)

III.4.6 Modules de fonctions (FM)

Un module de fonction est un module qui contient un programme prédéfini qui fournit une fonction ou un ensemble de fonctions spécifique à utiliser dans les programmes API. Les modules de fonction sont utilisés pour simplifier la programmation d'une logique de contrôle complexe et pour réduire le temps et les efforts nécessaires au développement d'un programmes API.



Figure 91 Module FM de commande de position

III.4.7 Modules de couplage (CP)

Sont utilisés pour la communication entre automates ou avec d'autres appareils industriels telle qu'une HMI, SCADA...etc.



Figure 92 Module de communication

III.5 Les critères de choix d'un automate

Le choix de l'automate se base sur les critères suivants :

- Le nombre d'entrées / sorties disponibles
- Le type de processeur : la taille mémoire, la vitesse de traitement et les fonctions spéciales offertes par le processeur
- La disponibilité des modules spéciaux
- La possibilité de communication de l'automate avec des standards normalisés (ProfiBus).

III.6 Différents types de données API

Toutes les données traitées par l'automate que ce soit des variables, des constantes des chaînes de caractères doivent être représentées par un type donné ; les types les plus utilisés sont :

III.5.1 Booléen (Bool) :

Exprime des données qui ont deux valeurs soit vrai équivalent à 1 logique soit faux équivalent à 0 logique. Ce type est généralement associé aux capteurs et aux pré-actionneurs tout ou rien (TOR), elle occupe 1 bit dans la mémoire.

III.6.2 Integer (INT):

Exprime des données qui varie entre -32768 et 32767, il occupe 16 bits (2 octets) il est généralement utilisé pour les capteurs analogiques.

III.6.3 Double Integer (DINT)

Ce sont des nombres décimaux signé entre -2147483647 et +2147483647, Il est exprimé dans l'une des bases suivantes : décimale, hexadécimale, octale ou binaire.

III.6.4 Real :

Il occupe 32 bits (4 octets) il prend des valeurs entre -3.4028235E+38 to 3.4028235E+38

III.6.5 Temporisation :

TIME elle ne peut jamais être négative et commence T#24D_20H_31M_23S_648MS à T#24D_20H_31M_23S_647MS.

III.6.6 Chaîne : Elle doit être précédée et suivie par une apostrophe, et ne doit jamais excéder 255 caractères Le caractère spécial ('\$') est utilisée pour insérer des caractères non imprimables.

III.7 Câblage de l'API aux différentes E/S

Pour donner des ordres aux distributeurs et récupérer les informations des capteur, l'automate doit avoir une carte d'entrée et une carte de sortie.

III.7.1 Carte d'entrée

C'est l'interface entre l'unité de traitement et les capteurs, elle contient plusieurs adresses sous forme de bits, chaque capteur ou bouton sera branché à une entrée, elle permet de récupérer

le signal qui est véhiculé par le capteur et le transforme en un type de donnée exploitable par l'unité de traitement.

- **Branchement**

Les entrées doivent être alimentées en tension via les signaux d'entrée, pour cela, on branche la phase de l'alimentation avec le capteur et le neutre vers le point commun de la carte.

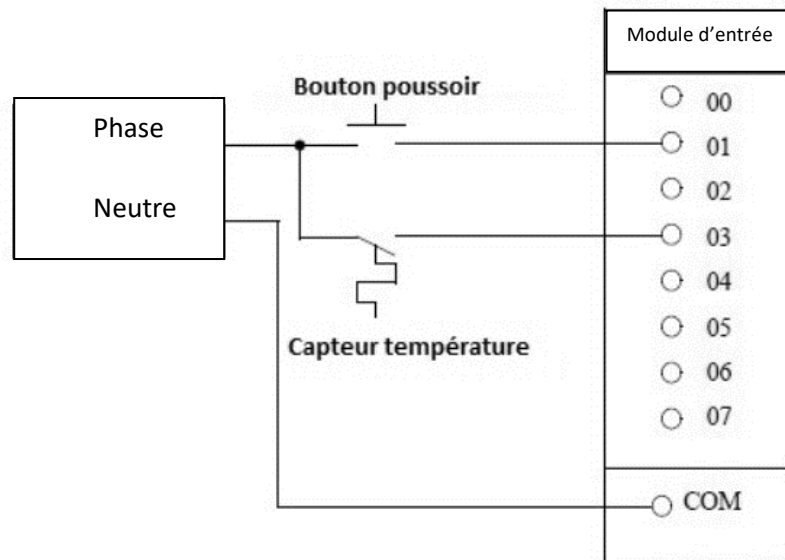


Figure 93 Exemple câblage de la carte d'entre (TOR)

III.7.2 Carte de sortie

C'est l'interface entre l'unité de traitement et les pré-actionneurs, elle contient plusieurs adresses sous forme de bits, chaque pré-actionneur sera branché à une sortie, elle permet de véhiculer l'ordre vers le pré-actionneur.

- **Branchement**

La tension de sortie est assurée par l'automate, donc la phase sera dans le signal de sortie qui va attaquer le pré-actionneur, il faut juste brancher le neutre vers le point commun de la carte.

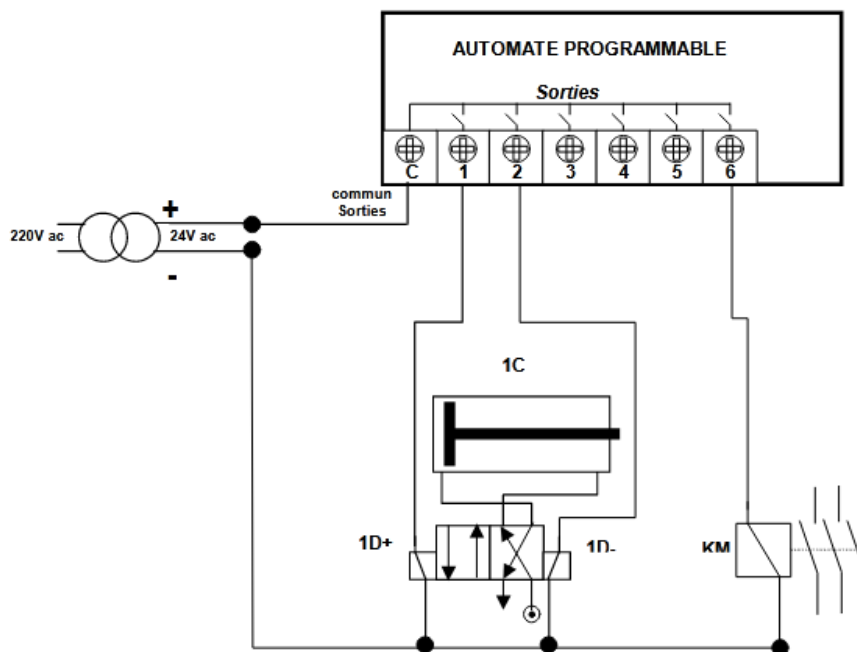


Figure 94 Exemple du câblage de la carte sortie

III. 8 Cartes d'entrées / sorties analogiques

Pour lire les valeurs des capteurs analogiques tel que la sonde de niveau et commander des actionneurs analogiques tel que le pourcentage d'ouverture d'une, il faut une interface qui est capable de lire et de sortir ce type de signal.

III.8.1 Type de signaux analogiques :

Signal d'entrée / sortie peut être une tension ou un courant, les cartes de l'automates sont configurables selon le cas, et ce signal peut être de plusieurs valeurs comment suivant :

- En tension : 0..10v, 2..10v, ...
- En courant : 0..20mA, 4..20mA , ...

III.8.2 Cartes d'entrée analogique

La carte d'entrée analogique convertit le signal d'entrée en une plage de nombre de 0 à 27648, et grâce à quelque fonction telle que « scale » on fait une normalisation pour exploiter le signal.

- **Branchement**

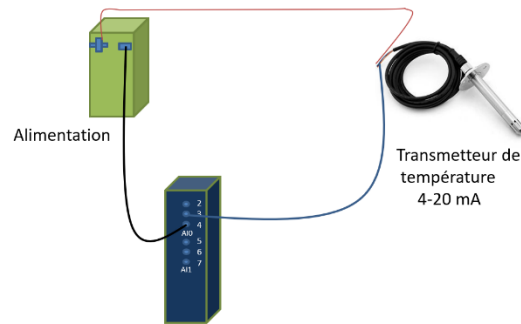


Figure 95 Câblage entrée analogique en courant

III.8.3 Cartes de sortie analogique

La carte de sortie analogique convertit la variable contenue dans la plage de nombre de 0 à 27648 à un signal électrique correspondant puis ce signal sera envoyé vers l'actionneur analogique.

- **Branchement**

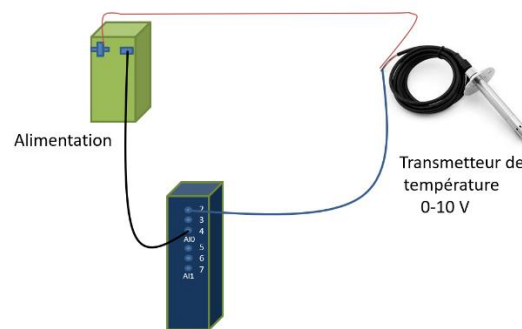


Figure 96 Câblage de la carte de sortie analogique

III.12 Conclusion

Dans ce chapitre nous avons présenté toutes les notions de base sur le fonctionnement de l'automate et de ses composants, dans le prochain chapitre nous allons présenter les techniques de modélisation des cahiers des charges ainsi que la procédure de programmation des automates programmables.

