

Département d'Automatique, Télécommunication et d'Electronique

Projet de Fin d'Etudes

Pour l'obtention du diplôme de Master

Filière : Automatique et Informatique Industrielle

Spécialité : Automatique

Thème

**Migration d'un automate TSX micro (Schneider) vers Siemens S7 1200
(TIA Portal) au niveau de CEVITAL**

Préparé par :

- BENBOUABDELLAH Nacereddine
- HAFHOUF Lamine

Dirigé par :

Mme Noura BELLAHSENE
Mr Mahrez BERKOUK

Examiné par :

Mr Lyes TIGHZERT
Mme Saliha GADA

Année universitaire : 2023/2024

Remerciement

Tout d'abord nous remercions Allah le tout-puissant de nous avoir donné la connaissance, le courage, la patience et la santé pour réaliser notre projet de fin d'étude.

Nous adressons nos remerciements les plus sincères à notre promotrice, Mme. Noura Rezika BELLAHSENE, pour avoir suivi nos travaux de si près, pour son aide et tous les conseils qu'il nous a prodigués durant toute la durée de ce travail.

Nous tenons à remercier notre co-promoteur monsieur BERKOUK Mahrez du pôle sucre du complexe Cevital pour toutes les informations qu'il nous a donné et sa prise en charge tout au long de notre stage de nous avoir fait l'honneur d'assurer la réalisation et l'encadrement de notre travail.

Nos remerciements et notre sincère gratitude Aux membres du jury pour avoir pris le temps de lire et d'évaluer ce travail.

Enfin, nous remercions toute personne qui nous a aidées de près ou de loin.

Dédicaces

Quoi que je fasse quoi je dise mes mots ne vaut rien devant ce que j'ai vécu durant ces années et ce que j'ai reçu de la part de ma chère famille et mes amis

Je dédie ce modeste travail :

Ma chère famille qui mon soutenu et m'aider durant mon parcours scolaire et universitaire.

À mes amis qui on ma deuxième famille.

A toutes et tous personne qui on participer dans ma réussite de pris ou de loin.

À deux personne sans eux je ne pourrai jamais arriver à ce que je suis aujourd'hui bien sûr après la bénédiction de ALLAH

MAY Ramzi et HAFHOUF Hamza.

Lamine

Dédicace

Avant tout je remercie Dieu Le tout puissant de m'avoir donné le courage, la volonté, la patience, et la santé durant toutes ces années et que grâce à lui ce travail a pu être réalisé.

Je dédie ce modeste travail :

A ma très chère mère

Quoi que je fasse ou quoi que je dise, je ne saurai point te remercier comme il se doit. Ton affection me couvre, ta bienveillance me guide et ta présence à mes coté a toujours été ma source de force pour affronter les différents obstacles. Que ce modeste travail soit le fruit de vos innombrables sacrifices.

A mon très cher père

Ce travail est dédié à mon père, qui m'a toujours poussé et motivé dans mes études. J'espère qu'il apprécie cet humble geste comme preuve de reconnaissance de la part d'un fils qui a toujours prié pour son âme.

A ma chère sœur et famille

Je remercie ma chère sœur et ma famille tel que ma **tante Kahina, Sofiane et Hichem** pour leurs soutiens moraux et leurs conseils précieux tout au long de mes études.

A mon cher binôme

Pour son entente et sa sympathie

A ma chère Kaouther

Nacereddine

Table des matières

Introduction générale.....	1
Chapitre I : Présentation de l'entreprise et des différentes parties de la machine à étudier	
I.1 Présentation de l'entreprise CEVITAL	3
I.1.1 Historique de l'entreprise.....	3
I.1.2 Situation géographique	4
I.1.3 Activités de l'entreprise	5
I.1.4 Missions et objectifs	5
I.2 Différentes parties de la machine à étudier	6
I.2.1 Description générale	6
I.2.2 Caractéristiques.....	7
I.2.3 Energies et produits mis en œuvre	8
I.2.4 La partie électrique.....	9
I.2.5 La partie pneumatique.....	12
I.2.6 Convoyeurs	16
I.2.7 Station de colle.....	17
I.2.8 Ensemble cavité	20
I.2.9 Partie instrumentation	20
I.3 Etude de la machine.....	21
I.3.1 Diagramme SADT niveau A-0 pour la barquetteuse	24
I.4 Conclusion	25
II Chapitre II : APIs et logiciels de programmation	
II.1 Introduction	26
II.2 Historique	26
II.3 Qu'est-ce qu'un API ?	27
II.4 Architecture des automates.....	28

II.4.1	Aspect extérieur.....	28
II.4.2	Structure interne	29
II.5	Fonctionnement d'un A.P.I	31
II.6	Les Fonctions principales d'un A.P.I	32
II.6.1	Fonctions logiques ou booléennes.....	32
II.6.2	Fonctions arithmétiques et sur mots.....	33
II.6.3	Fonctions de communication	34
II.6.4	Autres fonctions	34
II.7	Langages de programmation des API.....	35
II.7.1	LD (Ladder Diagram).....	36
II.7.2	IL (Instruction List).....	36
II.7.3	FBD (Function Block Diagram).....	36
II.7.4	ST (Structured Text).....	36
II.7.5	SFC (Sequential Function Charts).....	36
II.8	Les blocs dans TIA Portal	36
II.9	Choix d'un API.....	37
II.10	Les Avantages des APIs	38
II.11	Les inconvénients des APIs	39
II.12	Les automates Schneider	39
II.12.1	L'automates Schneider TSX Micro.....	39
II.12.2	PL7 Pro.....	41
II.13	Présentation de la gamme SIMATIC de SIEMENS.....	42
II.13.1	Présentation de l'automate programmable S7-1200	42
II.13.2	Caractéristiques techniques de l'APIS7-12001214C	44
II.13.3	Caractéristiques de la CPU 1214C DC/DC/RELAY (6ES7214-1HG400XB0)	45
II.13.4	Logiciel de programmation TIA Portal	45
II.13.5	Définition de la supervision	48

II.14	Conclusion	52
III	Chapitre III : Cahier de charges de la machine et programmation de l'API	
III.1	Introduction	53
III.2	Cahier de charges.....	53
III.2.1	Dépilage_Transfert_Collage	53
III.2.2	Formation de caisse	54
III.2.3	Forçage Piston	55
III.3	GRAFCETs	56
III.3.1	Dépilage_Transfert_Collage	57
III.3.2	Formation de Caisse	57
III.3.3	Forçage piston	58
III.4	Migration vers SIEMENS S7 1200 ?	58
III.5	Choix de la CPU	59
III.6	Programmation	60
III.6.1	La conception du programme sur TIA PORTAL V16.....	61
III.6.2	Configuration matérielle	62
III.6.3	Table des variables API.....	63
III.6.4	Tableau d'affectation.....	67
III.6.5	Les blocs dans TIA Portal	68
III.7	Conclusion	86
IV	Chapitre IV : Interfaces de supervision	
IV.1	Introduction	87
IV.2	Configuration de l'IHM.....	87
IV.3	Création d'un nouveau modèle.....	87
IV.4	Vue d'accueil.....	89
IV.5	Vues des GRAFCETs	93
IV.5.1	La configuration des étapes	94

IV.5.2	Configuration des actions	95
IV.5.3	Configuration des transitions	95
IV.6	Vues des GRAFCETs	96
IV.6.1	GRAFCET « Dépilage_Transfert_Collage »	96
IV.6.2	GRAFCET « Formation_Caisse ».....	98
IV.6.3	GRAFCET « Forçage_Piston ».....	98
IV.7	Vue générale	99
IV.8	AlarmesIHM.....	103
IV.8.1	Classes d'alarmes.....	103
IV.8.2	Types des alarmes	104
IV.8.3	Configuration d'alarme	104
IV.9	Conclusion.....	106
Conclusion générale		107
Références bibliographiques		
Annexes		

Liste des figures

Figure I. 1 : Historique de l'entreprise CEVITAL.....	3
Figure I. 2 : Situation géographique de l'entreprise CEVITAL Bejaia	4
Figure I. 3 : Vue générale de la barquetteuse SECS	7
Figure I. 4: Dimensions de la caisse	8
Figure I. 5: Constitution d'un moteur à rotor à cage.....	9
Figure I. 6 : Moteur électrique à un seul sens avec bielle-manivelle.....	10
Figure I. 7 : Contacteur électrique	11
Figure I. 8 : Relais thermique	11
Figure I. 9 : Sectionneur porte fusible	12
Figure I. 10 : Les venteuses à soufflet	13
Figure I. 11 : Ejecteur pneumatique et une pompe à vide	14
Figure I. 12: Les deux types des vérins.....	15
Figure I. 13 : Le FLR FESTO et ces composants	15
Figure I. 14 : Convoyeur à taquets et son codeur de contrôle.....	16
Figure I. 15 : Convoyeur à courroie (Photo prise en entreprise)	16
Figure I. 16 : Illustration descriptif des différents composants du ITW Challenger QUATTRO.....	17
Figure I. 17 : Ecran tactile de commande.....	19
Figure I. 18 : Photo réelle des équerres de la barquetteuse (prise en entreprise).....	20
Figure I. 19 : Capteur à proximité	20
Figure I. 20: Etapes de fonctionnement du détecteur de présence à action mécanique	21
Figure I. 21 : Différentes vues de bras margeur (réalisé sur le logiciel Solid Works).....	22
Figure I. 22 : Stockage et dépilage des découpes en cartons	22
Figure I. 23 : Représentation des postes	23
Figure I. 24 : Identification de la zone d'étude	24
Figure I. 25 : Diagramme SADT niveau A-0 de la barquetteuse.....	24
Figure I. 26 : diagramme FAST de la barquetteuse	25
Figure II. 1 : L'évolution des automates SIEMENS.....	27
Figure II. 2 : Types des APIs	28
Figure II. 3 : Automate modulaire (Siemens).....	29
Figure II. 4: Architecture d'un API	29
Figure II. 5 : Module d'alimentation d'un API.....	30

Figure II. 6 : Module CPU d'un API	30
Figure II. 7 : Cycle d'un API	32
Figure II. 8 : Exemple d'opérations logiques sur TIA Portal	33
Figure II. 9 : Exemple d'addition et soustraction sur le TIA Portal	33
Figure II. 10 : Compteur incrémental	34
Figure II. 11: L'automate TSX 37-05	40
Figure II. 12 : L'automate TSX 37-08	40
Figure II. 13: L'automate TSX 37-10.....	41
Figure II. 14 : L'automate TSX 37-21	41
Figure II. 15 : Automate programmable S7-1200 1214C.....	44
Figure II. 16 : Vue de portail	46
Figure II. 17 : Vue du projet	47
Figure II. 18 : S7 PLCSIM	48
Figure II. 19 : Etapes pour ajouter un nouveau panel	50
Figure II. 20 : Parcourir et sélectionne d'API.....	51
Figure II. 21 : Connexion entre API et IHM	51
Figure II. 22: Vue IHM.....	51
Figure III. 1: Grafcet de "Dépilage_Transfert_Collage"	57
Figure III. 2 : Grafcet de "Formation de caisse"	57
Figure III. 3 : Grafcet de "Forçage piston"	58
Figure III. 4 : L'organigramme pour création d'un projet sous TIA PORTAL V16	61
Figure III. 5 : Configuration matérielle	62
Figure III. 6: Configuration de l'appareil	63
Figure III. 7: Les tableaux des variables d'API.....	63
Figure III. 8 : La table des variables des entrées d'API	64
Figure III. 9: La table des variables des sorties d'API	64
Figure III. 10 : La table des étapes	65
Figure III. 11 : La table des transitions.....	66
Figure III. 12 : La table des variables de la partie PRL.....	66
Figure III. 13: Etapes d'affichage du tableau d'affectation.....	67
Figure III. 14 : Tableau d'affectation	67
Figure III. 15 : DB global « DB_INPUT ».....	68
Figure III. 16 : DB d'instance « LES TEMPOS_DB »	68

Figure III. 17: Les deux premiers transitions	68
Figure III. 18: Les deux transitions t5_6 et t5_8	69
Figure III. 19 : Les deux transitions t8_9 et t9-10_11	70
Figure III. 20 : L'étape initiale x0	70
Figure III. 21 : L'étape x5	71
Figure III. 22 : Les deux transitions t34_35 et t34_36	71
Figure III. 23 : La transition t39_42-40	72
Figure III. 24 : L'étape initiale x30 de GRAFCET "FORMATION_CAISSE"	72
Figure III. 25 : L'étape x32	73
Figure III. 26 : L'étape x39	73
Figure III. 27 : L'étape x44	73
Figure III. 28 : Les transitions t22_20 et t23_20	74
Figure III. 29 : Les t21_23 et t21_23.....	74
Figure III. 30 : L'étape initiale x20 et l'étape x21 de GRAFCET "FORCAGE_PISTON"	75
Figure III. 31 : Exemple d'un réseau Ladder d'une sortie	75
Figure III. 32 : Configuration des alarmes TOR	76
Figure III. 33 : Alarme analogique	77
Figure III. 34 : Le bloc FB1 des temporisations utilisées.....	79
Figure III. 35 : Le bloc OB1	80
Figure III. 36 : La partie PRL	80
Figure III. 37 : Quelques exemples des étapes	83
Figure III. 38 : La table des mots déclarés	83
Figure III. 39 : Exemples des transitions.....	83
Figure III. 40 : Exemples des étapes.....	84
Figure III. 41 : Le bloc « JMP_LIST ».....	85
Figure III. 42 : Exemples des étapes	86
Figure IV. 1 : Adresse Ethernet	87
Figure IV. 2 : Création d'un nouveau modèle	88
Figure IV. 3 : Eléments graphiques	88
Figure IV. 4 : Modèle créé	89
Figure IV. 5 : Vue Slide-in droite	89
Figure IV. 6 : Vue d'accueil	90
Figure IV. 7 : Vue de la machine	90
Figure IV. 8 : Fiche technique de l'automate SIEMENS S7 1200	91

Figure IV. 9 : Configuration d'un bouton "First Page"	91
Figure IV. 10 : Vue Gestion des utilisateurs	92
Figure IV. 11 : Configuration du vue « Gestion des utilisateurs »	93
Figure IV. 12 : Vue de Menu	93
Figure IV. 13 : Table des variables externes (les sorties)	94
Figure IV. 14 : Table des variables internes	94
Figure IV. 15 : Configuration d'une étape	95
Figure IV. 16 : Etapes de configuration du bouton qui affiche la vue textuelle	95
Figure IV. 17 : Etape de configuration d'une transition	96
Figure IV. 18 : Vue GRAFCET de « Dépilage_Transfert_Collage »	96
Figure IV. 19 : Configuration des boutons « Vidéo »	97
Figure IV. 20 : Exemple de lecture d'une vidéo	97
Figure IV. 21 : Vue GRAFCET de « Formation_Caisse »	98
Figure IV. 22 : Vue GRAFCET de « Forçage_Piston »	98
Figure IV. 23 : Vue générale	99
Figure IV. 24 : Etapes pour ajouter un module DI/DQ	100
Figure IV. 25 : Tableaux qui montre les 2 variables ajoutées	100
Figure IV. 26 : Réseau d'initialisation des GRAFCETs	101
Figure IV. 27 : Le Bouton arrêt d'urgence	101
Figure IV. 28 : Désactivation des étapes	101
Figure IV. 29 : Vue générale avant l'initialisation	102
Figure IV. 30 : Vue générale après l'initialisation	102
Figure IV. 31 : Vue générale après l'appui sur « AU »	103
Figure IV. 32 : Vue classes d'alarmes	104
Figure IV. 33 : Types des alarmes	104
Figure IV. 34 : Vue des alarmes TOR	105
Figure IV. 35 : Archivage des alarmes	106
Figure A. 1 : Les composants d'automate TSX Micro	109
Figure A. 2 : Exemple d'un réseau en langage à contact	110
Figure A. 3 : Exemple de programme en langage liste d'instructions	1110
Figure A. 4 : Configuration logiciel	111
Figure A. 5 : Vue de menu	112
Figure A. 6 : Configuration matériel	113
Figure A. 7 : Configuration de GRAFCET	113

Figure A. 8 : Exemple d'un traitement Postérieur	114
Figure A. 9 : Exemple de configuration d'une temporisation.....	115
Figure A. 10 : Exemple de gestion de la durée des temporisateurs	115
Figure A. 11 : Table des entrées.....	116
Figure A. 12 : Table des sorties	116
Figure A. 13 : La partiePR120.....	120
Figure A. 14 : La partie chart	122
Figure A. 15 : La partie POST	124
Figure B. 1 : Les transitions du GRAFCET « Dépilage_Transfert_Collage ».....	127
Figure B. 2 : Les étapes du GRAFCET « Dépilage_Transfert_Collage ».....	130
Figure B. 3 : Les transitions du GRAFCET « FORMATION_N_CAISSSE »	133
Figure B. 4 : Les étapes du GRAFCET « FORMATION_CAISSSE »	136
Figure B. 5 : Les transitions du GRAFCET « FORCAGE_PISTON »	137
Figure B. 6 : Les étapes du GRAFCET « FORCAGE_PISTON ».....	138
Figure B. 7 : Les sorties d'API sous TIA PORTAL	140
Figure B. 8 : Le bloc FC5 des alarmes	141

Liste des tableaux

Tableau I.1 : Les composants principaux de la machine SECS	7
Tableau I.2 : Dimensions en mm minimales et maximales de la découpe	8
Tableau I.3 : Tableau représentant dimensions des venteuses et ses coupelles (selon la découpe en carton)	13
Tableau 4 : Composition de l'unité Filtr15	
Tableau 5 : Table des CAXn et CDXn.....	60
Tableau A. 1 : Les composants d'automate TSX Micro.....	109

Liste des abréviations

API	Automates programmables industriels
AC	Alternative current
AU	Arrêt d'urgence
CC	Courant Continu (Continu Current)
CEI	Commission électrotechnique internationale
CONT :	Langage Contacts
CPU:	Central Processing Unit
DB :	Bloc de Données
DC :	Courant Direct
DI/DQ :	Entrée/Sortie
E/VAC	Ethylene Vinyl Acetate
E/S :	Entrée/Sortie
FAST :	Fonction Analysis System Technique
FB :	Bloc Fonctionnel
FBD :	Function Block Diagram
FC :	Bloc de Fonctions
FLR :	Filter,Regulator, and Lubricator
GRAFCET :	Grphe Fonctionnel de Commande Etape Transition
I :	Entrée (input)
IHM :	Interface Homme-Machine
IL :	Instruction List
Init :	Initialisation
IP :	Internet Protocol
LD :	Ladder
LED	Light Emitting Diode
M :	Mémoire
MAS	Moteur Asynchrone
MPI	Interface multipoints (Multipoint Interface)
OB	Blocs d'Organisation
PS :	Power Supplay
PROFINET :	Process Field Network
Q :	Sortie (output)

RAM : La mémoire vive (Random Access Memory)
Rly : Relay
RN 9, RN26 Route Nationale 9, Route Nationale 26
ROM : La mémoire morte (Read Only Memory)
S : Système
SADT : Structured Analysis and Design Technique
SCL: Structured Control Language
SFC : Sequential Function Charts
ST : Littéral
TCP : Transmission Control Protocol
TGBT : Tableau Général Basse Tension
TIA PORTAL : Totaly Integrated Automation PORTAL
TM : Temporisation
TOR capteurs Tout Ou Rien
TSX : Référence automate de marque schneider.
V : Volt
UC : Unité Centrale
X: Eape
W : Mot (word)

Chapitre I : Introduction Générale

Introduction générale

Depuis l'introduction des automates programmables, l'industrie a connu une expansion révolutionnaire, entraînant des avancées technologiques très importantes. Aujourd'hui, dans tous les pays, les APIs sont utilisés par l'industrie manufacturière car elles offrent des avantages considérables par rapport aux limites qui existent dans ce domaine. Ils assurent également la protection des individus contre les risques et les limitations résultant de facteurs externes environnementaux et des équipements utilisés dans les installations industrielles.