- **Exercices du chapitre**

Exercice 01 :

Réaliser le branchement de l'automate avec deux distributeurs 4/2 pneumatiques et un contacteur afin de réaliser la commande automatique suivante.

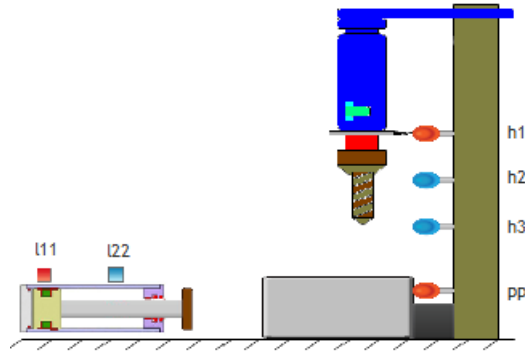


Figure 97 Perceuse industrielle

Exercice 02

Identifier les modules suivants :



Composant n°01



Composant n°02



Composant n°06

Figure 98 Composants d'automates programmables (partie 01)



Composant n°03

Composant n°04

Composant n°05

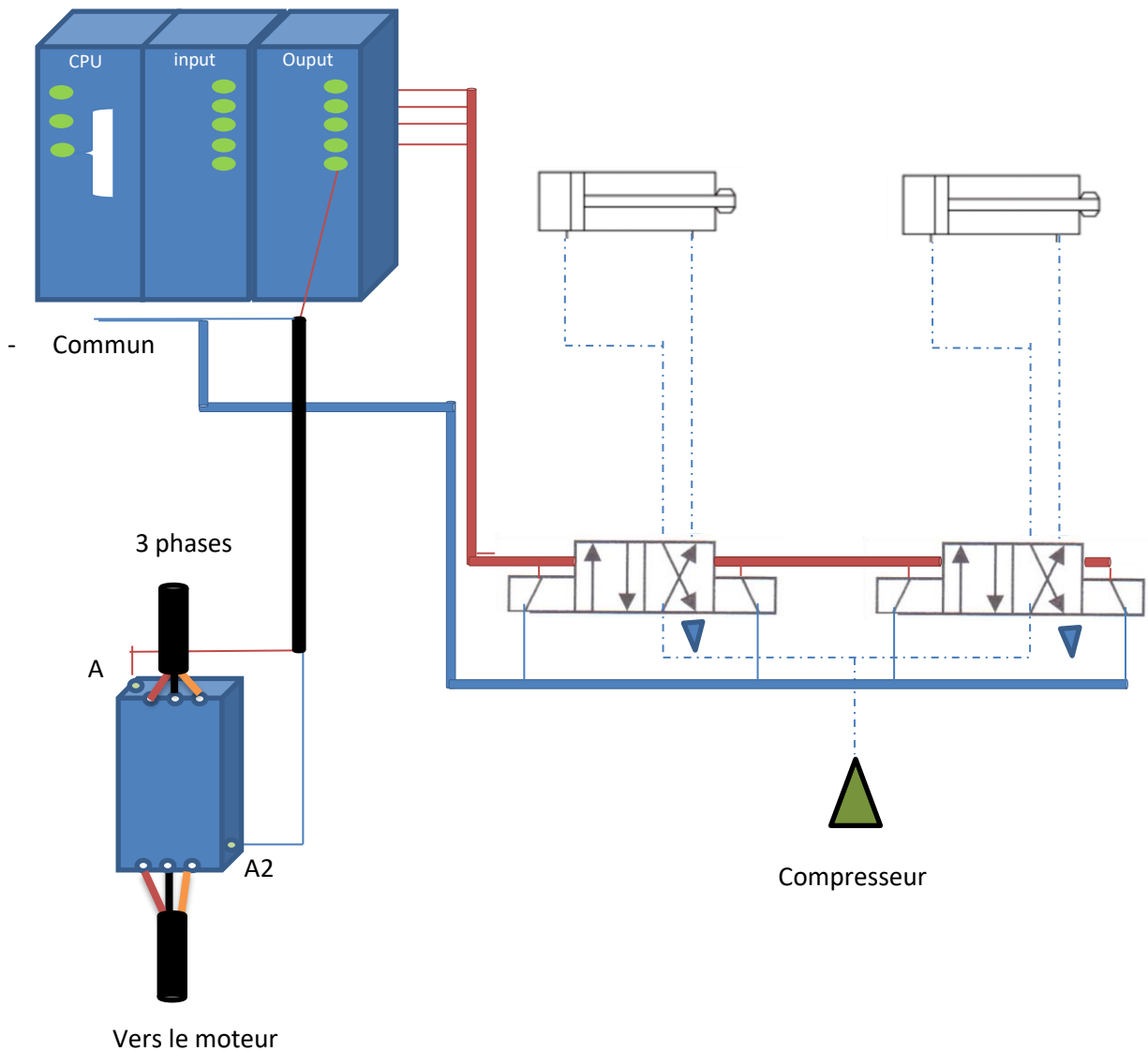
Composant n°06

Figure 99 Composants d'automates Programmables (partie 02)**Exercice 03 :**

Identifier tous les paramètres que contient cet automate

**Figure 100** Automate programmable S7-300 de SIEMENS

- **Solution :**

Exercice 01

Pour les capteurs, on procède de la même manière, chaque capteur doit être alimenté puis branché sa sortie vers l'automate. On a six capteurs donc on va utiliser 6 adresses du module d'entrée de l'automate.

Exercice 02 :

Identification des modules :

1. Module d'entrée digital (TOR), 8 entrées de la gamme 1200 de Siemens, pour la CPU 1221.

2. Automate 1200 de Siemens avec un module d'entrée de 14 entrées et un module de sortie 8 sorties. La CPU de référence 1214c avec DC/DC/DC qui signifie que l'alimentation de la CPU se fera en DC les entrées en DC et les Sortie également en DC
3. Module de communication Ethernet, il permet à l'automate de se connecter d'ans un réseau Ethernet.
4. Module de communication RS232, il permet à l'automate de se connecter à l'aide d'un port RS232.
5. Adaptateur PC USB, il permet de faire connecter l'automate au PC
6. Le Rack : il sert de support aux modules d'automate et également synchronise la communication
7. Cpu 300 : c'est une cpu d'un automate de la gamme 300

Exercice 03

Identifier les paramètres d'un automate :

- Il s'agit d'un automate compact SIEMENS
- Gamme S7-300
- CPU 314C
- Deux ports Ethernet
- Un port profibus
- 5 entrées analogiques de résolution 12 bit
- 2 sorties analogiques de résolution 12 bit
- 24 entrées digitales 24 v
- 16 sorties digitales 24 v

Chapitre IV Programmation des Automates Programmables Industriels

IV. Introduction

Dans ce chapitre on présentera les outils et les différentes procédures pour convertir le model GRAFCET obtenu en langage de programmation de l'Automate Programmable.

On présentera également le câblage et l'adressage des entrées / sorties d'un automate et on terminera se chapitre par un exemple qui recapitule toutes les procédures d'automatisations à savoir de cahier des charges jusqu'à la programmation des modes marches et arrêts.

IV.1 Cycle d'un automate programmable

Le temps de cycle d'un automate à scrutation cyclique est le temps qu'il faut pour à l'automate pour exécuter le programme d'une manière complète.

Ce temps est directement lié à la taille et à la nature du programme à traiter. Le cycle comporte trois phases qui sont :

- Acquisition des entrées : lecture de la carte d'entrée
- Traitement du programme : exécution du programme avec une image des entrées obtenues.
- Mise à jour de l'ensemble des sorties.

Ces trois phases se répètent jusqu'à ce qu'on donne l'ordre d'arrêt de l'automate. L'utilisation d'une image des entrées a pour but d'éviter les aléas en cours de traitement. Celui-ci s'effectue sur un état des entrées mémorisé, donc figé pendant toute la durée du cycle, et non en fonction de leur état physique qui est susceptible d'évoluer au cours de ce même cycle. La mise à jour des sorties intervient en fin de cycle, quand toutes les commandes à transmettre vers l'extérieur sont définies.