L'évolution continue de la fabrication moderne et les progrès technologiques rapides poussent les entreprises à adopter des solutions d'automatisation plus complexes et plus efficaces. CEVITAL, en tant que leader dans le marché algérien, a bâti sa réputation sur l'excellence et l'innovation. Pour maintenir sa position dominante et répondre aux exigences croissantes du marché, il est crucial pour l'entreprise de suivre de près les évolutions technologiques dans son secteur d'activité.

Dans le cadre de notre travail notre principal objectif est l'automatisation d'une barquetteuse équipée d'un automate Schneider TSX Micro, qui ont été largement utilisés dans le passé, arrivent désormais en fin de vie. Leur obsolescence croissante pose des défis majeurs aux entreprises qui en dépendent encore de leurs opérations. C'est pour cela, la migration vers des automates plus modernes, tels que le Siemens S7 1200, devient une nécessité impérieuse pour assurer la continuité opérationnelle, la sécurité et la compétitivité.

Le travail que nous avons réalisé dans le cadre de ce projet est présenté en quatre chapitres selon la démarche suivante :

Dans le premier chapitre, nous présenterons d'abord l'entreprise CEVITAL et les différentes parties de la barquetteuse ainsi que ses équipements, y compris les équipements pneumatiques et électriques, les actionneurs et les capteurs, L'étude de ces composants a pour but d'obtenir une compréhension technique approfondie de l'équipement de la machine et de démontrer leur importance pour son bon fonctionnement.

Le deuxième chapitre sera consacré à la description des APIs, y compris la partie matérielle et logicielle des deux automates Schneider TSX Micro et Siemens S7 1200. Cette description a pour but de voir la différence entre les deux gammes d'automates et donc faciliter l'opération de la migration.

Le troisième chapitre comprend une explication de cahier de charges de la machine ainsi on a traduit ce dernier en GRAFCET, cela nous permettra de trouver quelques solutions pour programmer l'automate S7 1200 en langage LADDER vu que notre automate ne prend pas en charge le GRAFCET en tant qu'un langage de programmation.

Le dernier chapitre se concentre sur la description du WinCC et l'élaboration de la configuration d'un écran tactile avec ce logiciel pour offrir à l'opérateur de la barquetteuse une utilisation meilleure, et un meilleur suivi de l'évolution du processus en remplaçant le pupitre de manœuvre à entrées physiques par une IHM.

Enfin on termine par une conclusion qui va englober l'ensemble de notre étude en indiquant si nos objectifs ont été atteints.

**Chapitre I : Présentation de l'entreprise et des
différentes parties de la machine à étudier**

I.1 Présentation de l'entreprise CEVITAL

Fondé en 1998 par Issad Rebrab, le groupe CEVITAL œuvre pour le rayonnement et le développement de l'économie algérienne en intervenant dans des domaines variés par le biais de ses quatre pôles : le pôle agro-industrie et distribution, le pôle auto motive, immobilier et services, le pôle industrie, ainsi que le pôle verre, fenêtres et fermetures. Le premier projet fut celui d'une raffinerie d'huile végétale. Issad Rebrab cible volontairement les produits de base et de première nécessité pour le consommateur algérien, comme les huiles de table, la margarine, les graisses végétales et le sucre, à l'époque des secteurs largement placés sous monopole d'État, mais ouverts à l'importation.

Le groupe est le fleuron du secteur privé algérien avec des activités très diversifiées et tournées vers l'export. CEVITAL est le premier employeur privé en Algérie et compte aujourd'hui plus de 18 000 collaborateurs qui travaillent quotidiennement à son expansion tant en Algérie qu'à l'international [1].

I.1.1 Historique de l'entreprise

CEVITAL, plus qu'un simple groupe, c'est l'histoire d'une famille unie par la passion de l'entrepreneuriat et le souci de l'excellence. Née le 12 mai 1998, cette épopée a débuté par une activité de conditionnement le 12 décembre de la même année. La pose de la première pierre de la raffinerie a suivi le 17 février 1999, avant son inauguration officielle le 14 août 1999. Depuis ses débuts modestes, CEVITAL s'est imposé comme un géant industriel, bâtissant sa renommée sur des valeurs solides et une vision ambitieuse. La figure suivante représente l'évolution de l'entreprise depuis sa création jusqu'au aujourd'hui :

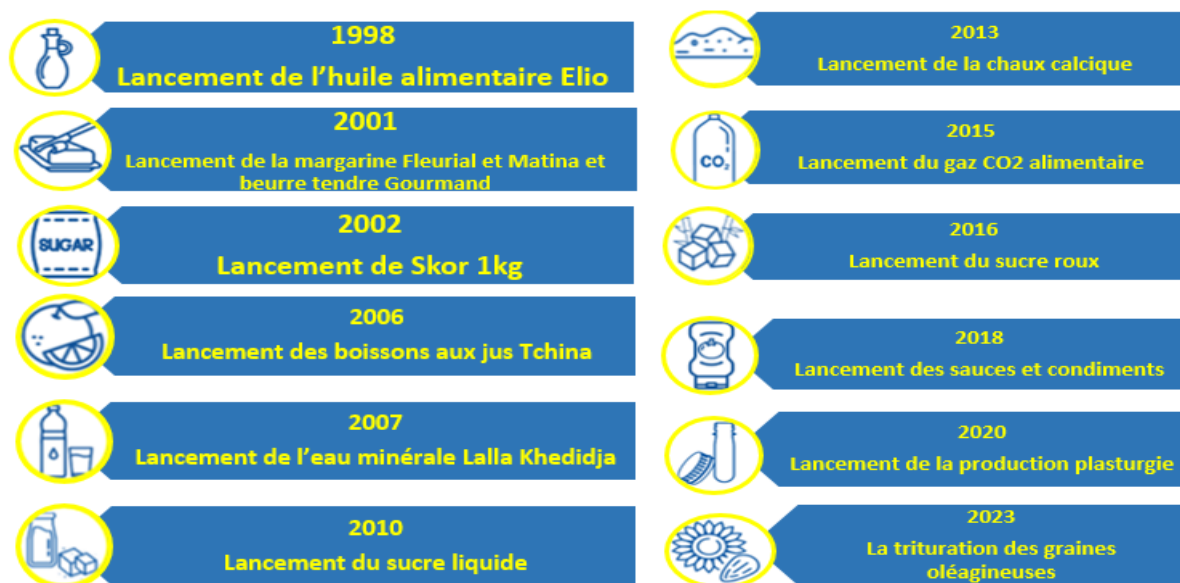


Figure I. 1 : Historique de l'entreprise CEVITAL

I.1.2 Situation géographique

Le complexe Cevital est situé à 3 km sud-ouest de la ville, au quai du port de Béjaïa, près de la RN26 et la RN9. La localisation géographique de l'entreprise constitue un avantage majeur qui lui est bénéfique, car elle lui offre l'avantage de la proximité économique afin de maximiser ses bénéfices. En raison de sa proximité avec le port et l'aéroport de Béjaïa.

Le plus grand complexe privé en Algérie est son complexe de 45 000 m². Grâce à ses silos portuaires, elle dispose d'une capacité de stockage annuelle de 182 000 tonnes, ainsi qu'un terminal de déchargement portuaire de 200 000 tonnes/heure pour la réception de matières premières. Finalement, elle dispose d'un réseau de distribution significatif [2].

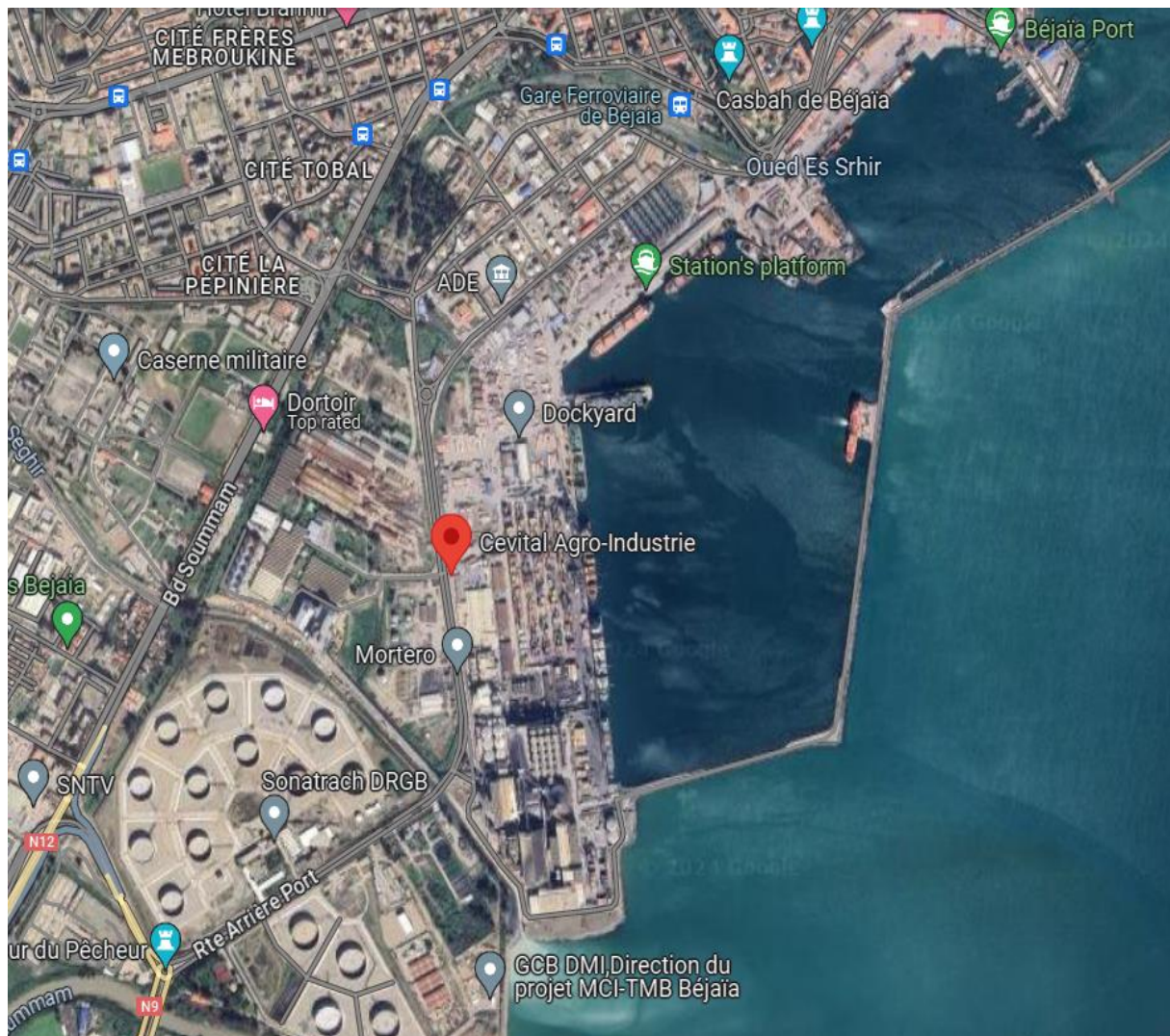


Figure I. 2 : Situation géographique de l'entreprise CEVITAL Béjaïa

I.1.3 Activités de l'entreprise [2]

Les activités de Cevital sont concentrées sur la production et la commercialisation des huiles végétales, de margarine et de sucre. Elle se présente comme suit :

- Raffinage des huiles (1800 tonnes/jour)

Les huiles font aussi parties des plus anciennes productions de Cevital :

- Fleurial : 100% tournesol (depuis 1999).
- Fridor : 100% mélange tournesol, colza et palme.
- Elio : Huile végétale (soja, palme).

- Conditionnement d'huile finis (1400 tonnes/heure).

- Production de margarine (600 tonnes/jour).

- Fabrication d'emballage PET (Poly Ethylène Téréphtalate) (9600 unités/heure).

- Raffinage de sucre (capacité de production de 1600 tonnes/jour)

C'est le produit phare de Cevital il est disposé comme suit:

- Le sucre blanc : Les sacs de 50kg et des big bag de 1000 kg.
- La mélasse: C'est un résidu du processus de raffinage du sucre roux, destiné essentiellement à l'exportation.

- Stockage céréales (120 000 tonnes).

- Minoterie et savonnerie en cours d'étude.

- Sucre liquide.

- Station de traitement des pâtes de la neutralisation.

- Station d'épuration des eaux usées.

- Eau minérale et boissons.

- La margarine, Il y'a deux types de margarines:

- **Margarine de table**, Cevital dispose de 03 modèles de margarine de table :

- Matina en barquettes de 400g et en plaquettes de 250g.
- Fleurial en barquettes de 500g et en plaquettes de 250g.
- Rania en barquettes de 400g et en plaquettes de 250g.

- **Margarine de feuilletage**, La margarine de feuilletage existe en un seul format qui est

- La parisienne en plaquettes de 500g.

I.1.4 Missions et objectifs

L'objectif principal de l'entreprise est de renforcer la production et de garantir la qualité et le conditionnement des huiles, des margarines et du sucre à des prix beaucoup plus

Chapitre I. Présentation de l'entreprise et des différentes parties de la machine à étudier

concurrentiels, dans le but de répondre aux attentes du client et de le fidéliser. Les objectifs de Cevital peuvent être résumés de la manière suivante :

- L'élargissement de ces produits à travers tout le pays.
- L'importation de graines oléagineuses pour extraire directement les huiles brutes.
- L'amélioration de ces opportunités d'emploi sur le marché du travail.
- L'encouragement des agriculteurs à produire localement des graines oléagineuses.
- La modernisation de ces installations en matière de machines et de techniques afin d'accroître le volume de production.
- La position de ces produits sur le marché international grâce à leurs exportations.

Les données économiques nationales récentes sur le marché de l'agroalimentaire indiquent que les entreprises les plus performantes sont celles qui contrôlent efficacement et de manière optimale les coûts, les charges et offrent le meilleur rapport qualité/prix. Il est essentiel pour CEVITAL de se positionner sur le marché en s'associant avec les grandes entreprises commerciales internationales. Ces produits sont commercialisés dans diverses villes africaines telles que Lagos, Niamey, Bamako, Tunis, Tripoli...

I.2 Différentes parties de la machine à étudier

La barquetteuse est une machine essentielle dans de nombreuses industries, notamment l'agroalimentaire, la cosmétique et la pharmacie. Elle joue un rôle crucial dans l'emballage et la manutention des produits, en offrant de nombreux avantages qui en font un outil indispensable pour une production efficace et rentable.

I.2.1 Description générale

La machine est constituée d'un bâti inférieur mécano soudés, et d'une partie supérieure en tôle vissée reposant sur le sol par le biais de pieds réglables en hauteur (en option bâti en inox). L'encollage des découpes et la mise en forme du carton s'effectue à l'intérieur du bâti. L'arrivée des découpes provient du magasin situé à l'arrière de la machine. Tandis que l'évacuation des cartons formés se fait par l'avant de la machine. L'armoire électrique de commande de la machine est située à gauche de la machine. La commande de puissance pneumatique se trouve sur le côté gauche de l'armoire électrique. Les dispositifs de sécurité sont mus par boutons « coup de poings » d'arrêt d'urgence, par un carter fixe plein en partie inférieur et par des portes de protection avec du plexiglas. Les venteuses supportent chacune un poids de 2 Kg maximum.

Chapitre I. Présentation de l'entreprise et des différentes parties de la machine à étudier

Un seul opérateur est nécessaire pour le fonctionnement de la machine, son rôle sera de surveiller le bon fonctionnement du système, il devra aussi assurer le réapprovisionnement du magasin en carton et celui du générateur en colle.