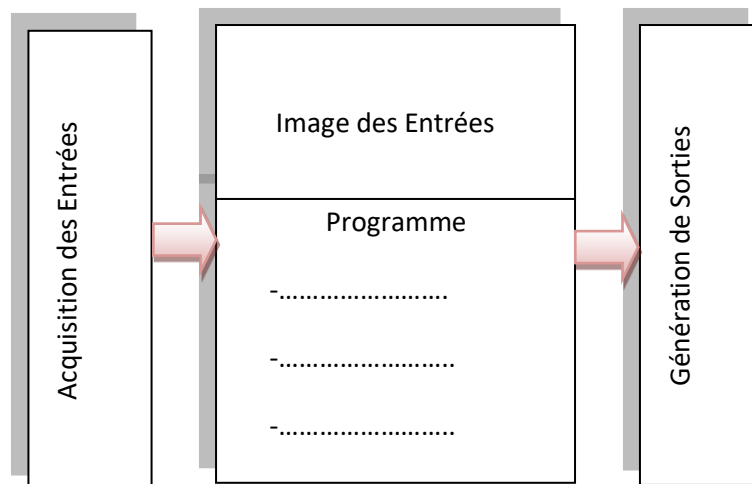


Figure 101 Etapes d'un cycle de fonctionnement

IV.2 Différents langages de programmation

La programmation des automates programmables industriels s'effectue à l'aide de cinq langages de programmation qui sont :

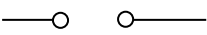
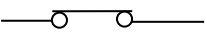
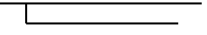
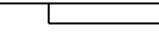
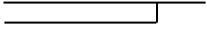
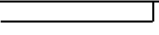
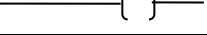
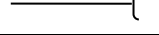
IV.2.1 Langage LADDER

C'est un langage de programmation graphique qui des symboles sous une forme analogue à celle des schémas électriques. Le Ladder appelé également langage contact très simple à utiliser et se compose :

- Des contacts qui représentent des variables booléennes.
- Des bobines qui représentent les sorties
- Des blocs fonctionnels qui permettent de réaliser des opérations plus complexes.

Le Ladder utilise des symboles très proche des schémas de câblage classique, le tableau suivant donne les symboles le plus utilisés :

Tableau 3 : symboles de langage Ladder

Fonction	Symbole	
	Européen	Américain
Contact ouvert au repos		
Contact fermé au repos		
Début d'un branchement		
Fin d'un branchement		
Affectation		

Exemple :

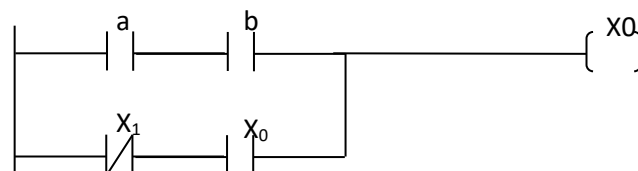


Figure 102 Exemple d'un réseau Ladder

IV.2.2 Langage List (IL) :

Ce langage textuel de bas niveau est un langage à une instruction par ligne, il peut être comparé au langage assembleur.

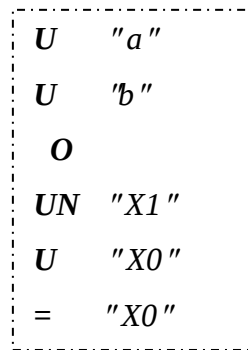
Exemple :

Figure 103 Exemple d'un réseau List

IV.2.3 Langage FBD :

FBD (« Function Block Diagram », ou schéma par blocs): ce langage permet de programmer graphiquement à l'aide de blocs, représentant des variables, des opérateurs ou des fonctions, il permet de manipuler tous les types de variables.

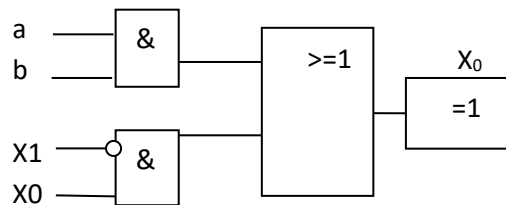


Figure 104 Exemple d'un réseau FBD

IV.2.4 Langage SFC :

Issu du langage GRAFCET, ce langage, de haut niveau, permet la programmation aisée de tous les procédés séquentiels.

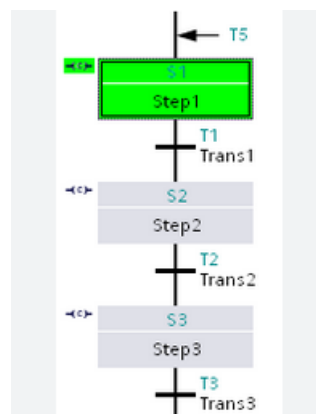


Figure 105 Exemple d'un grafcet en step 7

IV.2.5 Langage ST :

Ce langage est un langage textuel de haut niveau. Il permet la programmation de tout type d'algorithme plus ou moins complexe.

```

1  // Program control IF
2
3  IF "MyInt_1" > 20 THEN
4      "Light_2" := True;
5  ELSE
6      "Light_2" := False;
7  END_IF;
8

```

Figure 106 Exemple d'un réseau du langage structuré

IV.3 Traduction d'un grafcet en ladder

IV.3.1 Mise en équation d'un grafcet :

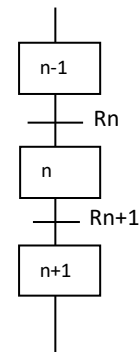
Règles générales pour qu'une étape s'active :

- L'étape immédiatement précédente soit active ;
- La réceptivité immédiatement précédente soit vraie ;
- L'étape immédiatement suivante soit non active ;
- Après activation l'étape mémorise son état.

Soit le grafcet ci-contre :

X_{n-1} : est la variable de l'étape **$n-1$**

X_n : est la variable de l'étape **n**



En utilisant les règles d'évolution d'un grafcet, on construira la table de vérité suivante :

Table de vérité

CAX _n	CDX _n	X _n	X _{n+}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

$CAX_n = X_{n-1} \cdot R_n$: Condition d'activation

$CDX_n = X_{n+1}$: Conditions de désactivation

Tableau de Karnaugh

	00	01	11	10
0	0	0	1	1
1	1	0	1	1

Après la simplification avec Karnaugh on obtient l'équation d'évolution d'étapes donnée ci-dessous :

$$X_n = CAX_n + \overline{CDX_n} \cdot X_n \quad \dots\dots\dots(1)$$

IV.4 Programmation de grafcet à séquence unique

La programmation d'un grafcet à séquence unique signifie que le système ne possède pas des situations de divergence et le grafcet est composé d'étapes consécutives.

IV.4.1 Situation de deux étapes consécutives

Condition d'activation de X_2

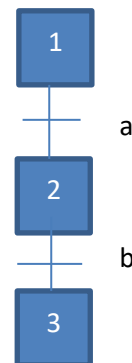
- $CAX_2 = X_1.a$

Condition de désactivation

- $CDX_2 = \overline{X_3}$

Donc l'équation d'évolution de X_2 est :

$$X_2 = X_1.a + \overline{X_3} \cdot X_2$$



Exemple 01 :

Soit le grafcet suivant :

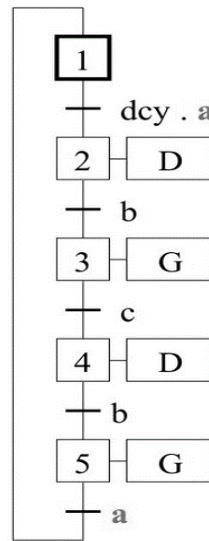


Figure 107 Exemple d'un grafcet

- Donner les équations des étapes
- Donner le lagage Ladder correspondant

Solution :

Pour trouver les équations des étapes on utilisera l'équation d'évolution des étapes (1)

- **Les équations**

$$X_1 = X_5 \cdot a + \overline{X_2} \cdot X_1$$

$$X_2 = X_1 \cdot Dcy \cdot a + \overline{X_3} \cdot X_1$$

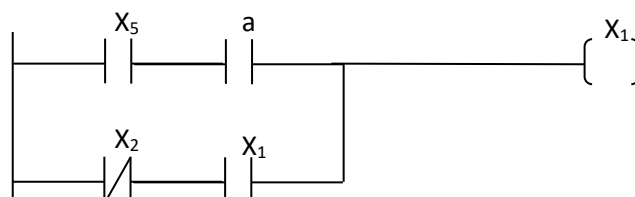
$$X_3 = X_2 \cdot b + \overline{X_4} \cdot X_3$$

$$X_4 = X_3 \cdot c + \overline{X_5} \cdot X_4$$

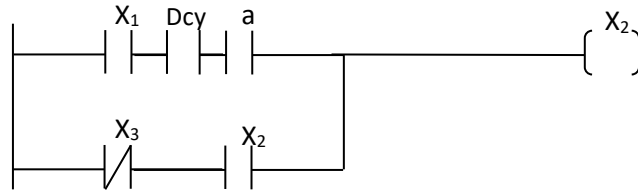
$$X_5 = X_4 \cdot b + \overline{X_0} \cdot X_5$$

- **Ladder** : chaque équation sera convertie en réseau comme suivant :

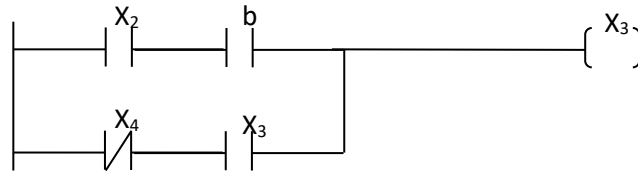
-Etape 1 :



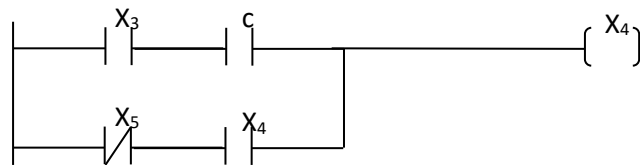
-Etape 2 :



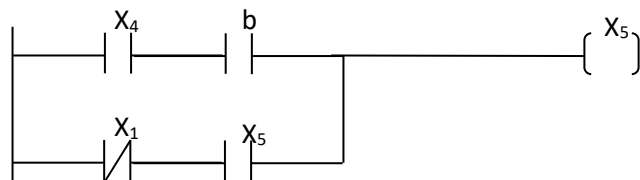
-Etape 3 :



-Etape 4 :



-Etape 5 :



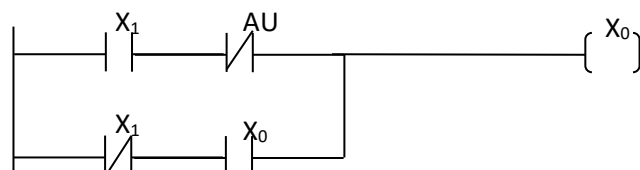
• **Exemple 02 :**

Donner le ladder des deux grafçets de commande de chariot de la figure 48 et la figure 49, le grafçet de sécurité et le grafçet de production.