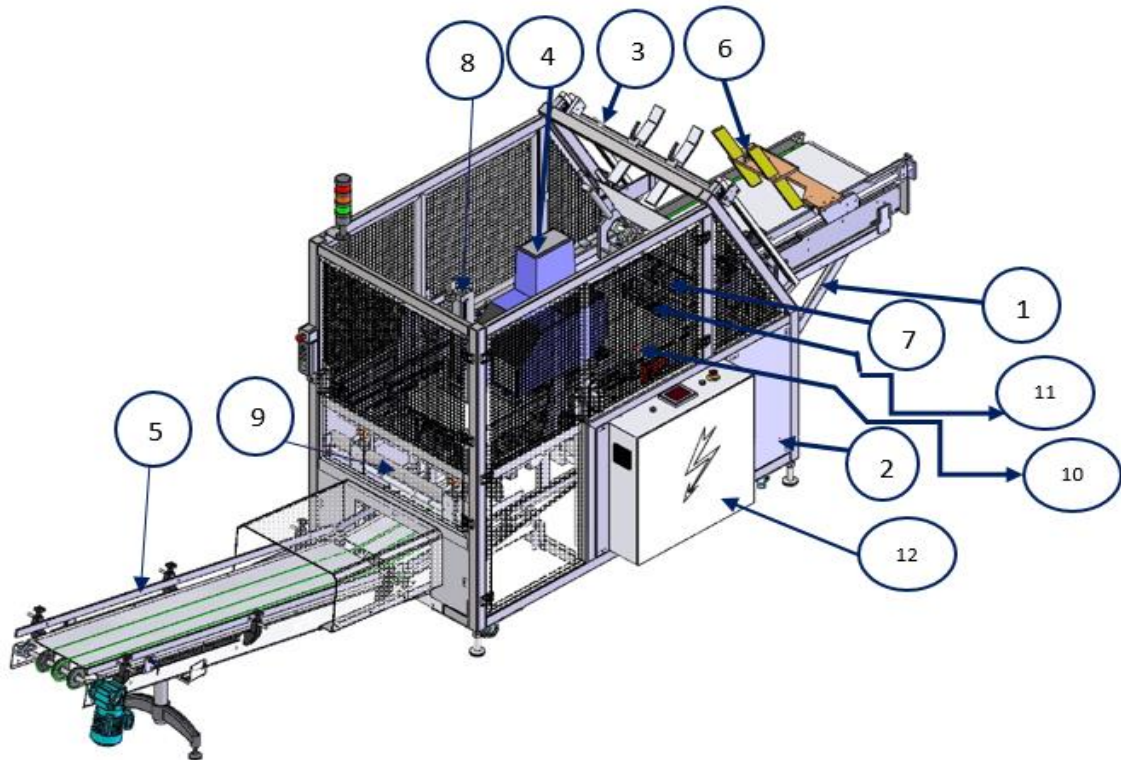


Figure I. 3 : Vue générale de la barquetteuse SECS

Tableau I.1 : Les composants principaux de la machine SECS

REP	Désignation
1	CHASSIS MAGASIN CARTON
2	BATI BARQUETTEUSE ECS
3	CAGE CARTON
4	ENCOLLEUR BARQ. M-NOR-MESA
5	CONVOYEUR CARTON
6	MAGASIN CARTON SYM
7	PRISE CARTON M-FES-IFM
8	PISTON BARQUETTEUSE B14 M-FES-IFM
9	PUIT DE FORMAGE BARQ. M-ST
10	CONVOYEUR DE TRANSFERT
11	TRANSFERT
12	ARMOIRE ELECTRIQUE

I.2.2 Caractéristiques

Longueur : 2.6m **Largeur :** 1.5m **Hauteur :** 1.1m de sol vers le stock de cartons

Masse de la machine équipée : 1000 à 1200 Kg

Chapitre I. Présentation de l'entreprise et des différentes parties de la machine à étudier

Cadence : 1000 à 1300 caisses/heure (selon le format et la qualité du carton)

Capacité du magasin : 800 mm de hauteur (environ 175 découpes)

Dimensions des boîtes : La figure suivante représente les dimensions minimales et maximales d'une boîte à fabriquer :

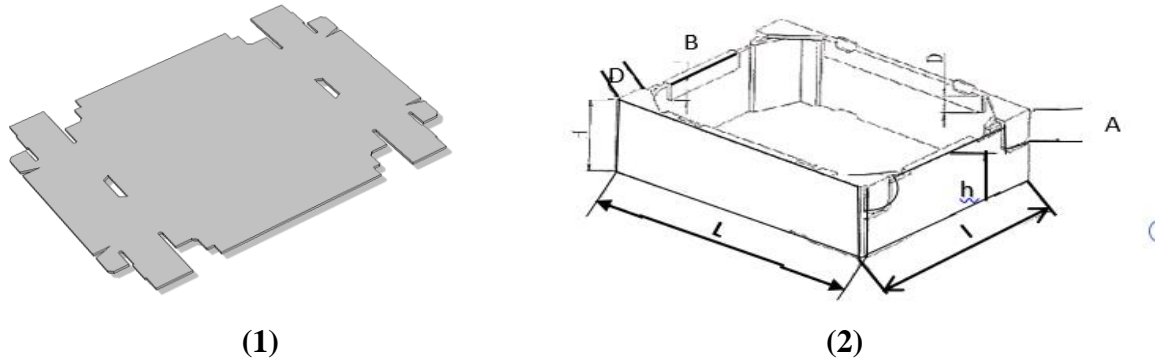


Figure I. 4: Dimensions de la caisse

(1) : Découpe à plat

(2) : Caisse fabriquée

Tableau I.2 : Dimensions en mm minimales et maximales de la découpe

	MIN(mm)	MAX(mm)
L	390	600
l	190	600
H	65	250
h	50	125
A	30	140
B	25	90
C	30	125
D	30	70

I.2.3 Energies et produits mis en œuvre

- **Energie électrique :** 230V/400V 3 Phases + Neutre.
Circuit de puissance 400V, 50 à 60 Hz. 10KW.
Circuit de commande 24V DC.
- **Energie Pneumatique** Air comprimé : 25 à 55 m³/heure asséché et compressé à 6 bars.
- **Energie thermique pour le chauffage de la colle :**
Nature Résistance (générateur, tuyaux et pistolets).
Puissance 4.36 KW
Température 38 à 218°C

I.2.4 La partie électrique

I.2.4.1 Moteurs électriques asynchrones

Notre barquetteuse comporte deux MAS, l'un pour actionner le convoyeur à taquets et l'autre pour faire descendre et remonter le piston de formation de caisse, ces MAS sont appelés aussi moteur à induction, son rôle est la conversion de l'énergie électrique en énergie mécanique grâce à des phénomènes électromagnétiques. Il est constitué de deux parties principales :

- **Le stator** : Il s'agit du composant fixe du moteur. Il est constitué d'une structure ferromagnétique avec trois enroulements électriques. Dans ces enroulements, le courant traverse et génère un champ magnétique à l'intérieur du stator.
- **Le rotor** : Le rotor est la partie tournante d'une machine électrique. C'est sur lui que se trouve l'enroulement électrique (ou le champ magnétique) qui crée le champ tournant [3] ; [4].

La création d'un champ magnétique par le stator (composé de bobines de fils électriques) entraînera le rotor, qui entraînera l'arbre moteur et mettra en marche la machine à entraîner. En particulier, le courant qui circule dans le bobinage du stator va générer un champ de rotation. En raison de l'action de ce champ tournant, le rotor sera entraîné par un courant induit, créant ainsi un autre champ magnétique dans le rotor. Dans cette situation, il y aura une légère variation du champ de rotation du rotor par rapport à celui du stator : c'est le glissement connu.

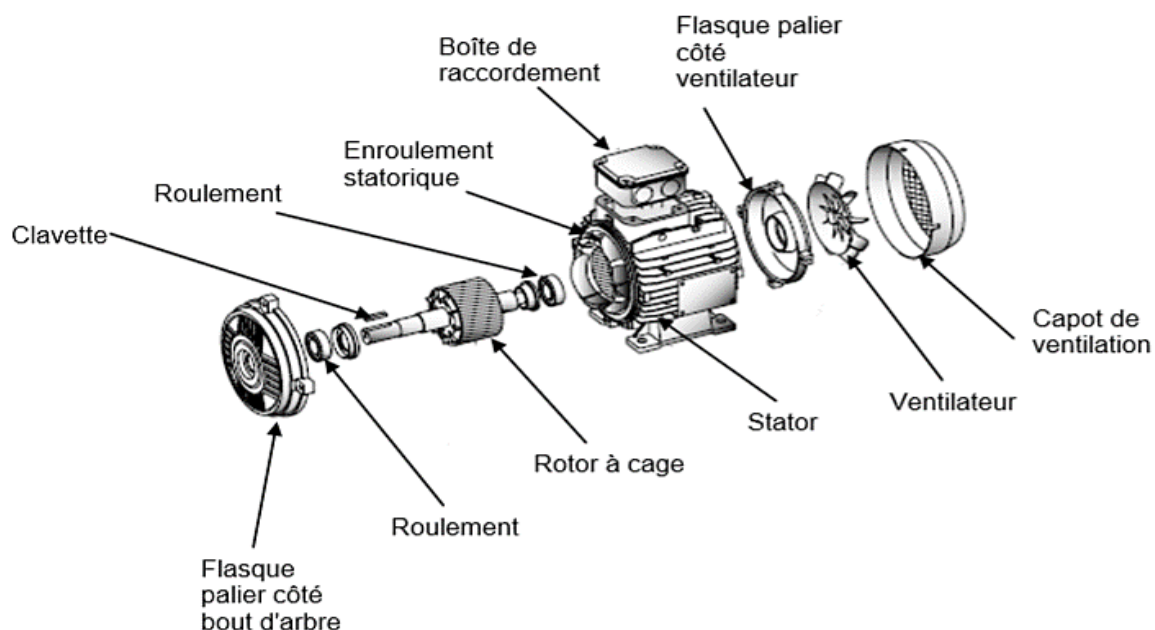


Figure I. 5: Constitution d'un moteur à rotor à cage

Les deux moteurs de notre cartonneuse ont un seul sens de rotation, comme on a besoin de descendre et de remonter le piston au niveau de la cavité de formation de caisse, le moteur piston est associé à un système mécanique « bielle à manivelle », la figure suivante montre le fonctionnement de ce système :

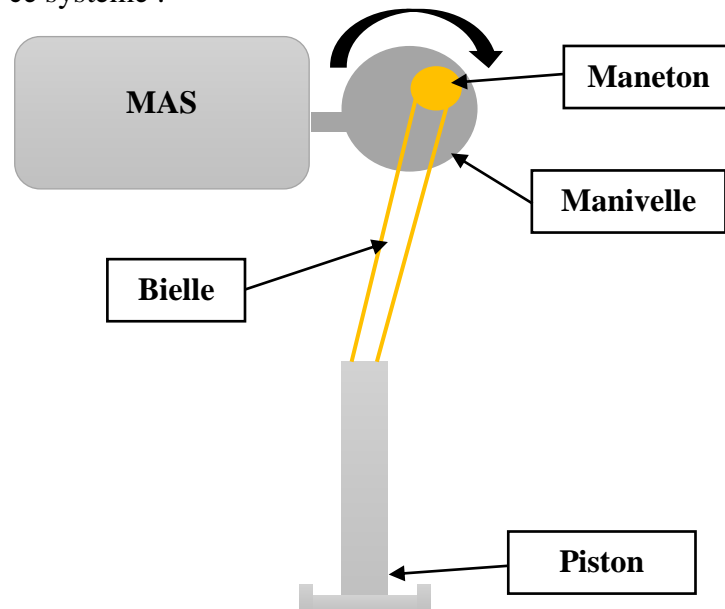


Figure I. 6 : Moteur électrique à un seul sens avec bielle-manivelle.

I.2.4.2 Armoire électrique

Une armoire électrique, aussi appelée tableau électrique, est un coffret métallique qui regroupe divers composants électriques destinés à la distribution, au contrôle et à la protection d'une installation électrique. Elle joue un rôle crucial dans la sécurité et le bon fonctionnement d'un système électrique. Elle contient tous les équipements électriques nécessaires aux fonctionnements et la protection de la barquetteuse (ces équipements sont alimentés à partir de la TGBT qui alimente toutes les armoires électriques) qui sont :

- **Contacteur** : Le contacteur est un dispositif électromagnétique qui assure le bon fonctionnement des moteurs, des résistances ou d'autres récepteurs des puissances importantes grâce à des contacts (pôles) de puissance. Selon le modèle, il est également équipé de contacts auxiliaires intégrés, qu'ils soient ouverts ou fermés. Il est également possible d'ajouter des additifs ou des blocs auxiliaires destinés exclusivement à la télécommande ou à la signalisation [5]. Donc en résumé, son rôle est :
 - **Commander** la mise sous tension ou hors tension d'un circuit électrique.
 - **Isoler** un circuit lorsqu'il n'est pas utilisé, et **Protéger** les circuits contre les surcharges et les courts-circuits.

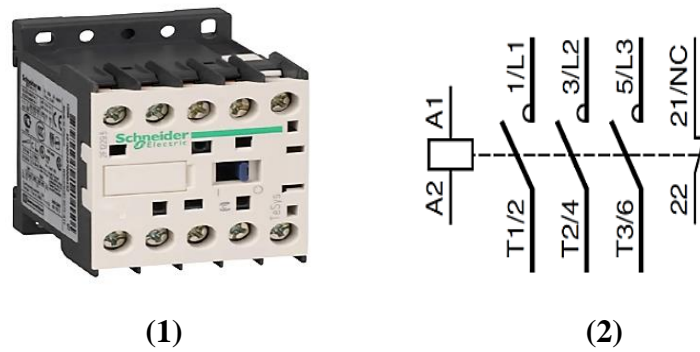


Figure I. 7 : Contacteur électrique

- (1) : Photo d'un contacteur réel
- (2) : Schéma électrique d'un contacteur

- **Relais thermique** : Les relais thermiques assurent la protection des moteurs électriques contre le dépassement des limites de fonctionnement, notamment en cas de surcharges le circuit de puissance n'est pas directement affecté par le relais thermique. Le relais thermique ouvre le circuit de commande d'un contacteur, ce qui interrompt le courant dans le récepteur [6] ;[7].

Le fonctionnement d'un relais thermique repose sur un bilame métallique qui est sensible à la température. Le courant du moteur traverse ce bilame. Quand le courant augmente, la courbure du bilame est due à l'échauffement. Cette courbure provoque l'activation d'un mécanisme de contact qui interrompt le fonctionnement du moteur.

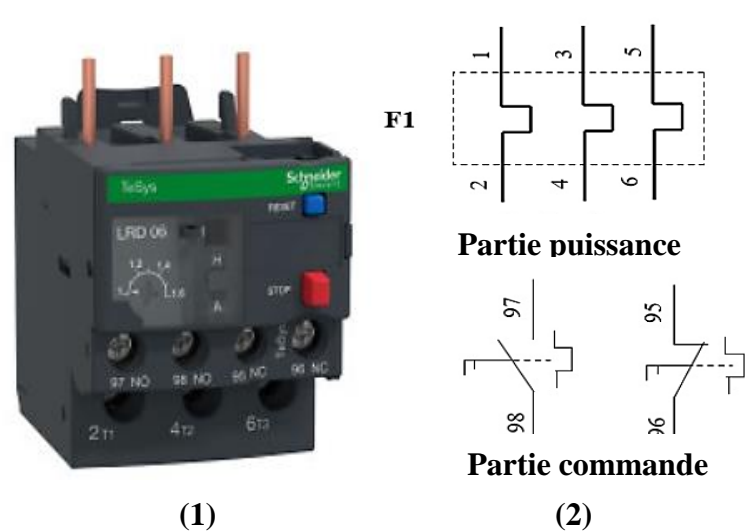


Figure I. 8 : Relais thermique

- (1) Photo d'un relais réel
- (2) Schéma électrique d'un contacteur

- **Sectionneur porte fusible** : Il s'agit d'un dispositif de protection électrique qui permet d'ouvrir ou de fermer un circuit manuellement pour isoler les circuits électriques d'alimentation du réseau [8].

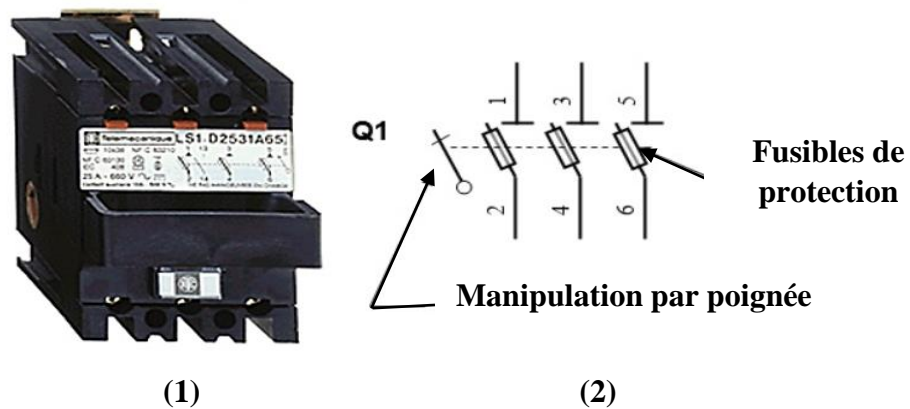


Figure I. 9 : Sectionneur porte fusible

- (1) Photo d'un sectionneur porte fusible réel
- (2) Schéma électrique d'un sectionneur porte fusible

- **Transformateurs**: Réduisent la tension du courant alternatif du secteur à un niveau approprié pour les différents composants de l'armoire.
- **API (Schneider TSX micro)** : L'automate Schneider TSX Micro est un contrôleur programmable industriel (API) compact et modulaire conçu pour les applications de petite et moyenne taille dans les domaines de l'automatisation industrielle et du contrôle de processus. Il appartient à la gamme Modicon de Schneider Electric. L'automate agit comme le cerveau de la barquetteuse, contrôlant et coordonnant l'ensemble des éléments de la machine. Il reçoit des signaux des capteurs (présence de barquettes, positionnement, etc.) et envoie des instructions aux actionneurs (convoyeurs, systèmes de scellage, etc.) pour automatiser les différentes étapes du processus de conditionnement.

I.2.5 La partie pneumatique

I.2.5.1 Les ventouses

La ventouse est utilisée pour manipuler divers produits ayant des surfaces, formes et dimensions variées. La forme des ventouses est déterminée par l'objectif d'utilisation. Dans notre machine on a une ventouse à soufflet qui est utilisée pour la prise pour les charges non planes avec un effet rotule et souvent utilisée pour les charges légères. Il est conseillé d'utiliser les ventouses à soufflets pour saisir les plaques cartonnées en raison de leur forme, de leur flexibilité et de leur capacité à se déformer lorsqu'elles sont maintenues.