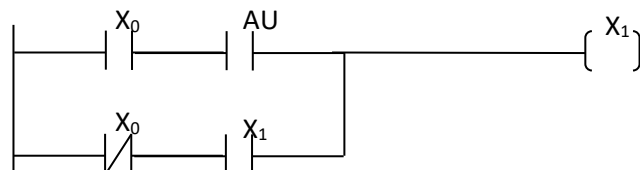
Solution :

• **Ladder des étapes**

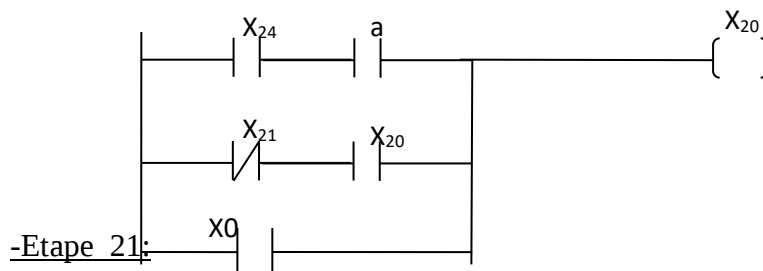
-Etape 0:



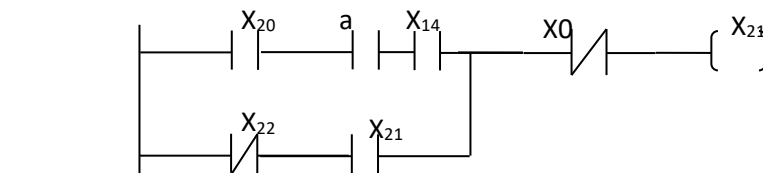
-Etape 1:



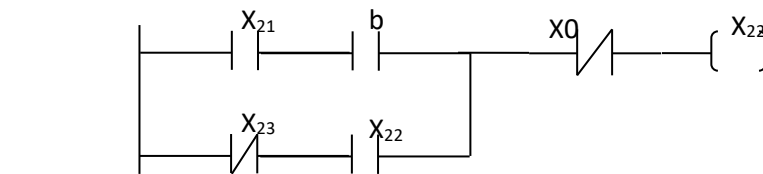
-Etape 20:



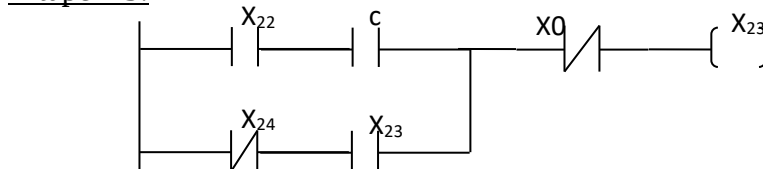
-Etape 21:



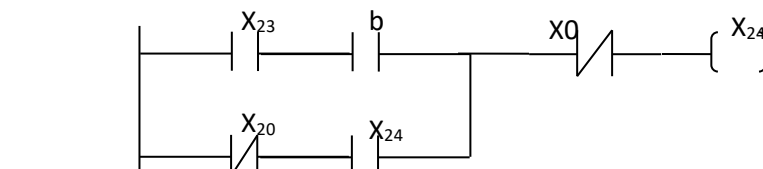
-Etape 22:



-Etape 23:



-Etape 24:



IV.5 Programmation de graficets à séquences multiples.

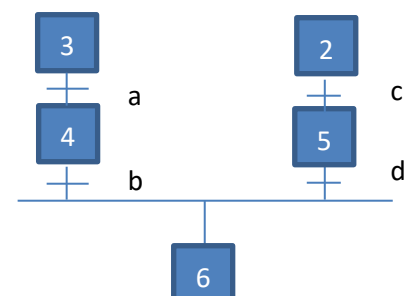
Dans ce type de graficet on retrouve des situations de divergence, exprimé soit par la sélection des mode manuel et automatique ou deux sous système qui fonctionne en parallèles. Dans ce qui suit on va voir quelques situations de séquences multiples et comment les programmer.

IV.5.1 Situation de convergence en « OU » :

Condition d'activation de X_4

- $CAX_4 = X_3.a$

Condition de désactivation



- $CDX_4 = X_6$

$$X_4 = X_3.a + X_6 . \overline{X_4}$$

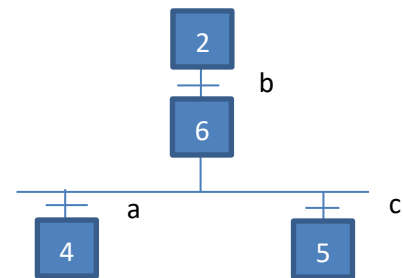
IV.5.2 Situation de divergence en « OU »

Condition d'activation de X_6

- $CAX_6 = X_2.b$

Condition de désactivation

- $CDX_6 = \overline{X_4 + X_5}$



$$X_6 = X_2 . b + \overline{X_4 + X_5} . X_6$$

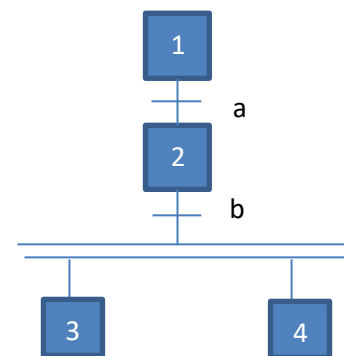
IV.5.3 Situation de convergence en « ET » :

Condition d'activation de X_2

- $CAX_2 = X_1.a$

Condition de désactivation

- $CDX_2 = \overline{X_3 . X_4}$



$$X_2 = X_1.a + \overline{X_3 . X_4} . X_2$$

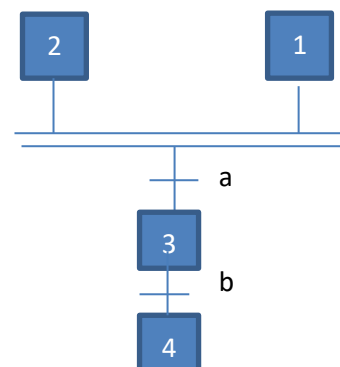
IV.5.4 Situation de divergence en « ET »

Condition d'activation de X_3

- $CAX_3 = X_2.X_1.a$

Condition de désactivation

- $CDX_3 = \overline{X_4}$

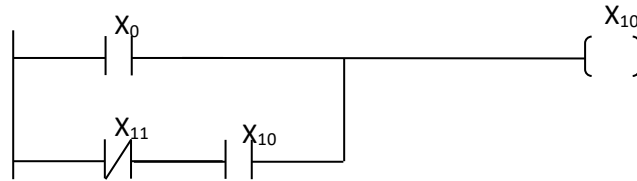


$$X_3 = X_1 . X_2 . a + \overline{X_4} . X_3$$

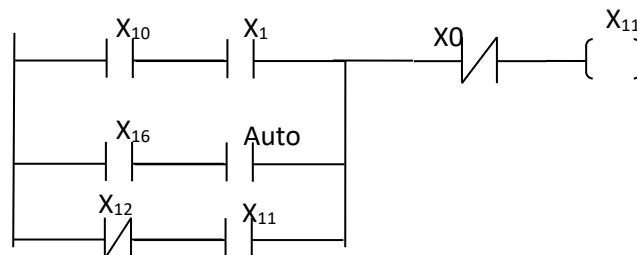
Exemple 01 :

Considérant le grafctet de conduite de la figure 50 ou on a une divergence en **OU**

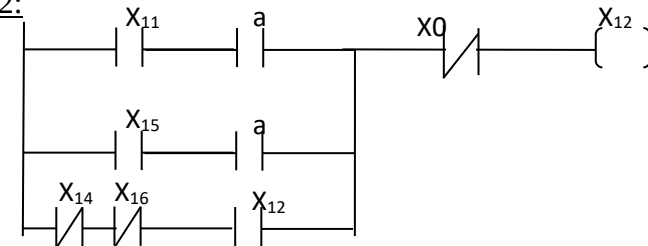
-Etape 10:



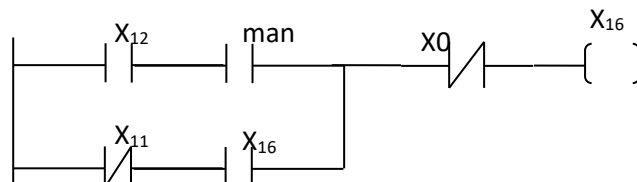
-Etape 11:



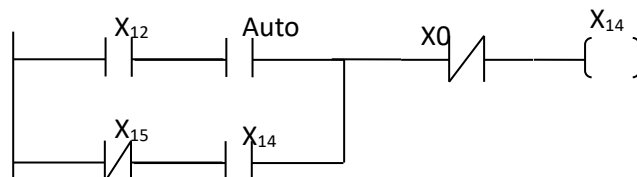
-Etape 12:



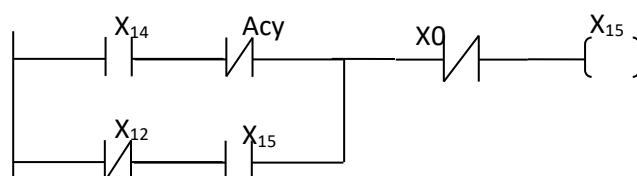
-Etape 16:



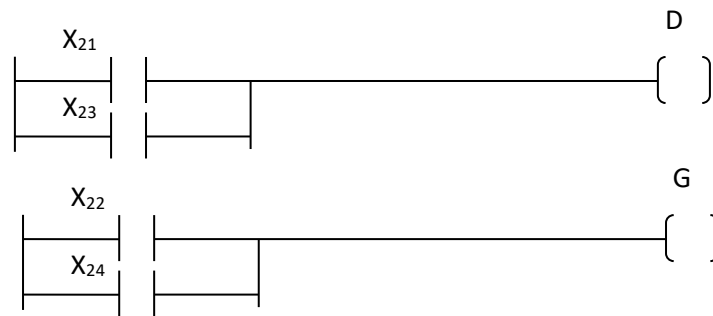
-Etape 14:



-Etape 15:



- **Ladder des actions**



Exemple 02

- Donner le ladder de ce grafcet

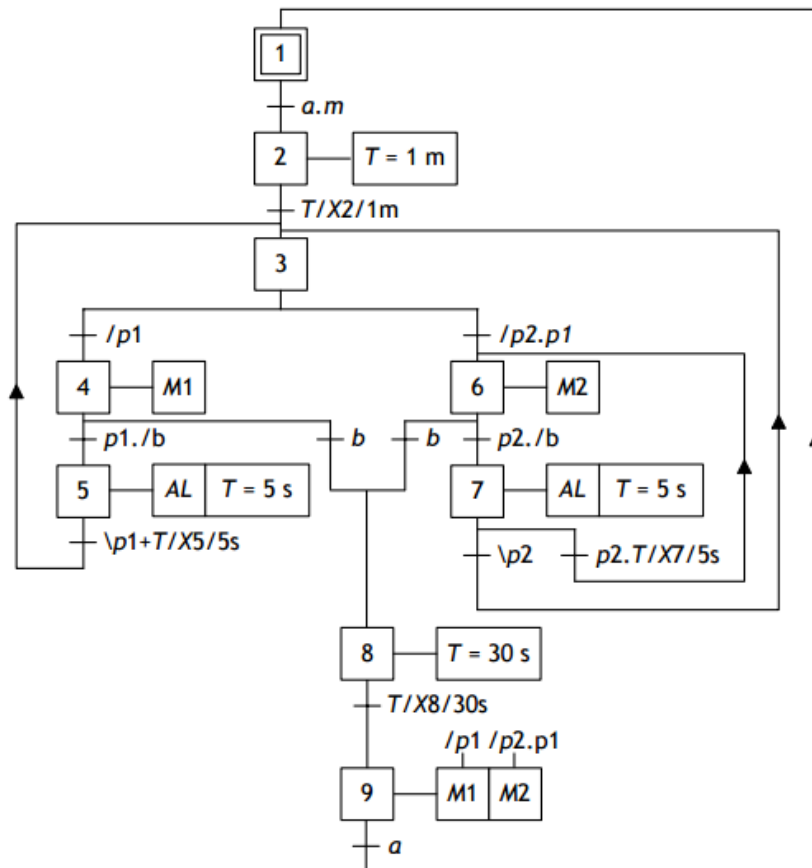


Figure 108 Exemple de grafcet avec divergence en "ou"

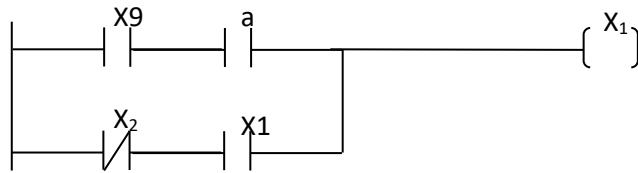
Solution :

Etape 01

- **Equation :**

$$X_1 = X_9 \cdot a + \overline{X_2} \cdot X_1$$

- Réseau 01 :

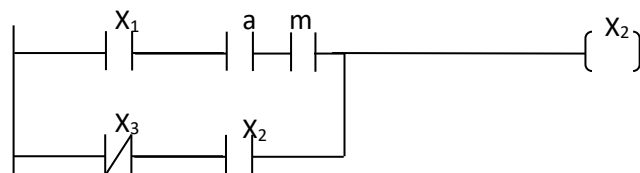


Etape 02

- Equation :

$$X_2 = X_1 \cdot a \cdot m + \overline{X_3} \cdot X_2$$

- Réseau 02 :

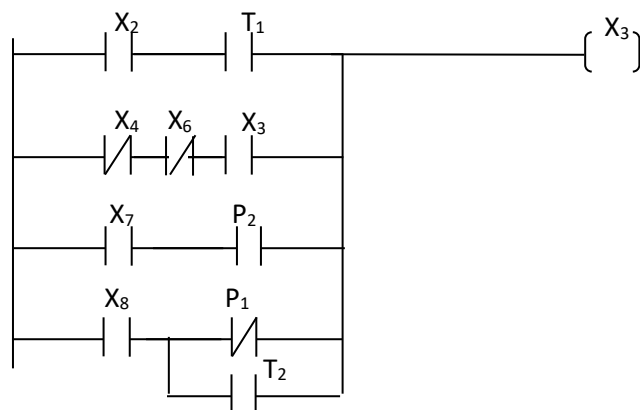


Etape 03

- Equation :

$$X_3 = X_2 \cdot T_1 + X_8 \cdot (\overline{p_1} + T_2) + X_7 \cdot p_2 + \overline{X_4} + \overline{X_6} \cdot X_3$$

- Réseau 03 :

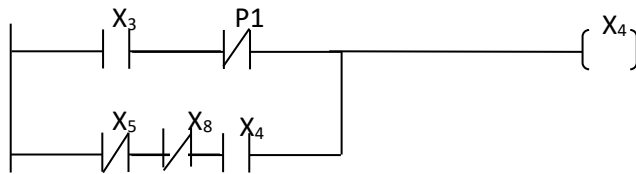


Etape 04

- Equation :

$$X_4 = X_3 \cdot \overline{P_1} + \overline{X_5} + \overline{X_8} \cdot X_4$$

- Réseau 04 :

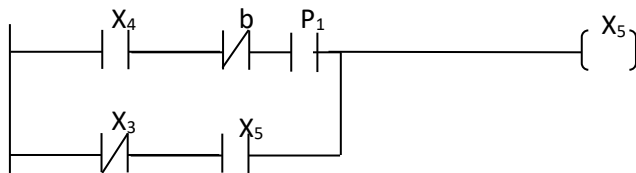


Etape 05

- Equation :

$$X_5 = X_4 \cdot \overline{b} \cdot P_1 + \overline{X_3} \cdot X_5$$

- Réseau 05 :

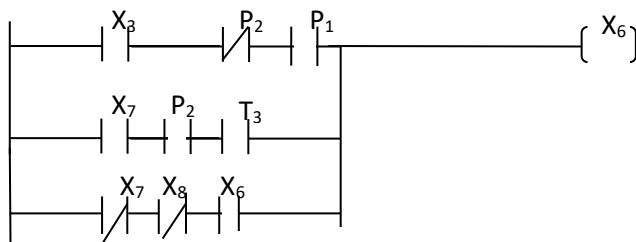


Etape 06

- Equation :

$$X_6 = X_3 \cdot \overline{P_2} \cdot P_1 + X_7 \cdot P_2 \cdot T_3 + \overline{X_7} + \overline{X_8} \cdot X_6$$

- Réseau 06 :

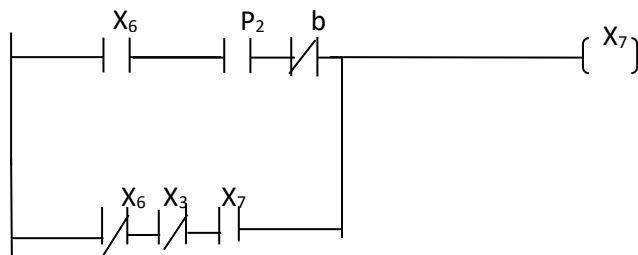


Etape 07

- Equation :

$$X_7 = X_6 \cdot \overline{b} \cdot P_2 + \overline{X_6} + \overline{X_3} \cdot X_7$$