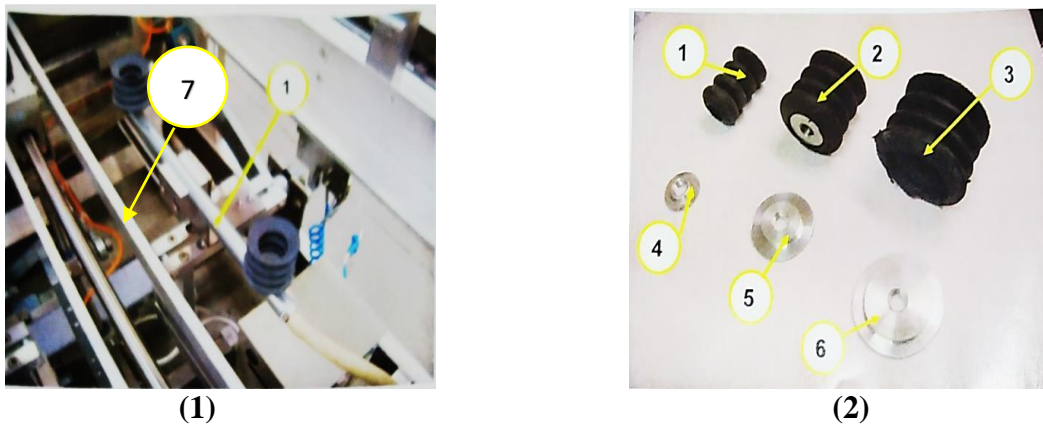


Figure I. 10 : Les ventouses à soufflet

- (1) Photo réel de la ventouse de la cartonneuse
- (2) Différentes ventouses avec ses coupelles

Tableau I.3 : Tableau représentant dimensions des ventouses et ses coupelles (selon la découpe en carton)

REP	Désignation
1	Ventouse diam.35
2	Ventouse diam.50
3	Ventouse diam.65
4	Coupelle ventouse 35
5	Coupelle ventouse 50
6	Coupelle ventouse 65
7	Tube ventouse Diam 20Lg 350

Les ventouses à soufflet sont alimentées soit par :

I.2.5.2 Les pompes à vide électriques

Elles ont le même principe de fonctionnement d'un compresseur en inversant les cotées d'application. On trouve les pompes à palettes et celles à turbines... Ils se caractérisent par la facilité d'alimentation et sa précision. Par contre, elles ont l'inconvénient de leur encombrement et leur prix très élevé.

I.2.5.3 Ejecteur Pneumatique

En se basant sur le principe de l'effet Venturi, l'éjecteur est alimenté en air comprimé. Ce dernier passe à travers une buse, augmentant sa vitesse et créant une dépression dans la canalisation entourant la buse. Ce dispositif de génération de vide se distingue par sa compacité, sa simplicité, son entretien minimal et son coût comparé à celui d'une pompe à vide [9] ; [10].

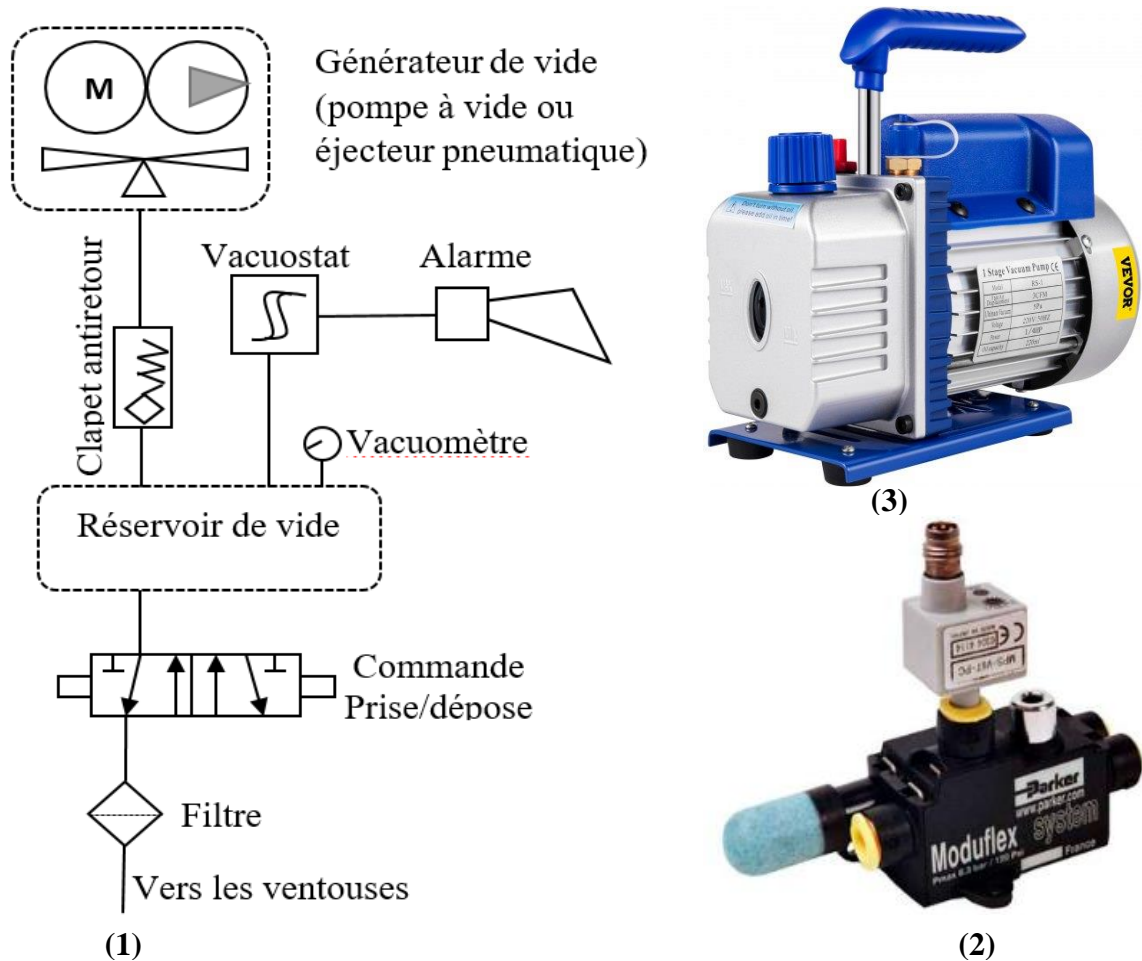


Figure I. 11 : Ejecteur pneumatique et une pompe à vide

- (1) Schéma type d'un circuit de vide
- (2) Ejecteur pneumatique
- (3) Pompe à vide

I.2.5.4 Vérins et dispositifs d'alimentation

Les vérins sont des dispositifs qui convertissent l'énergie d'un fluide en énergie mécanique pour générer principalement un mouvement linéaire. Notre machine est constituée d'un vérin simple effet qui remonte le margeur et deux autre vérins à double effet de pressage lors la de formation de caisse.

- **Les vérins à simple effet** ne possèdent qu'une seule entrée d'air sous pression et exercent une force uniquement dans une direction. Le mouvement de retour à vide est effectué par la détente d'un ressort de rappel intégré dans le corps du vérin.
- **Les vérins à double effet** se distinguent des vérins à simple effet par le fait qu'ils génèrent une force constante à la fois en poussant et en tirant, ce qui leur permet d'effectuer un travail dans les deux sens de déplacement.

La figure suivante montre la différence entre les deux types des vérins :

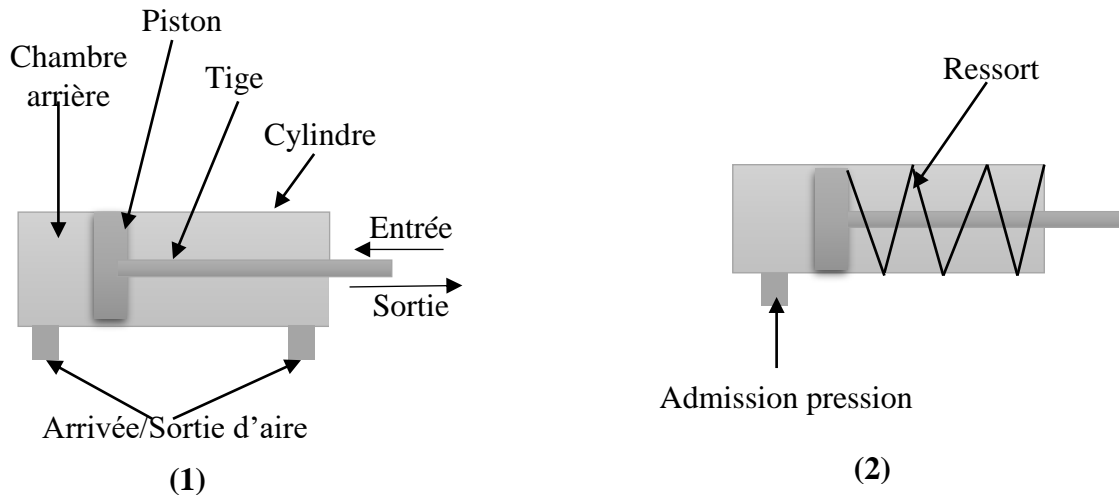


Figure I. 12: Les deux types des vérins

- (1) Vérin double effet
- (2) Vérin simple effet

La barquetteuse est alimentée en énergie pneumatique avec une pression de 6 bars à partir de la station de compresseur à travers d'une unité de conditionnement FLR qui adapte l'énergie pneumatique au système. La figure suivante montre le FLR et ces composants utilisé :

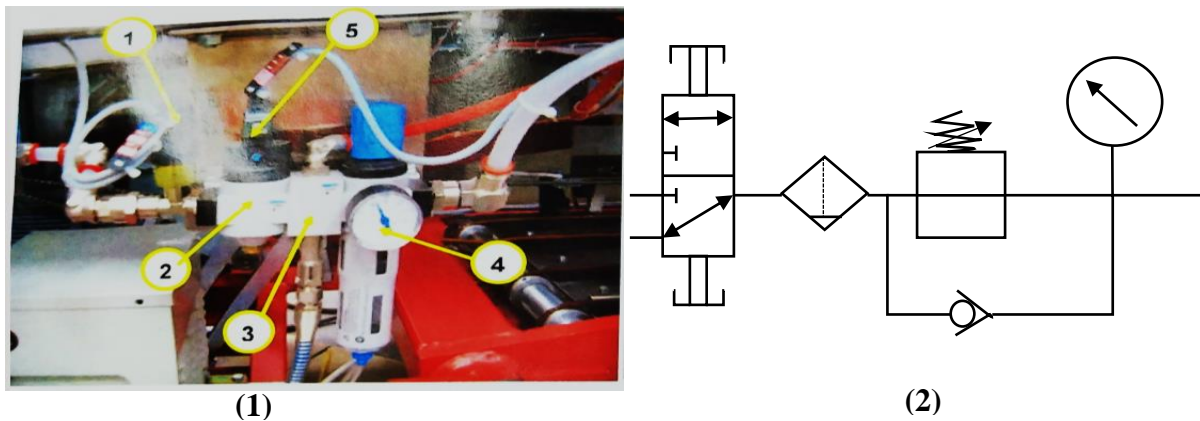


Figure I. 13 : Le FLR FESTO et ces composants
 (1) Photo réelle d'un FLR FESTO (prise en entreprise)
 (2) Représentation pneumatique du filtre FLR FESTO

Tableau 4 : Composition de l'unité Filtre

REP	Désignation
1	Pressostat réf.0166 40703 1 027
2	Vanne arrêt d'urgence réf.HEE-D-MINI-24
3	Bloc dérivation FESTO 162786 réf. FRM-D- MINI
4	Filtre FESTO 159631 réf. LFR-D-MINI
5	Connecteur FESTO réf.MSFG-24/42-50/6034411

Pour assurer le bon fonctionnement et le contrôle de la tige d'un vérin on utilise un dispositif qui contrôle le débit de l'air comprimé « Distributeur ».

I.2.6 Convoyeurs

Un convoyeur est un **mécanisme** ou **une machine** qui permet le **transport de charges** isolées ou de **produits en vrac** de façon **continue** sur un **trajet prédéterminé**. Notre cartonneuse est constituée deux convoyeur.

- **Convoyeur d'entrée (Convoyeur à taquet contrôlé par un codeur)** : Un convoyeur à taquet contrôlé par un codeur est un système de transport de matériaux en vrac qui utilise une série de taquets fixés à une chaîne ou à une courroie pour déplacer le matériau le long d'une auge. La vitesse et la position du convoyeur sont contrôlées par un codeur, qui est un capteur rotatif qui fournit des informations sur la vitesse et la position de la chaîne ou de la courroie. La figure suivante montre un convoyeur à taquets et un codeur qui le contrôle.

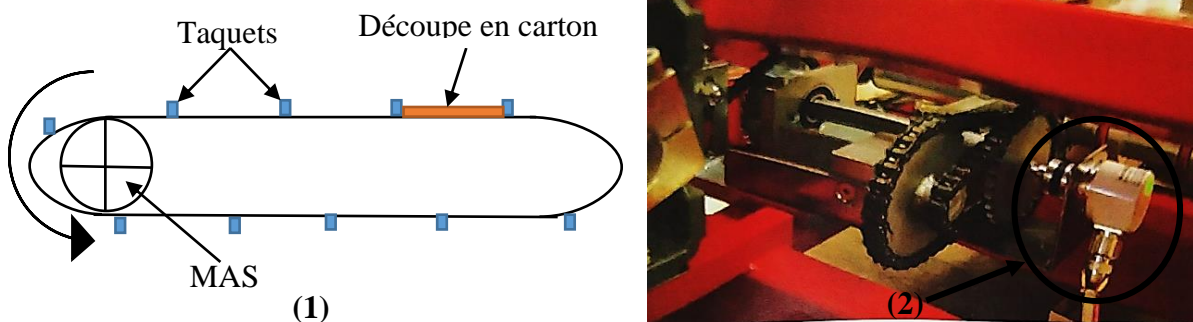


Figure I. 14 : Convoyeur à taquets et son codeur de contrôle
(1) Convoyeur d'entrée (convoyeur à taquets)
(2) Codeur de contrôle (Photo prise en entreprise)

- **Convoyeur de sortie (convoyeur à courroie)** : Le type de convoyeur à courroie le plus courant utilise une courroie en caoutchouc ou en plastique pour transporter des matériaux en vrac ou des pièces individuelles. La courroie est entraînée par un moteur électrique et se déplace sur des poulies à chaque extrémité du convoyeur.



Figure I. 15 : Convoyeur à courroie (Photo prise en entreprise)

I.2.7 Station de colle

La barquetteuse SECS utilise une colle à prise rapide qui est conforme aux normes agroalimentaires. La colle thermo fusible à base d'E/VAC est largement utilisée pour l'emballage cartonné en raison de sa performance et de sa conformité aux normes.

I.2.7.1 Description

L'unité d'alimentation en adhésif (ASU) auquel la barquetteuse SECS est équipée s'agit du ITW Challenger QUATTRO. C'est une unité d'alimentation en colle thermofusible contrôlée par un microcontrôleur et conçue selon les normes métriques. Son panneau de commande « tout en icônes » est convivial pour les opérateurs internationaux.

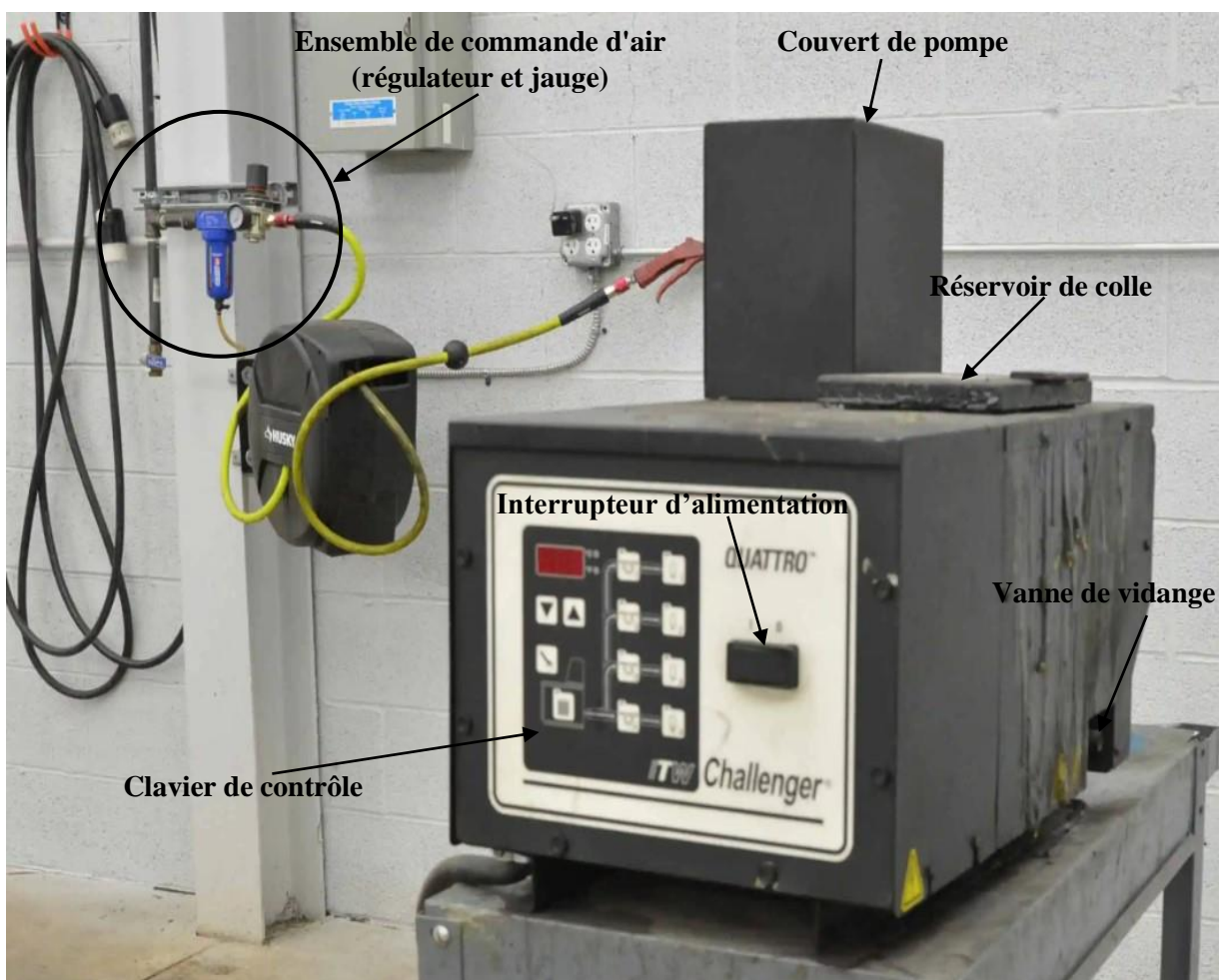


Figure I. 16 : Illustration descriptif des différents composants du ITW Challenger QUATTRO

L'adhésif thermofusible QUATTRO ASU utilise un microprocesseur de contrôle de température afin de maintenir une surveillance étroite de la température de l'adhésif pour un réservoir et jusqu'à quatre tuyaux et quatre têtes. L'opérateur choisit les points de consigne de

température pour jusqu'à neuf zones, et le système émet automatiquement des avertissements en cas d'erreurs de l'opérateur et de problèmes du système.