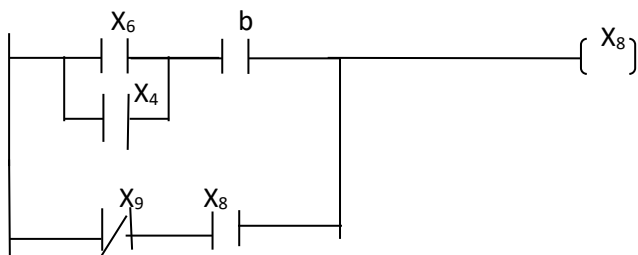
- Réseau 07 :

**Etape 08**

- Equation :

$$X_8 = X_6 \cdot b + X_4 \cdot b + \overline{X_9} \cdot X_8$$

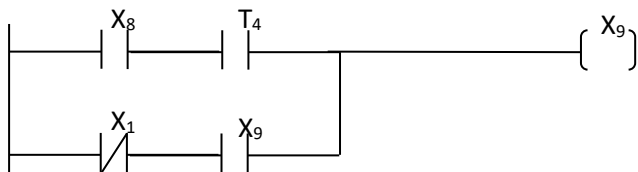
- Réseau 08 :

**Etape 09**

- Equation :

$$X_9 = X_8 \cdot T_4 + \overline{X_1} \cdot X_9$$

- Réseau 09 :

**IV.6 Conclusion :**

Dans ce chapitre nous avons présenté tous les outils nécessaires à l'automatisation des systèmes, de la modélisation avec le GRAFCET au programme API.

Dans le chapitre suivant, on présentera le système SCADA qui également très important dans l'automatisation, à savoir, il permet de faire une supervision de l'automatisme avec laquelle on pourra commander et surveiller le système à partir d'un site éloigné.

- **Exercices du chapitre :**

Donner le programme Ladder des grafjets suivants :

1. Grafjet de conduite

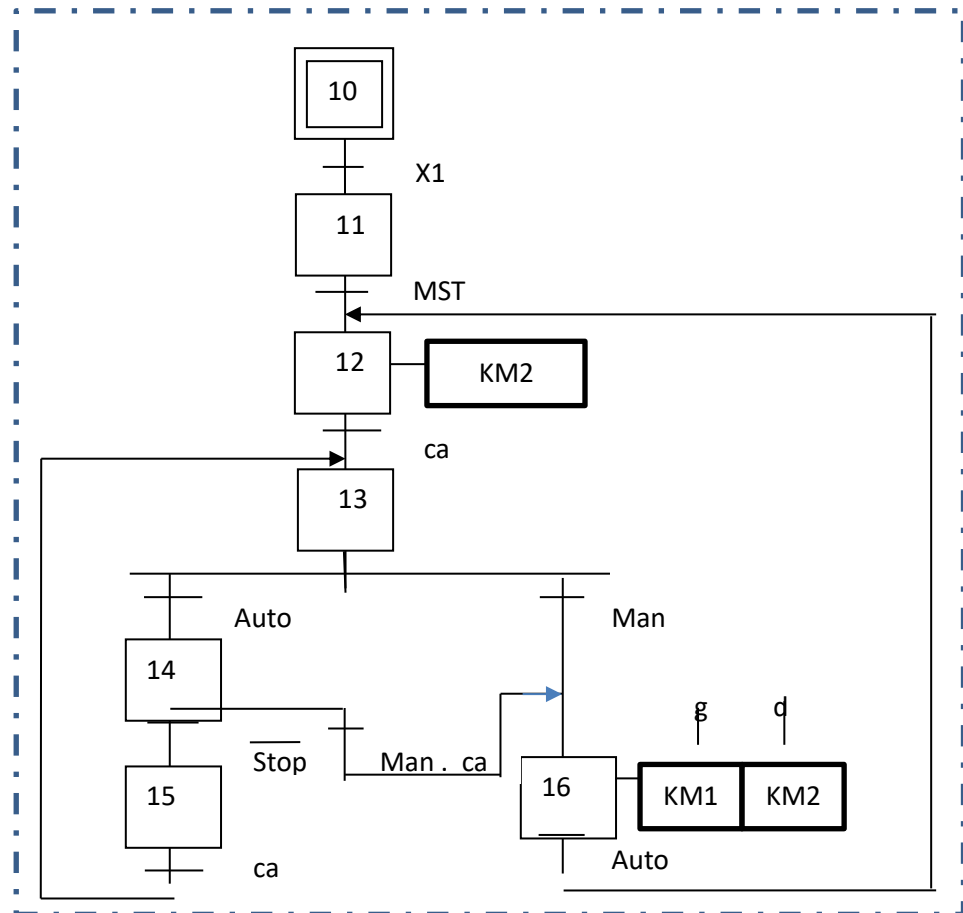


Figure 109 Grafjet de Conduite

2. Grafcet de sécurité

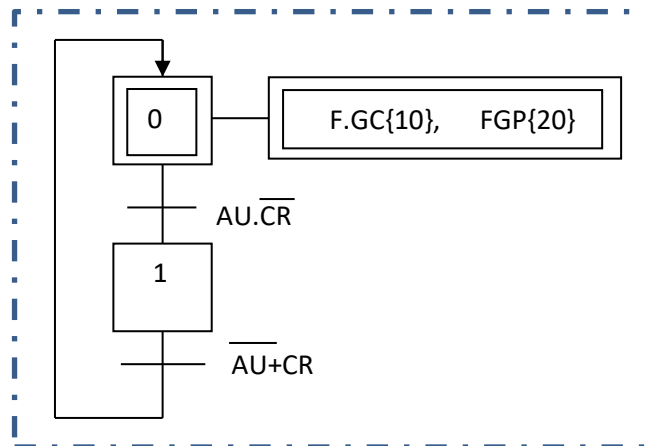


Figure 110 Grafcet de sécurité

3. Grafcet de production

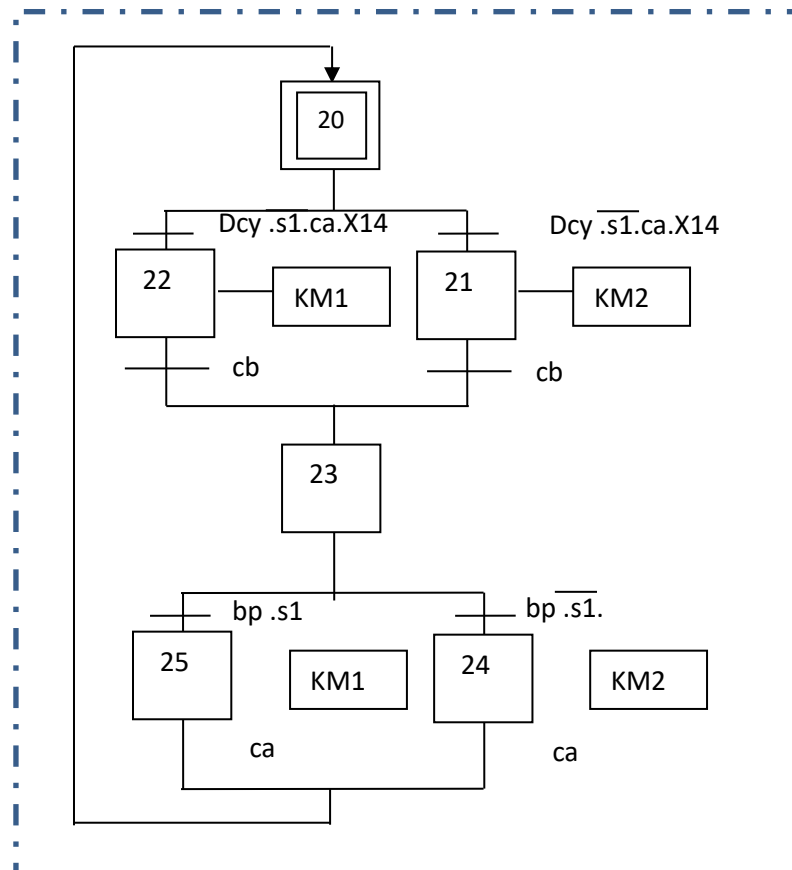


Figure 111 Grafcet de production

• Solution

Ces trois grafkets se complètent, le premier grafket est le grafket de conduite c'est lui qui gère le mode manuel et automatique et l'arrêt de fin de cycle et également la remise en route

du système, donc on va utiliser quelques étapes de ce grafcet pour activer quelques étapes de grafcet de production.

Le grafcet 02 a le niveau hiérarchique le plus élevé, il permet d'activer les autres grafkets et également de les désactiver, donc on trouvera des étapes de ce grafcet dans les équations des étapes des autres grafkets.

Ce grafcet de sécurité va forcer les autres grafkets à leurs étapes initiales à « 0 » si un arrêt d'urgence est enclenché.