La température du réservoir, des tuyaux et des applicateurs peut être réglée de manière précise et proportionnelle grâce au système disponible. Il est possible de programmer des délais de chauffage séquentiels afin de mettre en marche les tuyaux et les têtes. Il est possible de programmer une température "de veille" pour maintenir les zones de température à une température plus basse lorsque l'ASU n'est pas en fonctionnement, ce qui permet un retour rapide au fonctionnement normal.

En utilisant ces capacités de programmation de température flexibles, le système QUATTRO permet d'améliorer la durée de vie de l'adhésif en éliminant les températures élevées et prolongées de l'adhésif. Il diminue la consommation d'énergie et permet au système de retrouver rapidement des températures de fonctionnement normales.

La machine mère peut être verrouillée avec des températures d'adhésif prédéfinies, ce qui permet de démarrer automatiquement la production lorsque les températures d'adhésif sont appropriées pour l'application. Il est facile et rapide de programmer toutes les valeurs de température du système.

La pompe à piston à pression constante utilisée par l'ASU est extrêmement fiable. La pompe pneumatique permet de générer une pression d'adhésif élevée à partir d'une entrée d'air comprimé à basse pression.

I.2.7.2 Configuration du régulateur de température

La température du réservoir, des tuyaux et des applicateurs peut être réglée de manière précise grâce au contrôleur de température QUATTRO. Les instructions sont enregistrées sur l'écran tactile. Un message d'erreur sera affiché par le contrôleur chaque fois qu'une situation de capteur ouvert ou en court-circuit survient.

L'ASU utilise le circuit imprimé (PCB). Le réservoir, les tuyaux et les applicateurs fournissent des signaux de commande et de surveillance. Il est équipé de LED allumées pour signaler l'allumage de l'alimentation du chauffage. Cette carte contient les fusibles, les disjoncteurs et la fiche de configuration d'alimentation de l'ASU.

Pour tous les réglages de température, le système emploie des capteurs de détection de température à résistance au nickel (RTD) de 120 ohms.

Si toutes les zones de température sont dans une tolérance programmable, le contrôleur émet un signal prêt et le système est prêt à être mis en production. Le signal prêt contrôle également le fonctionnement de la pompe.

I.2.7.3 Contrôle du débit de colle

La vanne de dosage est l'élément principal responsable du contrôle du débit de colle dans la station QUATTRO challenger, cette vanne régule la quantité d'adhésif appliquée sur le support, garantissant une application précise et uniforme. Elle est pilotée par un système de commande électronique qui reçoit des instructions à partir de l'écran tactile (panneau de commande).

La vanne peut s'ouvrir et se ferme avec précision pour contrôler le débit d'adhésif, la durée d'ouverture de la vanne détermine la quantité d'adhésif.

Le système de commande électronique permet de régler le débit d'adhésif en fonction des besoins de l'application. Cette application s'effectue via des paramètres tels que la pression d'air ou la vitesse d'ouverture de la vanne.

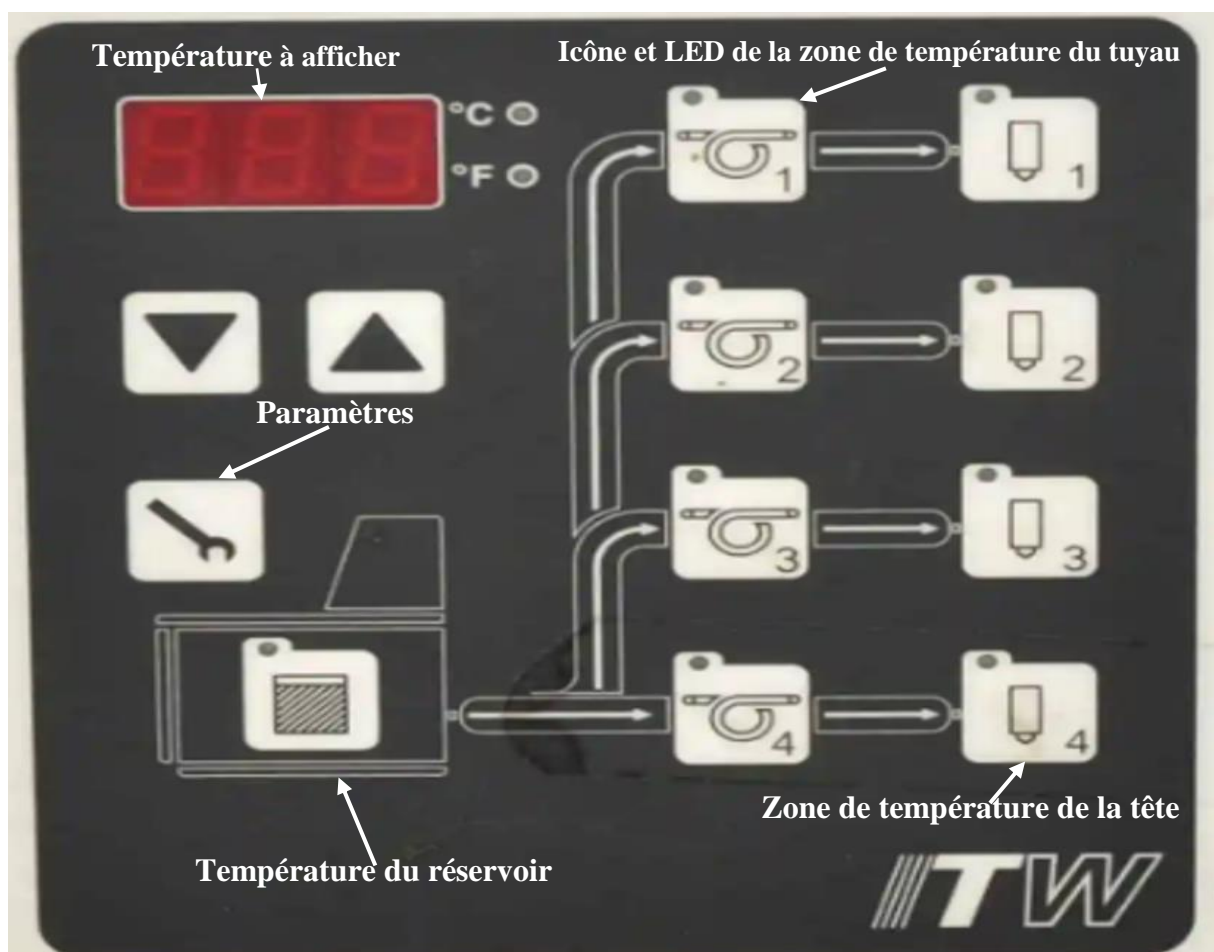


Figure I. 17 : Ecran tactile de commande

I.2.8 Ensemble cavité

Le système mécanique de transfert du carton, du magasin à la cavité de formation, assure le positionnement précis et constant de la découpe pour une mise en forme correcte de la caisse.



Figure I. 18 : Photo réelle des équerres de la barquetteuse (prise en entreprise)

I.2.9 Partie instrumentation

I.2.9.1 Détecteur de proximité

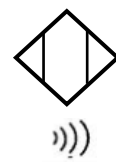
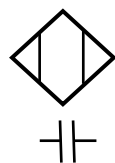
Un détecteur de proximité, connu sous le nom de capteur de proximité, est un appareil électronique qui peut identifier la présence ou l'absence d'objets à proximité sans nécessiter de contact physique direct. Ces dispositifs sont fréquemment employés dans les domaines de l'automatisation industrielle et du contrôle des machines pour diverses applications, telles que :

- **Détection de la présence de pièces:** Contrôler le flux de pièces dans les convoyeurs et les machines, par exemple dans notre cartonneuse on a des capteurs à proximité photoélectrique qui détectent la présence des découpes en carton dans le magasin et aussi dans le convoyeur à taquet.
- **Contrôle de position:** Déterminer la position d'objets en mouvement dans les processus de production, par exemple dans notre cartonneuse on a des capteurs à proximité inductif qui détectent la position de piston (haut ou bas)



(1)

Capteur capacitif Capteur inductif Capteur à ultrason Capteur magnétique



(2)

Figure I. 19 : Capteur à proximité

- (1) Photo réel d'un Capteur à proximité inductif
- (2) Symboles électrique des différents types des capteurs à proximité

I.2.9.2 Détecteur à contact

Encore appelé détecteur de présence à action mécanique, ou interrupteurs de fin de course, C'est un commutateur, commandé par le déplacement d'un organe de commande (corps d'épreuve). Lorsque le corps d'épreuve est actionné, il ouvre ou ferme un contact électrique.

De nombreux modèles peuvent être associés au corps : tête à mouvement rectiligne, angulaire ou multi direction associée à différents dispositifs d'attaque (à poussoir, à levier, à tige). Ce sont des détecteurs TOR (Tout ou Rien) [11].

Dans notre machine il est utilisé pour la détection de la présence de la caisse fabriquée. La figure suivante montre le fonctionnement de ce type des capteurs :



Figure I. 20: Etapes de fonctionnement du détecteur de présence à action mécanique

- (1) Le capteur à contact avant la détection de la caisse fabriquée
- (2) Le capteur à contact pendant la détection de la caisse fabriquée

I.3 Etude de la machine

Notre machine est constituée de trois parties principales qui sont « **Dépilage_transfert_colle** », « **Forçage Piston** » et « **Formation de Caisse** ». L'opérateur **charge les découpes de carton** dans le magasin de la barquetteuse. Les découpes doivent être correctement positionnées et orientées pour une alimentation optimale dans la machine. Et comme notre barquetteuse est une multi format l'opérateur **lit les dimensions du carton** sur l'emballage ou sur un document d'accompagnement, Il peut également utiliser une jauge **de mesure** pour déterminer précisément la longueur, la largeur et la hauteur du carton pour régler les paramètres de magasin selon les dimensions de découpe, si le magasin est rempli, l'opérateur clique sur le bouton début du cycle, le bras margeur prend une découpe et la dépose sur le convoyeur à taquet.

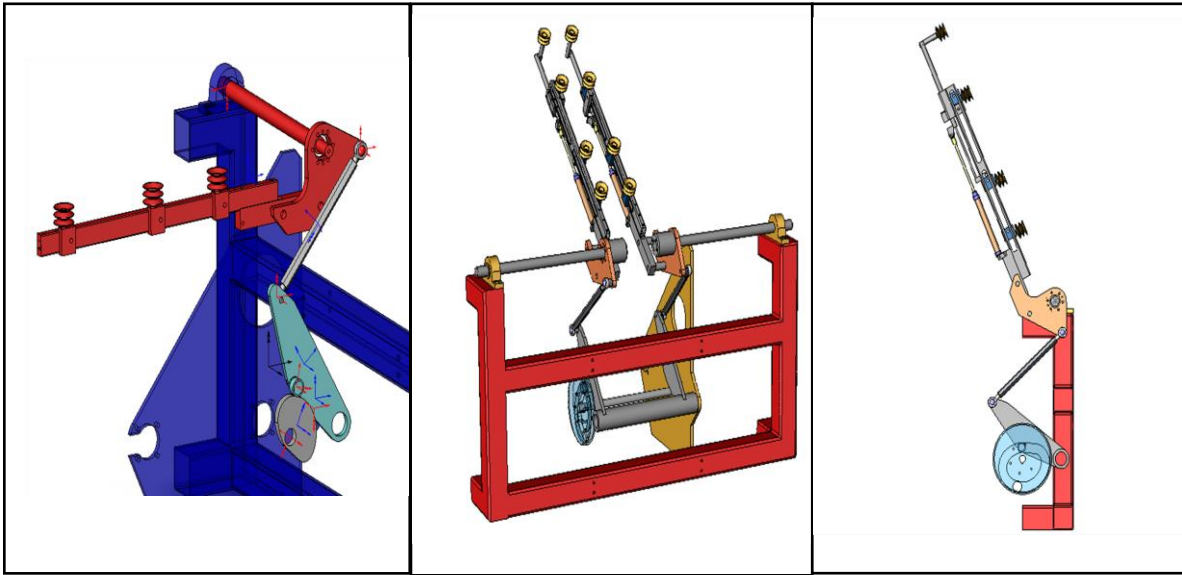


Figure I. 21 : Différentes vues de bras margeur (réalisé sur le logiciel Solid Works)

La figure suivante montre les étapes de stockage et de dépilage :

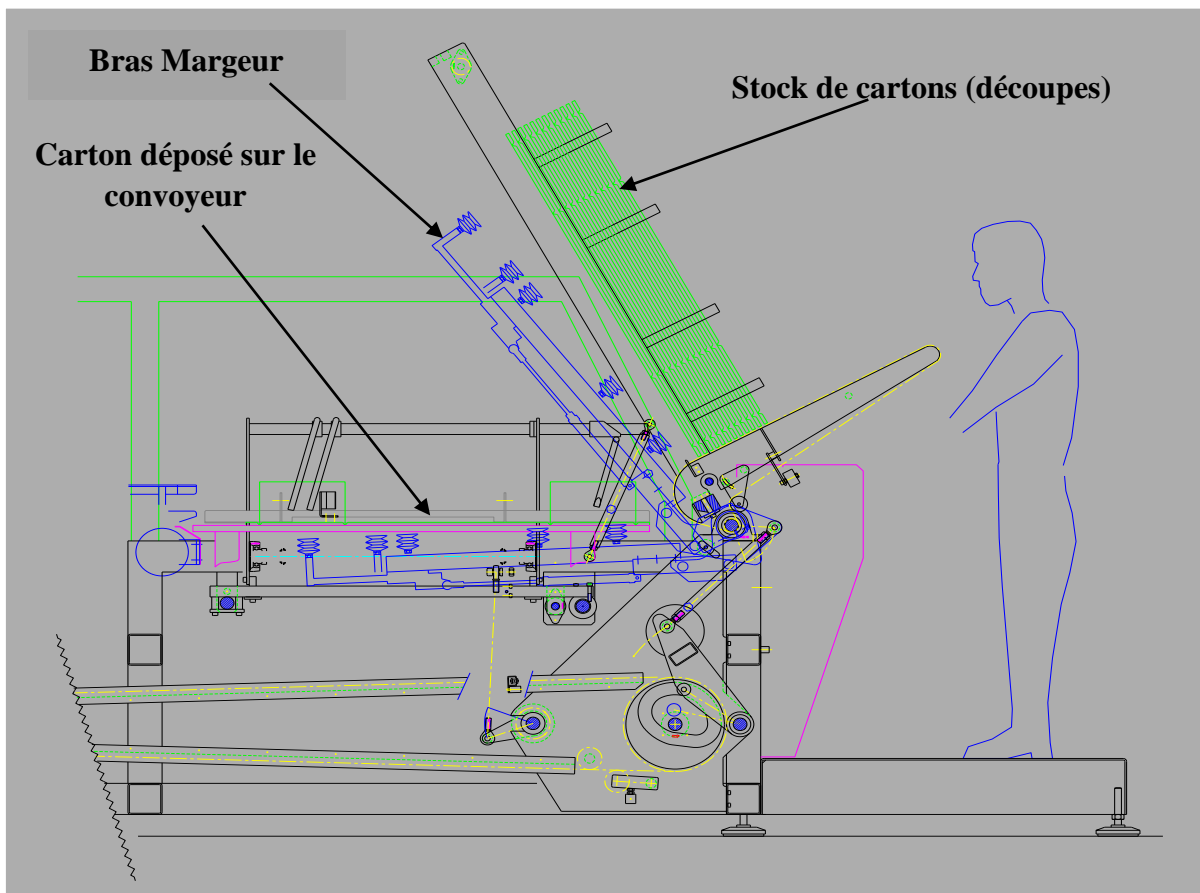


Figure I. 22 : Stockage et dépilage des découpes en cartons

Le convoyeur à taquets transfère la découpe vers la station de collage où les pistolets de colle éjectent une dose de colle chaude sur les deux cotées de la découpe. La formation de caisse se fait à partir de forçage d'un piston vers le bas pour que cette caisse prend sa forme dans la

Chapitre I. Présentation de l'entreprise et des différentes parties de la machine à étudier

cavité à équerres réglables (selon les dimensions souhaités), après deux vérins sort des deux côtés où la colle est appliquée pour la maintenir et assurer le collage. Une fois que le pressage est terminé les deux vérins revient aux ses états initiaux, et la caisse va transférer vers la station de pillage (cette caisse sera détectée par un détecteur à contact) par le convoyeur de sortie.