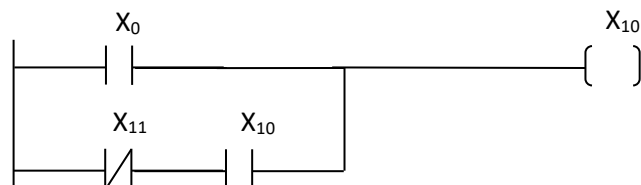
1. Grafcet de conduite :

Etape 10 :

- **L'équation**

$$X_{10} = X_0 + \overline{X_{11}} \cdot X_{10}$$

- **Ladder**



- **LIST**

- O 'X0'
- O
- UN 'X11'
- U 'X10'
- = 'X10'

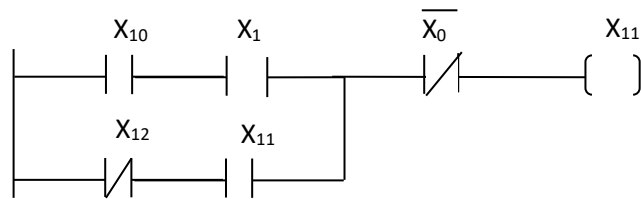
L'étape X_0 est due au forçage, car lorsqu'un l'étape « 0 » est active, les grafkets sont forcés à leurs étapes initiales.

Etape 11 :

- **L'équation**

$$X_{11} = (X_{10} \cdot X_1 + \overline{X_{12}} \cdot X_{11}) \cdot \overline{X_0}$$

- **Ladder**



- **LIST**

```

• U(
• U  'X10'
• U  'X1'
• O
• UN 'X12'
• U  'X11'
• )
• UN 'X0'
• =  'X11'

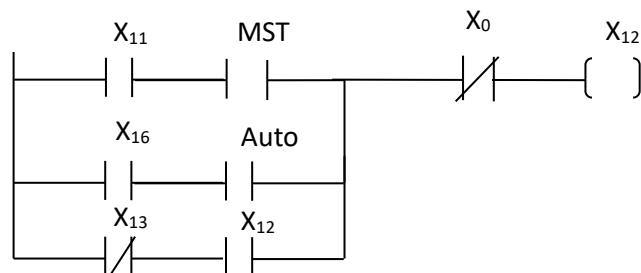
```

Etape 12 :

- **L'équation**

$$X_{12} = (X_{11} \cdot MST + X_{16} \cdot Auto + \overline{X_{13}} \cdot X_{12}) \cdot \overline{X_0}$$

- **Ladder**



- **LIST**

```

• U (
• U 'X11'
• U 'MST'
• O
• U 'X16'
• U 'Auto'
• O
• U 'X13'
• UN 'X12'
• )
• UN 'X0'
• = 'X12'

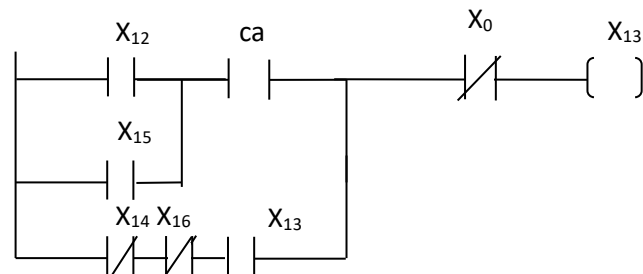
```

Etape 13 :

- **L'équation**

$$X_{13} = ((X_{12} + X_{15}).ca + \overline{X_{14} + X_{16}}.X_{13}).\overline{X_0}$$

- **Ladder**



- **LIST**

```

• U (
• O 'X12'
• O 'X15'
• U 'ca'
• O
• UN 'X14'
• UN 'X16'
• U 'X13'
• )
• UN 'X0'
• = 'X13'

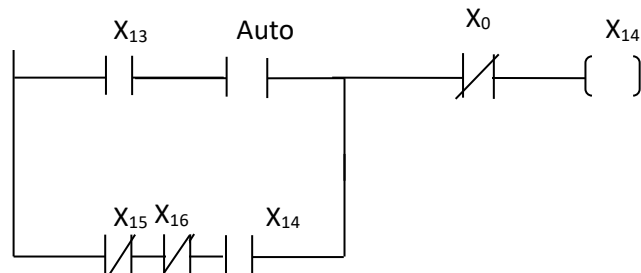
```

Etape 14 :

- **L'équation**

$$X_{14} = (X_{13} \cdot Auto + \overline{X_{15}} + X_{16} \cdot X_{14}) \cdot \overline{X_0}$$

- **Ladder**



- **LIST**

```

• U (
• U 'X13'
• U 'Auto'
• O
• UN 'X15'
• UN 'X16'
• U 'X14'
• )
• UN 'X0'
• = 'X14'

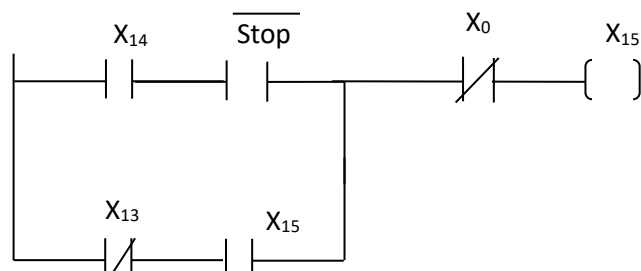
```

Etape 15 :

- **L'équation**

$$X_{15} = (X_{14} \cdot \overline{Stop} + \overline{X_{13}} \cdot X_{15}) \cdot \overline{X_0}$$

- **Ladder**



- LIST

```

• U (
• U 'X14'
• UN 'Stop'
• O
• UN 'X13'
• U 'X15'
• )
• UN 'X0'
• = 'X15'

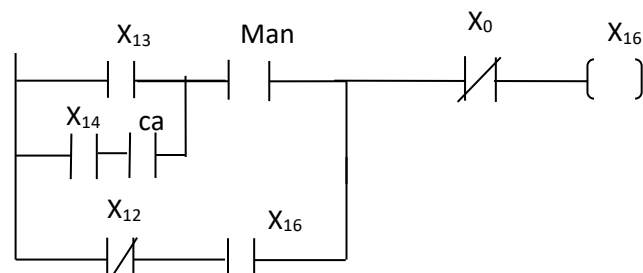
```

Etape 16 :

- L'équation

$$X_{16} = ((X_{13} + X_{14}.ca).Man + \overline{X_{12}}.X_{16}).\overline{X_0}$$

- Ladder



- LIST

```

• U (
• U(
• O 'X13'
• O
• U 'X14'
• U 'ca'
• )
• U 'Man'
• O
• UN 'X12'
• U 'X16'
• )
• UN 'X0'
• = 'X16'

```

Remarque :

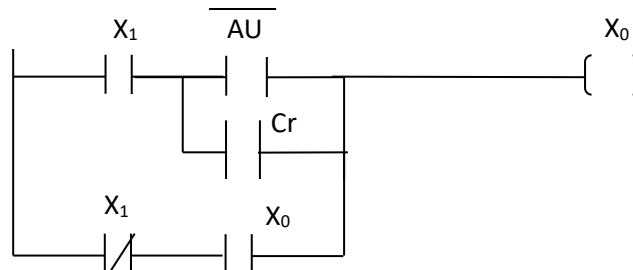
La relation $\overline{X.Y}$ ne peut pas être tracer en Ladder, il la convertir en $\overline{X} + \overline{Y}$

2. Grafcet de sécurité :**Etape 0 :**

- L'équation**

$$X_0 = X_1 \cdot (\overline{AU} + Cr) + \overline{X_1} \cdot X_0$$

- Ladder**



- LIST**

```

• U 'X1'
• U(
• ON 'AU'
• O 'Cr'
• )
• O
• UN 'X1'
• U 'X0'
• = 'X0'

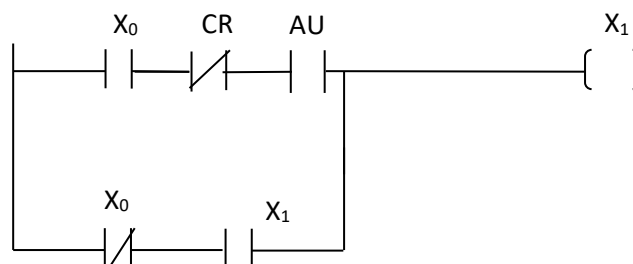
```

Etape 1 :

- L'équation**

$$X_1 = X_0 \cdot \overline{Cr} \cdot AU + \overline{X_0} \cdot X_1$$

- Ladder**



- LIST

```

• U  'X0'
• UN 'Cr'
• U  'AU'
• O
• UN 'X0'
• U  'X1'
• =  'X1'

```

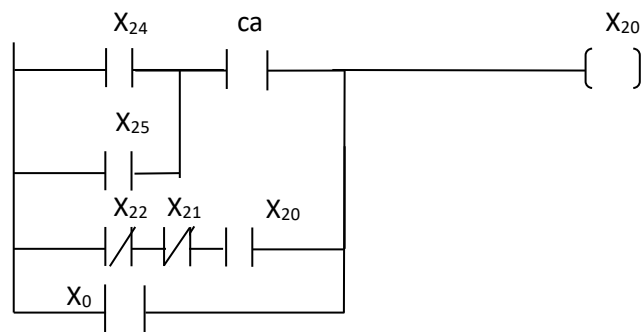
3. Grafcet de production

Etape 20 :