La figure suivante représente les différentes étapes mentionnées (répartition des postes) :

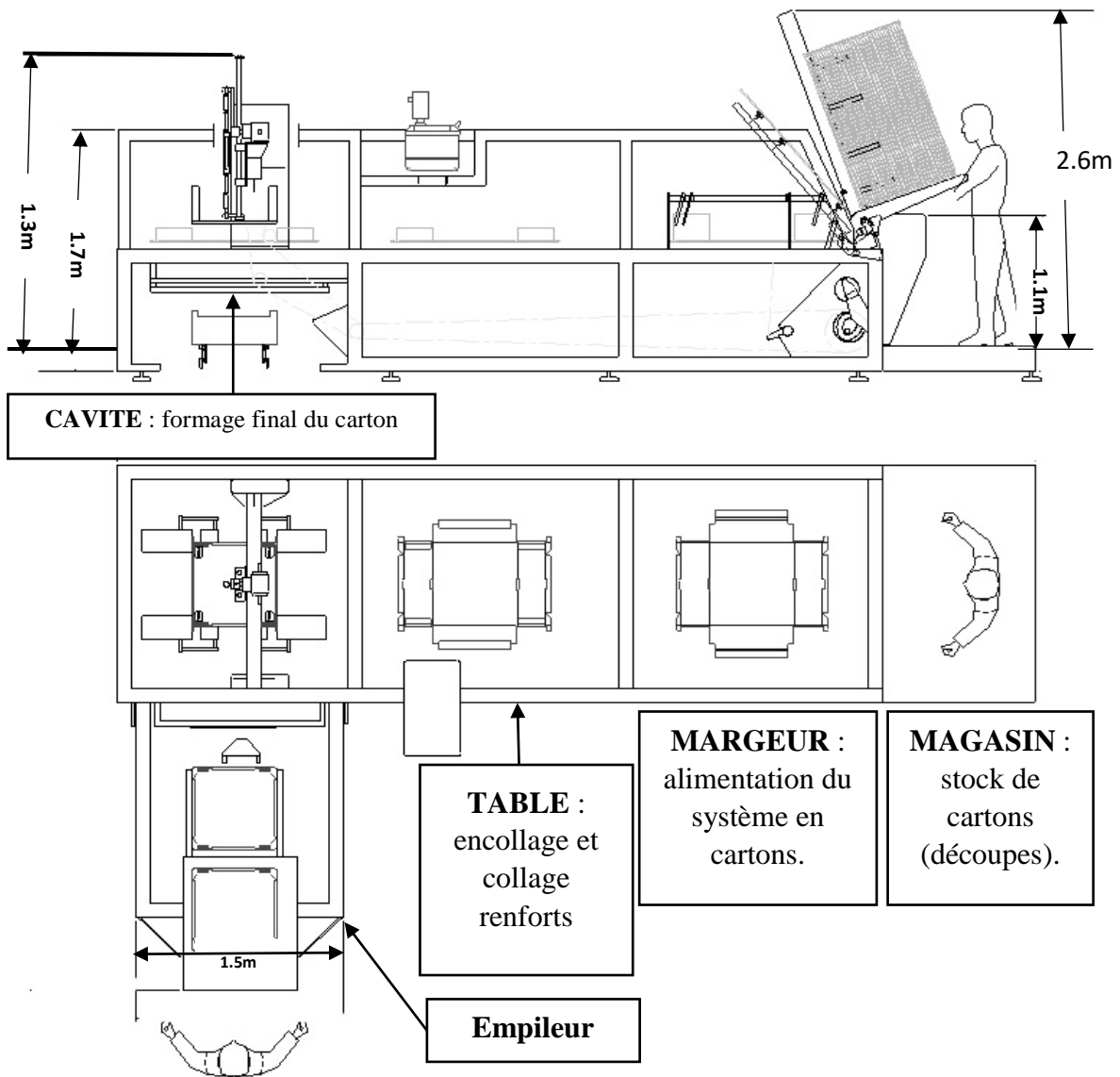


Figure I. 23 : Représentation des postes

Après avoir étudié les différentes parties de la barquetteuse et son fonctionnement, on peut identifier la zone d'étude :

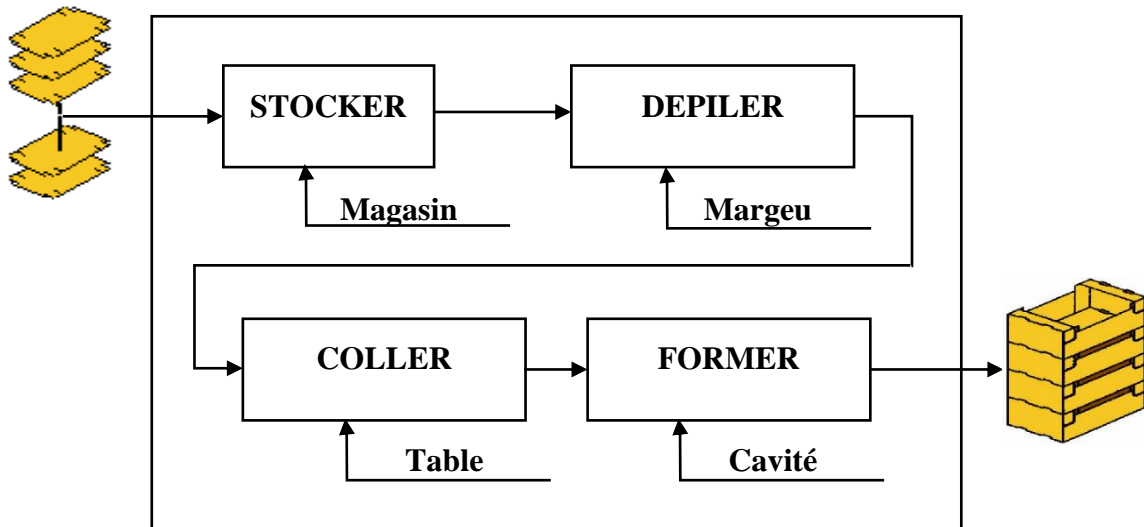


Figure I. 24 : Identification de la zone d'étude

I.3.1 Diagramme SADT niveau A-0 pour la barquetteuse

Le diagramme SADT (Structured Analysis and Design Technique) niveau A-0 joue un rôle crucial dans la compréhension et la conception d'une barquetteuse. Il s'agit d'un outil d'analyse structurée qui représente le système global à un niveau très élevé d'abstraction, en mettant l'accent sur les fonctions principales et leurs interactions.

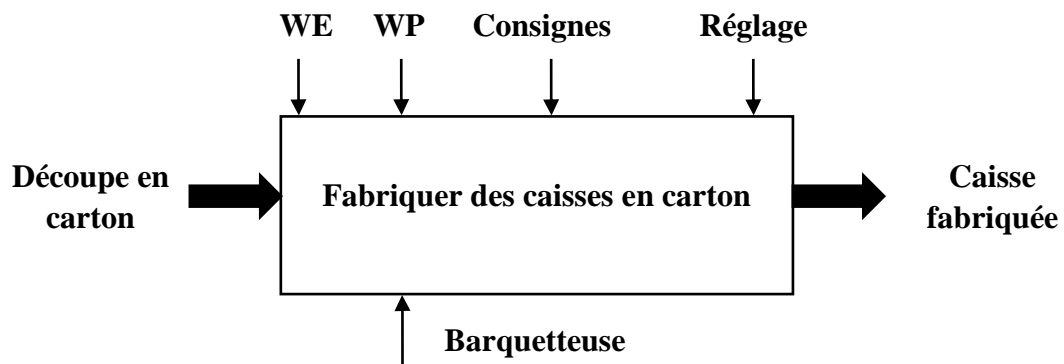


Figure I. 25 : Diagramme SADT niveau A-0 de la barquetteuse

WE : Energie Electrique

WP : Energie Pneumatique

Pour résumer le fonctionnement général de la machine on a suggéré de dessiner le diagramme FAST de la fonction FP1 (fabriquer des caisses à partir d'une plaque carton)

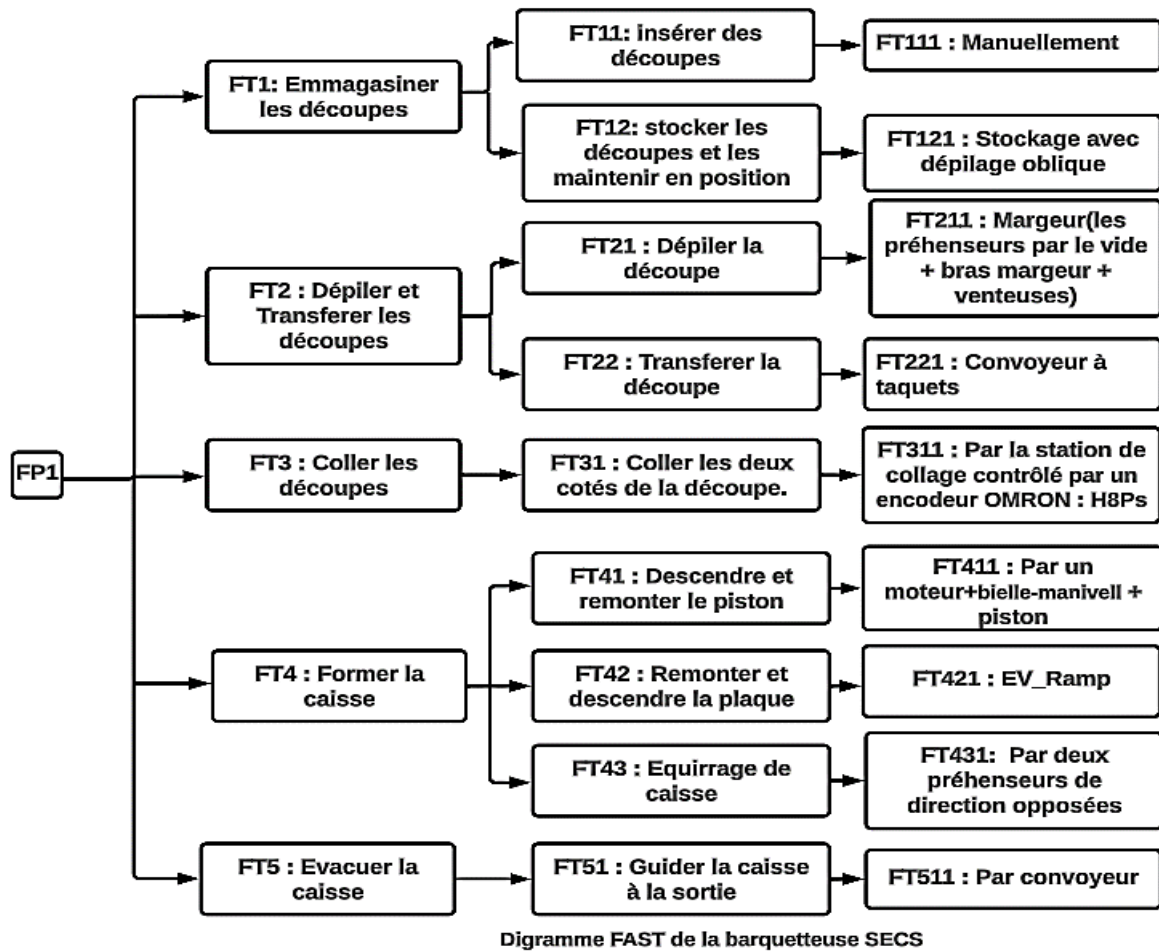


Figure I. 26 : diagramme FAST de la barquetteuse

I.4 Conclusion

L'analyse détaillée des composants mécaniques, électriques et pneumatiques de la barquetteuse, ainsi que de son fonctionnement étape par étape, fournit une base solide pour la rédaction d'un cahier des charges précis et complet, définissant clairement les exigences fonctionnelles et techniques du système d'automatisation. Cette compréhension approfondie facilitera également la programmation en TIA Portal, permettant de développer un code efficace et fiable pour contrôler et piloter la barquetteuse avec précision.

Chapitre II : APIs et logiciels de programmation

II.1 Introduction

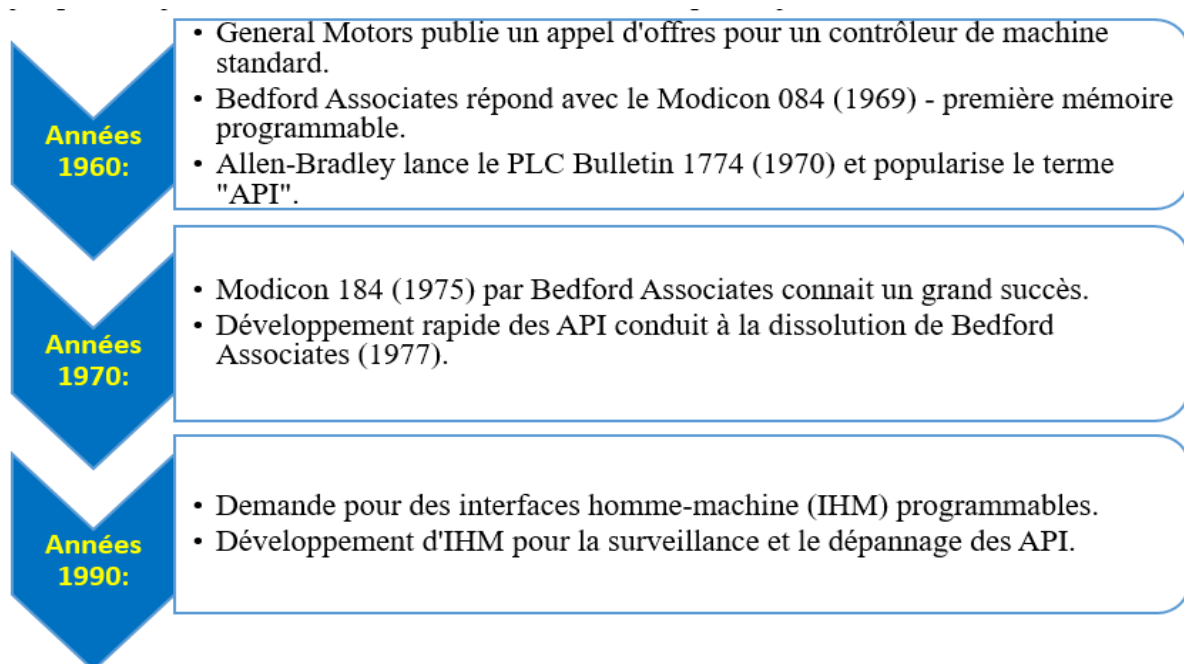
Les APIs sont devenus des éléments essentiels de l'infrastructure informatique moderne, en particulier dans le domaine industriel. Face aux développements technologiques rapides, les APIs jouent un rôle crucial en favorisant l'innovation, la collaboration, l'efficacité, l'agilité et la création de nouveaux modèles économiques. Les entreprises qui adoptent et exploitent efficacement les APIs seront mieux armées pour réussir dans l'environnement industriel en constante évolution.

Dans ce chapitre, on présente l'architecture et le fonctionnement des automates programmables, en commençant par l'automate Schneider TSX micro, par la suite on aborde brièvement la gamme SIMATIC de Siemens, et plus précisément l'automate S7-1200, CPU et modules que nous avons choisis pour la réalisation de notre projet, ainsi les logiciels de programmation.

II.2 Historique

Les Automates Programmables Industriels (API) ont été introduits dans l'industrie principalement pour remplacer les coûteux panneaux de relais, minuteriers et compteurs. Les constructeurs automobiles cherchaient des moyens de simplifier le démarrage des nouvelles lignes de production après chaque changement de modèle annuel, tout en réduisant les coûts d'installation.

Les APIs ont connu un développement considérable dans le temps depuis leur création jusqu'à nos jours le schémas ci-dessous illustre les étapes majeures de l'évolution des APIs :



Schémas II.1 : Développement des APIs à travers le temps

Aujourd'hui, L'évolution des API se poursuit à un rythme accéléré, ouvrant de nouvelles possibilités pour la connectivité, l'innovation et la création de valeur dans un monde numérique de plus en plus connecté. Les API intelligents, les API en temps réel et les API centrés sur le développeur ne sont que quelques exemples de la façon dont ces interfaces puissantes façonnent l'avenir de la technologie [12] ; [13].

La figure suivante montre l'évolution de l'une des marques plus utilisés dans le milieu industriel « SIEMENS » :

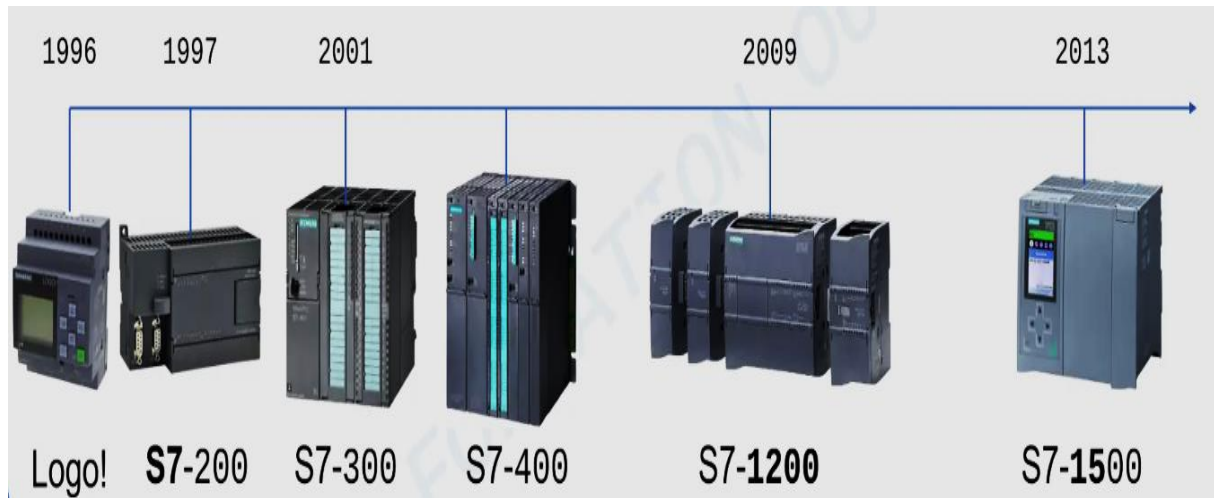


Figure II. 1 : L'évolution des automates SIEMENS

II.3 Qu'est-ce qu'un API ?

Selon la norme française NF 61131 dans sa partie 1 la définition donnée aux Automates programmables est : « Système électronique fonctionnant de manière numérique, destiné à être utilisé dans un environnement industriel, qui utilise une mémoire programmable pour le stockage interne des instructions orientées utilisateur aux fins de mise en œuvre de fonctions spécifiques, telles que des fonctions de logique, de mise en séquence, de temporisation, de comptage et de calcul arithmétique, pour commander au moyen d'entrées et de sorties tout ou rien ou analogiques divers types de machines ou de processus. L'automate programmable et ses périphériques associés sont conçus pour pouvoir facilement s'intégrer à un système d'automatisme industriel et être facilement utilisés dans toutes leurs fonctions prévues ».