- L'équation

$$X_{20} = X_0 + (X_{25} + X_{24}).ca + \overline{X_{22} + X_{21}}.X_{20}$$

- Ladder



- LIST

```

• U  'ca'
• U (
• O  'X24'
• O  'X25'
• )
• O
• UN 'X21'
• UN 'X22'
• U  'X20'
• O  'X0'
• =  'X20'

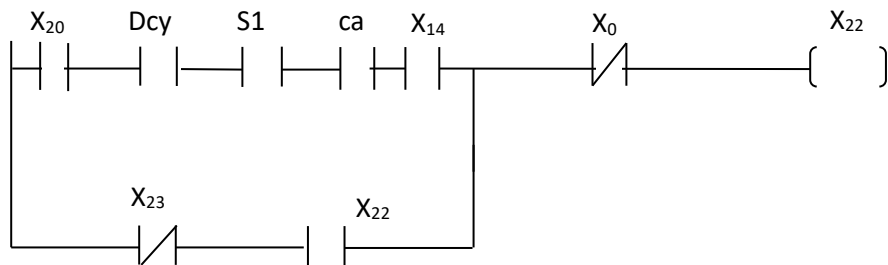
```

Etape 22 :

- **L'équation**

$$X_{22} = (X_{20} \cdot Dcy \cdot S1 \cdot ca + \overline{X_{23}} \cdot X_{22}) \cdot \overline{X_0}$$

- **Ladder**



- **LIST**

```

• U(
• U  'X20'
• U  'Dcy'
• U  'S1'
• U  'ca'
• U  'X14'
• O
• UN 'X23'
• U  'X22'
• )
• UN 'X0'
• =  'X22'

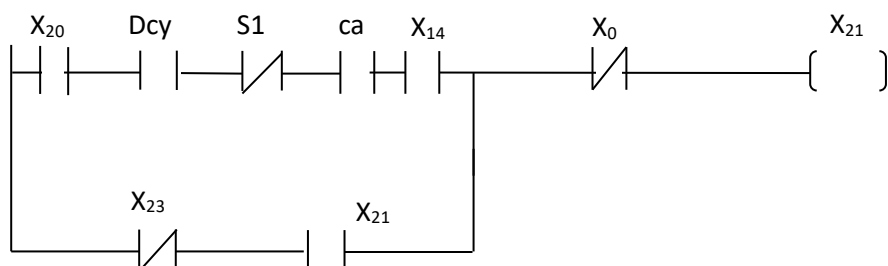
```

Etape 21 :

- **L'équation**

$$X_{21} = (X_{20} \cdot Dcy \cdot \overline{S1} \cdot ca + \overline{X_{23}} \cdot X_{21}) \cdot \overline{X_0}$$

- **Ladder**



- **LIST**

```

• U (
• U 'X20'
• U 'Dcy'
• UN 'S1'
• U 'ca'
• U 'X14'
• O
• UN 'X23'
• U 'X21'
• )
• UN 'X0'
• = 'X21'

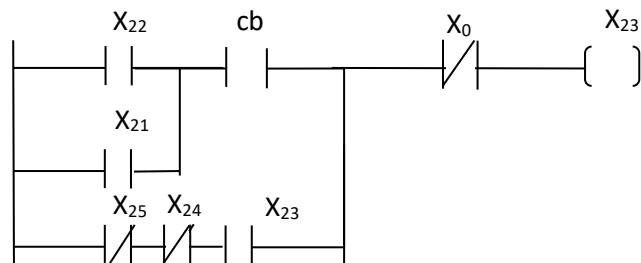
```

Etape 23 :

- **L'équation**

$$X_{23} = ((X_{22} + X_{21}).cb + \overline{X_{25} + X_{24}}.X_{23}).\overline{X_0}$$

- **Ladder**



- **LIST**

```

• U(
• U 'cb'
• U (
• O 'X22'
• O 'X21'
• )
• O
• UN 'X24'
• UN 'X25'
• U 'X23'
• )
• UN 'X0'
• = 'X23'

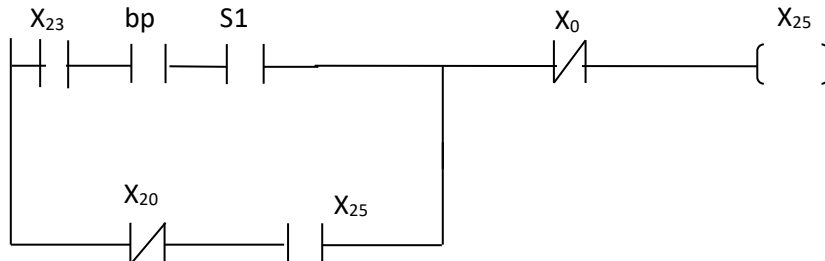
```

Etape 25 :

- **L'équation**

$$X_{25} = (X_{23} \cdot bp \cdot S_1 + \overline{X_{20}} \cdot X_{25}) \cdot \overline{X_0}$$

- **Ladder**



- **LIST**

```

• U(
• U  'X23'
• U  'bp'
• U  'S1'
• O
• UN 'X20'
• U  'X25'
• )
• UN 'X0'
• =  'X25'

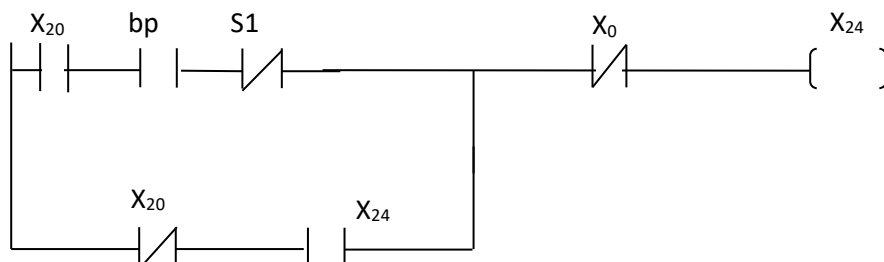
```

Etape 24 :

- **L'équation**

$$X_{24} = (X_{20} \cdot bp \cdot \overline{S_1} + \overline{X_{20}} \cdot X_{24}) \cdot \overline{X_0}$$

- **Ladder**



- **LIST**

```

• U (
• U 'X23'
• U 'bp'
• UN 'S1'
• O
• UN 'X20'
• U 'X24'
• )
• UN 'X0'
• = 'X24'

```

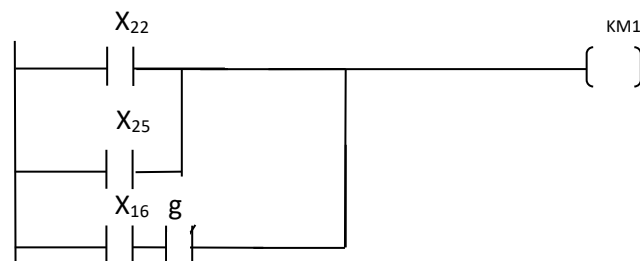
4. Programme des Actions

- **L'Action KM1**

- **L'équation**

$$KM1 = (X_{22} + X_{25} + X_{16} \cdot g)$$

- **Ladder**



- **LIST**

```

• O 'X22'
• O 'X25'
• O(
• U 'X16'
• U 'g'
• )
• = 'KM1'

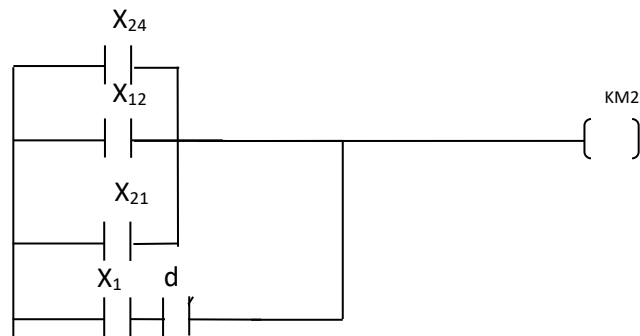
```

- **L'Action KM2**

- **L'équation**

$$KM2 = (X_{12} + X_{24} + X_{21} + X_{16} \cdot d)$$

- **Ladder**



- **LIST**

- O 'X₂₄'
- O 'X₁₂'
- O 'X₂₁'
- O(
- U 'X₁₆'
- U 'd'
-)
- = 'KM2'

Références :

1. Hamdi Hocine, « Automatismes logiques : modélisation et commande », volumes 1 et 2, éditions de L'UMC, 2006.
2. William Bolton, « Les automates programmables industriels », Dunod, 2010.
3. J.C. Humblot, « Automates programmables industriels », Hermes Science Publications, 1993.
4. Simon Moreno, Edmond Peulot, « Le GRAFCET : conception, implantation dans les automates programmables industriels », Delagrave, 2009.
5. Kevin Collins, « La programmation des automates programmable [sic] industriels », Meadow Books, 2007.
6. G. Michel, « Les A.P. I : architecture et applications des automates programmables industriels », Dunod, 198