A partir de là, et en respectant les caractéristiques essentielles qui ressortent de cette définition, nous considérerons plus simplement comme automate programmable un système :

- Construit autour d'un processeur numérique, spécifique ou non.
- Pouvant être relié à de nombreux signaux physiques.
- Fonctionnant dans des conditions industrielles grâce à une protection adaptée.

- Doté d'un logiciel de programmation permettant un traitement rapide des fonctions logiques (TOR).
- Doté de possibilités d'échanges avec d'autres processeurs.

Ceci constitue un « noyau » minimal susceptible de nombreuses variantes et extensions propres à un automate donné. [14]

II.4 Architecture des automates

II.4.1 Aspect extérieur

Selon le nombre d'E/S, il existe deux types d'API, le monobloc destiné aux applications simples dont le nombre d'E/S est petit ce type caractérisé par un cout faible, moins complexe et pas extensible comparant par le deuxième type modulaire, ce type est important en terme de nombre d'E/S, complexité, prix, et extensibilité.

Un API compact est un dispositif où tous les modules sont dans un seul dispositif. Ce type de PLC a un nombre fixe de modules d'entrée/sortie. L'alimentation, la CPU ainsi que la carte de communication sont aussi insérées dans ce dispositif. On distinguera les modules de programmation (LOGO de Siemens, ZELIO de Schneider, MILLENIUM de CROUZET...) des micro automates.

Un API modulaire est un type de API qui se compose de différents modules (alimentation, CPU, modules E/S ...) où tous ces modules sont installés dans le rack [14] ; [15].

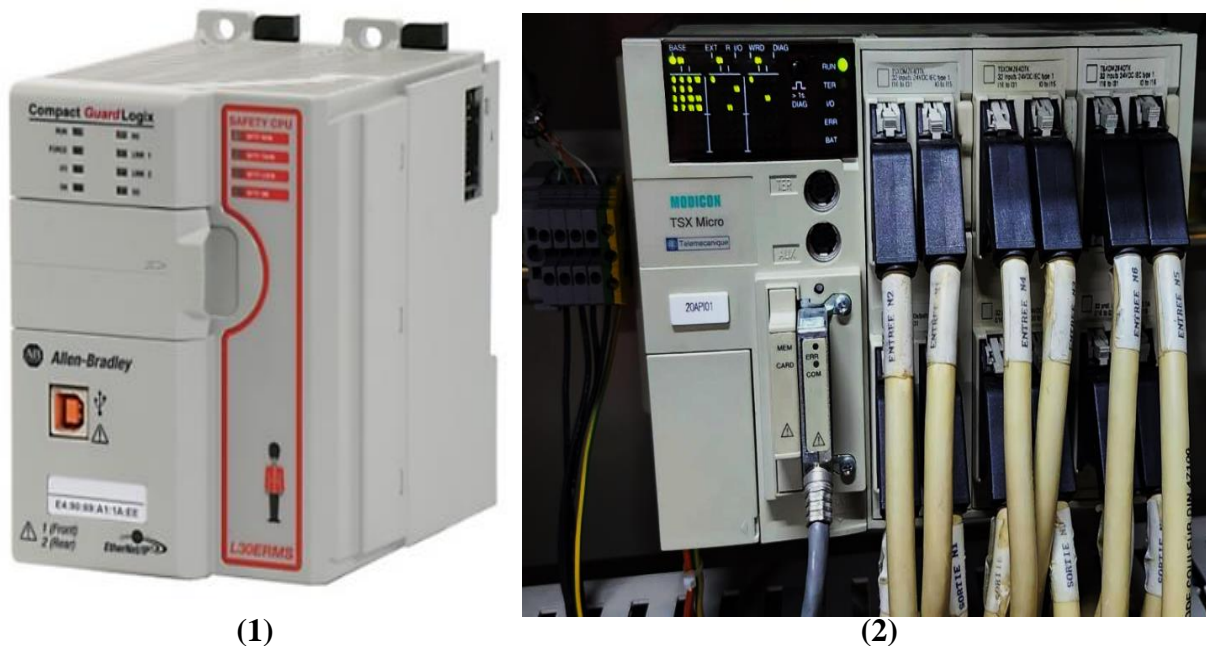


Figure II. 2 : Types des APIs

- (1) Automate compact (Allen-bradley)
- (2) Automate modulaire 'Modicon' (prise en entreprise)

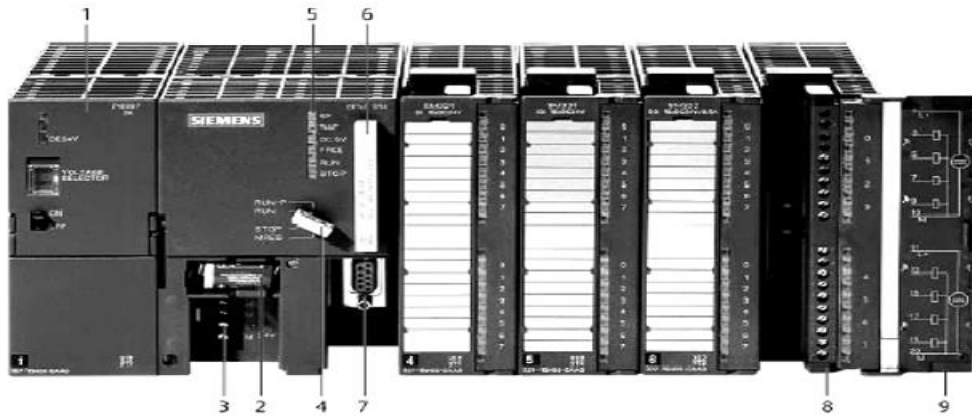


Figure II. 3 : Automate modulaire (Siemens)

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

II.4.2 Structure interne

De manière générale, l'automate programmable industriel est composé de plusieurs éléments de base décrite ci-dessous :

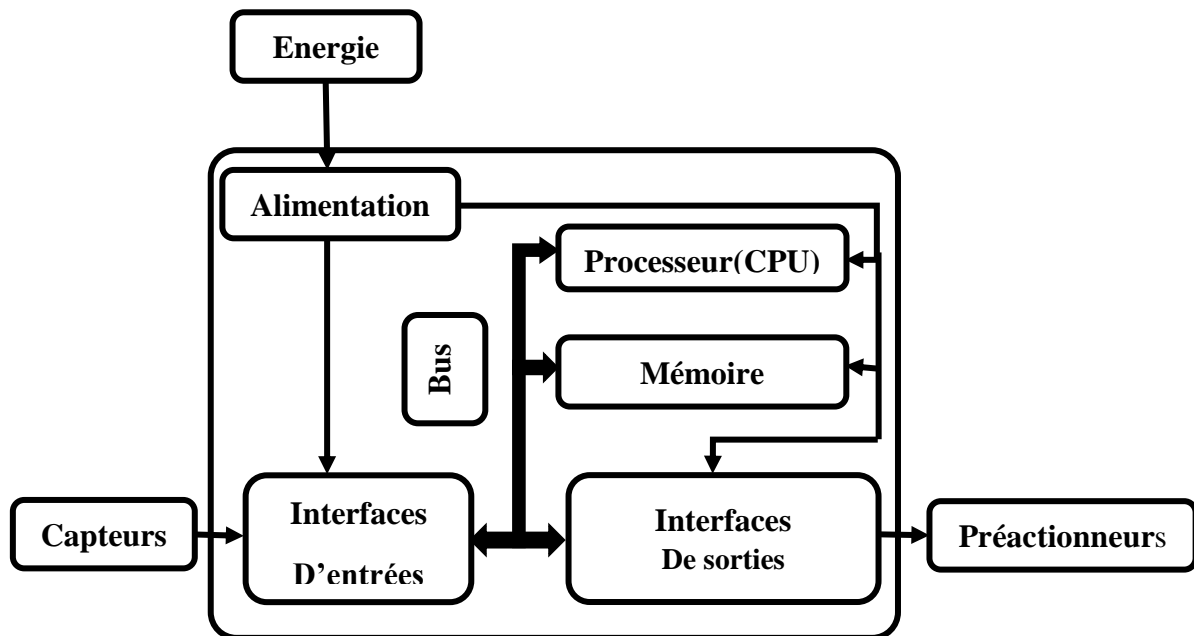


Figure II. 4: Architecture d'un API

L'alimentation : Le bloc d'alimentation (Power Supply (PS)) est indispensable puisqu'il convertit une tension alternative en une basse tension continue (24V,48V...) nécessaire au processeur et aux modules d'entrées-sorties. L'alimentation ne fait pas toujours partie de l'automate qui sera donc directement alimenté par une base tension.

Les alimentations peuvent être de différentes puissances d'un API à l'autre et nécessitent un courant allant de 2A à 50A, en fonction du nombre d'interfaces d'E/S alimentées par elle [16][17].

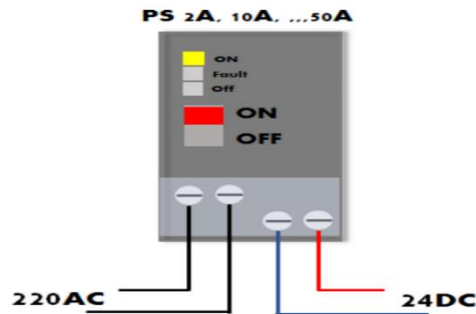


Figure II. 5 : Module d'alimentation d'un API

CPU (Unité centrale de traitement) : également appelée processeur, est l'élément clé du PLC. Elle exécute les tâches nécessaires pour mettre en œuvre les fonctions de l'automate. La CPU est dotée d'un programme qui guide le PLC dans l'exécution des instructions de contrôle. Elle est généralement installée dans la deuxième fente du rack.

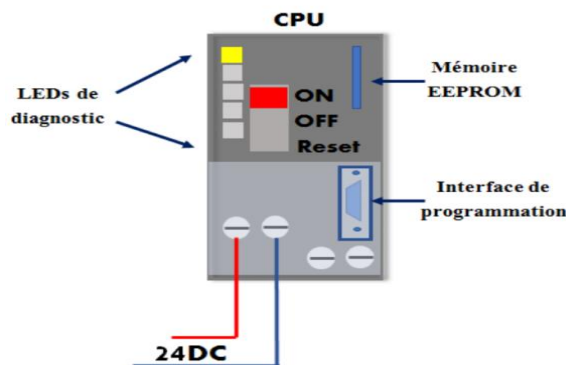


Figure II. 6 : Module CPU d'un API

Les coupleurs E/S (d'entrée/sortie) : Les coupleurs d'entrées/sorties sont des cartes électroniques qui assurent la liaison entre l'UC de l'automate programmable et la Partie Opérative. Les coupleurs d'entrées reçoivent les signaux des capteurs et des commandes de l'opérateur. Les coupleurs de sorties reçoivent les signaux de l'UC. Ils les amplifient et les rendent compatibles avec les pré-actionneurs commandés. Les APIs possèdent plusieurs E/S : E/S TOR, E/S numérique, et E/S analogique.

Module de communication : La CPU est généralement équipée d'un port de protocole de communication, qui peut être unique ou multiple. Dans certaines installations spécifiques, il peut être nécessaire d'ajouter des ports de communication supplémentaires. Si besoin, un module de communication peut être utilisé. Le placement de ce module doit suivre le module E/S ou intervenir après l'utilisation d'un FM. Les PLCs utilisent des ports intégrés tels que USB, Ethernet, RS-232, RS-485 ou RS-422 pour communiquer avec des périphériques

externes tels que des capteurs et des actionneurs, ainsi qu'avec des systèmes tels que le logiciel de programmation, les systèmes SCADA et les interfaces homme-machine (IHM). La communication s'effectue via différents protocoles de réseau industriel tels que Modbus RTU ou Ethernet/IP. Il est important de noter que de nombreux de ces protocoles sont spécifiques au fournisseur.

Une mémoire : Elle permet le stockage des instructions constituant le programme de fonctionnement ainsi que diverses informations. Il peut s'agir de mémoire vive RAM (modifiable à volonté, mais perdue en cas de coupure de tension) ou de mémoire morte EEPROM (seule la lecture est possible).

II.5 Fonctionnement d'un A.P.I

Quand l'API est en marche, trois étapes se succèdent :

- **PHASE 1 : Lecture des entrées (%I)**

Au cours de cette étape de quelques microsecondes, les entrées sont capturées et leurs états logiques sont conservés dans une zone spécifique de la mémoire de données. Le programme n'a pas été surveillé. Les éditions ne sont pas actualisées.

- **PHASE 2 : Traitement du programme (T)**

Pendant cette phase qui s'étend sur quelques millisecondes :

- Les consignes du programme sont suivies individuellement. La valeur stockée dans la mémoire de données est utilisée lorsque le programme doit lire l'état d'une entrée.
- Le programme enregistre les données des sorties et les stocke dans une section spécifique de la mémoire de données dédiée aux sorties.
- On ne vérifie pas les entrées. Les sorties ne sont pas actualisées.
- Pendant cette phase, seules la mémoire de données et la mémoire programme sont mises à contribution. Si une entrée change d'état sur le module d'entrées, l'API ne « voit » pas ce changement.

- **PHASE 3 : Ecriture des sorties (%Q)**

Pendant cette phase qui s'étend sur quelques millisecondes :

- Les états des sorties mémorisés précédemment dans la mémoire de données sont reportés sur le module de sorties.
- Les entrées ne sont pas scrutées.
- Le programme n'est pas exécuté.

Les trois étapes se répètent constamment de manière cyclique lorsque l'API est en marche.

L'API procède à des traitements internes avant chaque cycle pour vérifier ses circuits et les sollicitations extérieures. Un circuit électronique nommé « chien de garde » surveille le temps de cycle de quelques millisecondes. En cas de défaut, l'API signale le défaut et interrompt le traitement si le temps de cycle mesuré par le chien de garde dépasse le temps de cycle maxi configuré.

La figure suivante montre le fonctionnement cyclique d'un API :

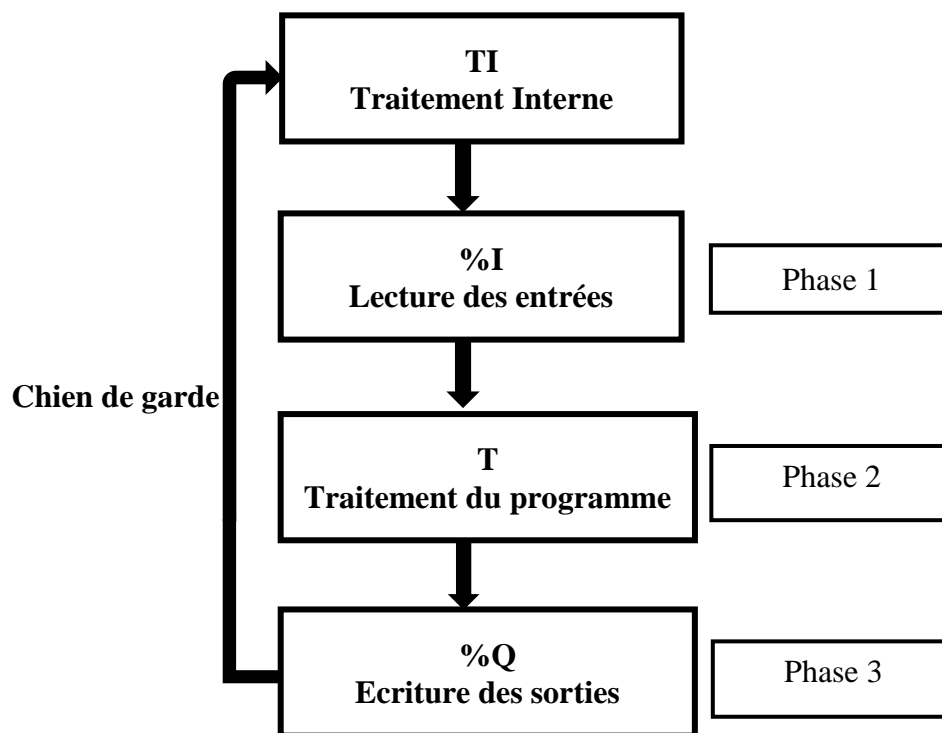


Figure II. 7 : Cycle d'un API

II.6 Les Fonctions principales d'un A.P.I

II.6.1 Fonctions logiques ou booléennes

On les appelle aussi fonctions binaires ou sur bits, leurs variables correspondant à 1 bit (Binary digit). Elles mettent en jeu des variables binaires, dites aussi discrètes, booléennes, ou tout-ou-rien (TOR), à deux états notés 0 et 1. Les fonctions ET(AND), OU(OR), NON(NOT) sont indispensables pour réaliser d'autres fonctions plus complexes. ET et OU sont généralisables à plus de deux entrées. On rencontre souvent NON ET (NAND), NON OU (NOR) et (EXCLUSIVE OR).

L'API possède également une fonction mémoire, quasi-indispensable pour le traitement des structures séquentielles comme représente la figure II.8, il s'agit d'une fonction logique dont le résultat est mémorisé : une entrée (SET) met à 1 la sortie qui reste à 1 même si l'entrée revient à 0; une autre entrée (RESET) remet de même la sortie à 0. Il faut prêter attention au comportement de la sortie lorsque les deux entrées sont à 1, ce qui peut survenir transitoirement ou suite à une erreur ou défaillance. Attention aussi au fait que selon la norme NF61131-3, ce n'est pas une fonction mais un bloc fonctionnel [14].

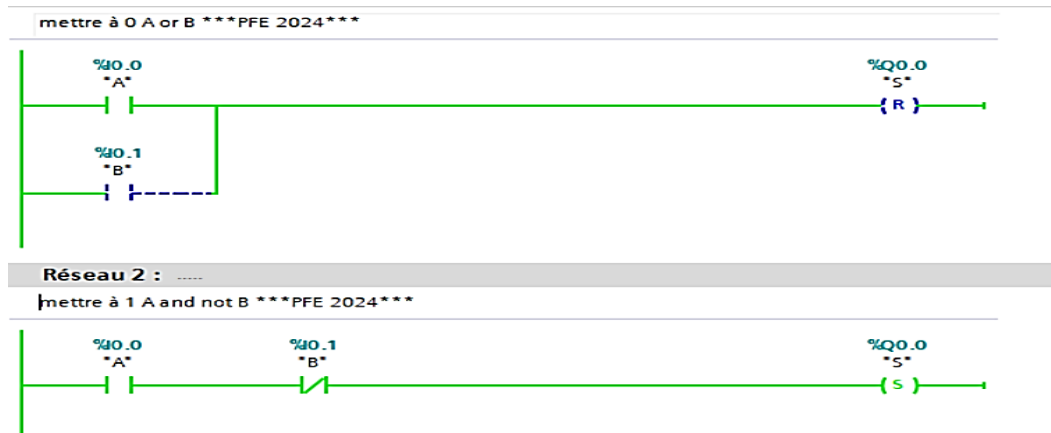


Figure II. 8 : Exemple d'opérations logiques sur TIA Portal

II.6.2 Fonctions arithmétiques et sur mots

Pour qu'un API effectue du traitement arithmétique (calcul de proportions par exemple), de la régulation, et même du comptage, qu'il envoie un message, il doit travailler sur un groupe de bits. Nombres entiers et mots standard s'expriment sur 16 bits.

Il existe donc des fonctions sur mot (lecture, écriture, comparaison, etc.), et des fonctions de calcul qui opèrent sur 16,32 ou 64bits, la figure II.9 montre un exemple de ces fonctions. La bibliothèque de fonctions arithmétiques et sur mots est une des caractéristiques de la capacité de traitement d'un API, ou de son orientation vers un type d'activité donné : régulation, télésurveillance de locaux, etc... [14]

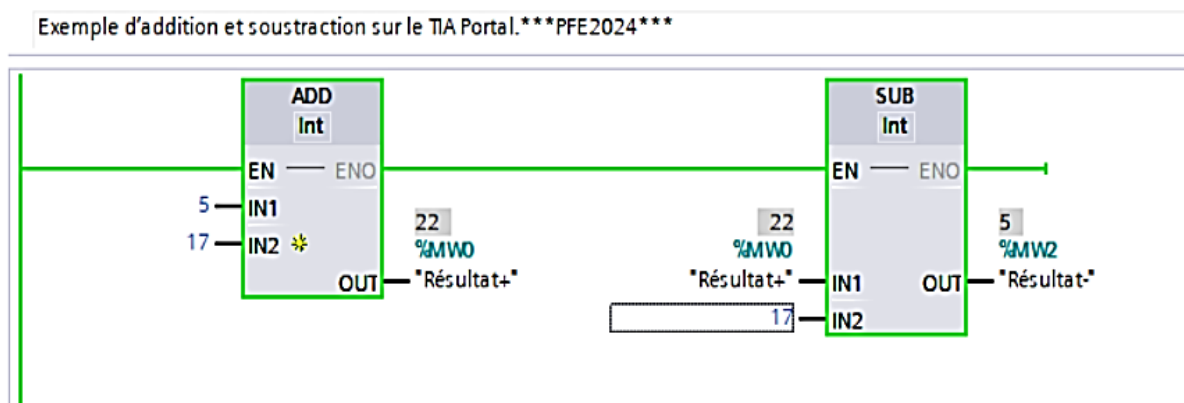


Figure II. 9 : Exemple d'addition et soustraction sur le TIA Portal

II.6.3 Fonctions de communication

Les fonctions de communication (émission, réception, etc.) selon un protocole donné font désormais partie du bagage minimum de l'API. Certains modèles possèdent même plusieurs jeux pour différents modes d'échange. [14]

II.6.4 Autres fonctions

Comptage et temporisation méritent d'être mentionnés explicitement car ils occupent une place importante dans l'univers de l'API. Les compteurs permettent de mémoriser l'occurrence d'événements : ce sont les changements d'état d'une entrée TOR ou d'une variable binaire interne, dénommés front montant pour le passage de 0 à 1, front descendant pour celui de 1 à 0. Outre ses applications directes (détermination d'un nombre de pièces, de passages, de défauts, etc.) le comptage s'emploie dans certaines fonctions métier, par exemple en positionnement. Toujours numériques, les temporisations s'expriment en fonction d'une base de temps choisie dans une gamme de valeurs : 1 s, 0,1 s, 0,01 s par exemple; elles réalisent un comptage d'événements internes particuliers, les impulsions de la base de temps. La présélection définit le nombre d'impulsions attendues. On peut ainsi retarder le début ou la fin d'une action. Un API dispose aussi d'un saut avant, c'est-à-dire de la possibilité de court-circuiter une partie de programme sur test d'une condition vraie ou fausse, mais pas nécessairement d'un saut arrière, qui risque d'allonger le temps d'exécution et va à l'encontre des exigences du temps réel.

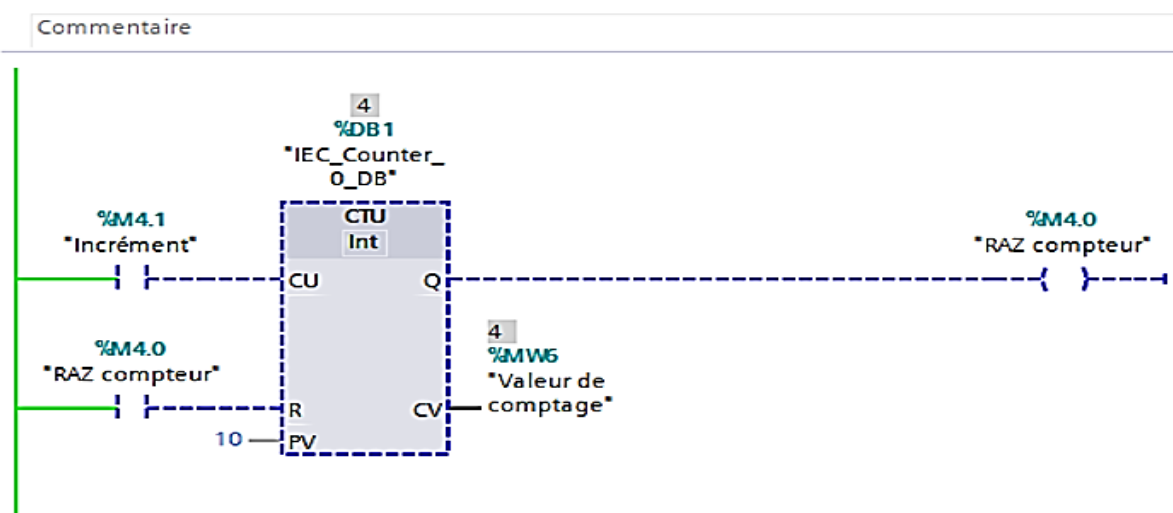


Figure II. 10 : Compteur incrémental

La figure II.10 représente un exemple d'un compteur à 10, il commence de la valeur 0, à chaque fois le bit %M4.0 s'active, le compteur s'incrémente et on peut voir la valeur de

comptage au-dessus du bloc, quand le comptage atteint la valeur 10 la sortie du compteur s'active, et le bit %M4.1 va réinitialiser le compteur. [14]

II.7 Langages de programmation des API

La norme CEI 61131 : CEI 61131 est une norme de la Commission électrotechnique internationale (CEI) destinée à réglementer plusieurs aspects de la programmation, équipements et communication des automates programmables industriels. Cette norme était nommée CEI 1131 auparavant. LaCEI611313 est la troisième partie de la famille de la norme CEI6113. Elle se compose des parties suivantes :

- Partie 1 : Vue d'ensemble Générale.
- Partie 2 : Matériel.
- Partie 3 : Langages de Programmation.
- Partie 4 : Directives d'Utilisation.
- Partie 5 : Communication.
- Partie 6 : Sécurité fonctionnelle.
- Partie 7 : Fuzzy Logic.
- Partie 8 : Directives d'application.

La CEI 61131-3, est une norme industrielle publiée la première fois en 1993. Cette norme définit cinq langages de programmation utilisés pour les automates programmables, deux langages textuels (Liste d'instructions et texte structuré) et deux langages graphiques (Diagramme à contacts et diagramme de bloc fonctionnel). Le dernier représente un ensemble d'éléments graphiques et textuels appelé "diagramme fonctionnel séquentiel" pour structurer l'organisation interne des programmes pour automate programmable et des blocs fonctionnels [15].

Avantages de la norme CEI 61131 :

- Formation des programmeurs d'applications.
- La création de programmes de plus en plus grands.
- La mise en œuvre de systèmes de programmation de plus en plus complexes.

Les systèmes de programmation API suivent progressivement la tendance du marché des logiciels de masse du monde PC. Ici aussi, la pression des coûts peut être réduite avant tout par la normalisation et la synergie. Étant donné que la norme rapproche des systèmes auparavant spécifiques à chaque fabricant, les fabricants et les clients ont tout à gagner de la norme CEI 61131-3.

Selon la norme CEI 61131-3 on a défini donc cinq langages de programmation pour les automates programmables qui sont :

II.7.1 LD (Ladder Diagram)

Ou schémas à relais, il se base sur une approche visuelle évoquant des schémas électriques. Ce langage graphique est essentiellement dédié à la programmation d'équations booléennes.

II.7.2 IL (Instruction List)

Ou listes des instructions, ce langage textuel de bas niveau est un langage à une instruction par ligne, il est très proche du langage informatique dit assembleur.

II.7.3 FBD (Function Block Diagram)

Ce langage permet de programmer graphiquement à l'aide de blocs, représentant des variables, des opérateurs ou des fonctions. Il permet la construction d'équations complexes.

II.7.4 ST (Structured Text)

Le texte structuré est un langage textuel de haut niveau qui est utilisé pour décrire des procédures complexes.

II.7.5 SFC (Sequential Function Charts)

Appelé en français les graphes de fonction séquentielle, est un langage issu du langage GRAFCET il permet la programmation aisée de tous les procédés séquentiels.

II.8 Les blocs dans TIA Portal

Les blocs sont les éléments fondamentaux de la programmation dans TIA Portal. Ils encapsulent des fonctionnalités et des données spécifiques, permettant de structurer et de réutiliser du code efficacement.

Les blocs principaux que vous rencontrerez dans TIA Portal sont les suivants :

- **Les blocs de données-DB**

Les données utilisateur sont stockées dans des zones données du programme appelées blocs de données (DB). Il existe deux catégories de blocs de données :

- **Une base de données globale** peut être créée sans tenir compte des autres blocs. Les données contenues dans un bloc de données global peuvent être lues ou écrites par tous les blocs (FB, FC et OB).

- **Le DB d'instance** : est associé à un certain bloc fonctionnel et contient les données locales de ce FB associé. Il ne peut être créé que si le FB associé est déjà présent dans le programme.

- **Les fonctions – FC**

Ces fonctions sont utilisées pour la programmation de fonction utilisées plusieurs fois, sachant que les données des variables temporaires sont perdus après l'exécution de la fonction, donc on peut dire que ce sont des blocs sans mémoire, mais si on veut mémoriser ces données, il faut utiliser des opérandes globaux.

- **Les blocs fonctionnels – FB**

Il s'agit de blocs de code qui conservent leurs paramètres d'entrée, de sortie et d'entrée/sortie de manière durable dans des blocs de données d'instance, permettant ainsi d'y accéder même après le traitement des blocs (contrairement à FC).

- **Les blocs d'organisation – OB**

Les blocs OB permettent de structurer notre programme en séparant la logique en différentes parties gérables. Cela facilite la compréhension, la préservation et la réutilisation du code.

Les OB sont appelés par le système d'exploitation en liaison avec les évènements suivants :

- Exécution cyclique du programme et traitement des erreurs.
- Exécution du programme déclenchée par des alarmes (processus, diagnostic...).

II.9 Choix d'un API

Lors du choix d'un PLC, il est nécessaire de prendre en compte les critères suivants :

- **Besoins spécifiques :**

- **Fonctionnalités** : quelles fonctionnalités l'API doit-elle fournir pour répondre à vos besoins spécifiques.
- **Cas d'utilisation** : Comment allez-vous utiliser l'API ? Intégrez-les aux application existantes, créez de nouveaux services, etc.
- **Données**: quelles données l'API prétend-elle fournir ?
- **Volume d'utilisation**: Quel est le volume d'utilisation attendu pour l'API ?

- **Aspects techniques:**

- **Langage de programmation**: L'API est-elle compatible avec votre langage de programmation préféré ?

- **Format de données:** Quel format de données l'API utilise-t-elle (JSON, XML, etc.) ?
 - **Documentation:** La documentation de l'API est-elle claire, complète et facile à comprendre ?
 - **Niveau de sécurité:** Quels protocoles de sécurité sont en place pour protéger l'API et les données ?
 - **Temps de cycle :** pour certains types d'applications, un contrôleur avec un temps de cycle rapide est essentiel pour garantir un fonctionnement correct du système et une meilleure exécution des tâches
 - **Taille de la mémoire et nombre d'entrées/sorties :** un CPU de taille suffisante doit être sélectionné pour le projet à exécuter et la même chose pour le nombre d'entrées et de sorties de l'automate.
 - **Bus industriels et interfaces de communication disponibles nativement sur l'automate :** Si vous utilisez un réseau de terrain spécifique, il est recommandé de choisir un automate supportant nativement ce monde de communication. Cela vous évitera d'acheter des modules supplémentaires.
- **Facteurs externes:**
- **Disponibilité en termes de service d'après-vente et de pièces de rechange :** certaines marques d'automates ont plus d'autorité pour fournir un meilleur service après-vente et un bon accompagnement de client.
 - **Coût:** L'API est-elle gratuite ou payante ? Quels sont les différents plans de tarification disponibles ?
 - **Fournisseur:** Quelle est la réputation du fournisseur de l'API ? Offre-t-il un support client adéquat ?
 - **Communauté:** Y a-t-il une communauté active d'utilisateurs de l'API ?
 - **Évolution:** L'API est-elle régulièrement mise à jour et améliorée ?

II.10 Les Avantages des APIs

- **Gain de temps et d'efficacité :** L'un des principaux avantages de l'automatisation est le gain de temps qu'elle permet. Les tâches répétitives sont effectuées plus rapidement et plus efficacement grâce à l'automation. Les erreurs sont également réduites, car les robots sont plus précis que les êtres humains pour exécuter certaines tâches. Les employés peuvent ainsi se concentrer sur des tâches plus complexes et à plus forte valeur ajoutée, ce qui permet d'augmenter la productivité et la rentabilité de l'entreprise.

- **Puissance et performance** : Les automates sont un outil très performant, un seul modèle peut être utilisé pour différentes opérations selon les besoins.
- **Robustesse** : Les éléments qui composent les automates sont particulièrement robustes (absence de mécanique tournante pour le refroidissement et le stockage des données, matériaux renforcés) leur permettant de fonctionner dans des environnements particulièrement hostiles (poussière environnante, vibrations des supports, variations de température...).
- **Facilité de modification du programme par rapport à la logique câblée** : Les automates sont fournis avec des langages de programmation simples comme la programmation de type Ladder ou Booléenne [16].

II.11 Les inconvénients des APIs

Cependant il existe des inconvénients de l'automate qu'on peut négliger comme :

- **La Cherté de l'appareil et son système** : Ils sont plus chers que des solutions informatiques classiques à base de microcontrôleurs par exemple.
- **Le plantage** : Blocage du programme ou du système qui paralyse par conséquent le fonctionnement de l'API. [17]

II.12 Les automates Schneider

Schneider Electric se distingue à l'échelle mondiale en matière d'automatisation industrielle. La société offre une variété d'automates programmables (API) spécialement conçus pour satisfaire tous les besoins d'applications, qu'elles soient simples ou complexes. Les API Schneider Electric sont employés dans divers secteurs tels que la production, l'agroalimentaire, l'industrie automobile et l'énergie.

II.12.1 L'automates Schneider TSX Micro

L'automate programmable compact (API) Modicon TSX Micro de Schneider Electric est spécialement conçu pour les applications de petite et moyenne taille. Ils proposent un rapport qualité-prix excellent, une utilisation facile et une variété de fonctionnalités pour satisfaire les besoins d'une variété d'applications industrielles [18].

La gamme d'automates programmables (PLC) TSX Micro comprend plusieurs types d'automates afin de répondre au mieux à vos besoins. Les automates TSX 37-05, TSX 37-08 et TSX 37-10 sont des modèles compacts capables d'intégrer

un ou deux modules d'entrées/sorties discrets dans leur configuration, selon le type de module. Il existe également des automates modulaires, les TSX 37-21 et TSX 37-22.

L'automate **TSX 37-05** intègre un module d'entrées/sorties (E/S) de 28 points (16 entrées + 12 sorties) dans sa configuration. Ce module se situe dans le premier emplacement et dispose de deux demi-emplacements disponibles qui permettent de recevoir soit un module de format standard, soit deux modules de demi-format. Sa capacité d'entrées/sorties maximale est de 92 E/S discrets, en installant dans l'emplacement disponible un module d'E/S discrets de 64 points connecté par un connecteur HE10 [19].

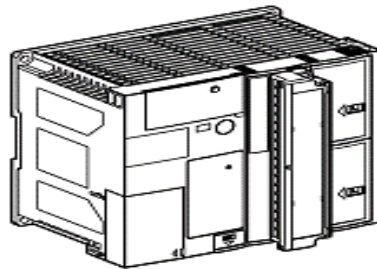


Figure II. 11: L'automate TSX 37-05

L'automate **TSX 37-08** intègre deux modules d'entrées/sorties (E/S) de 28 points chacun (16 entrées + 12 sorties) dans sa configuration. Ces modules se situent dans les deux premiers emplacements et disposent de deux demi-emplacements disponibles qui permettent de recevoir soit un module de format standard, soit deux modules de demi-format. Sa capacité d'entrées/sorties maximale est de 120 E/S discrets, en installant dans l'emplacement disponible un module d'E/S discrets de 64 points connecté par un connecteur HE10.

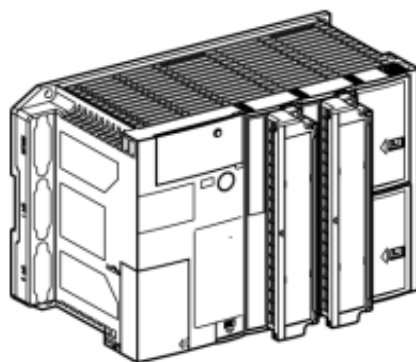


Figure II. 12 : L'automate TSX 37-08

Les automates **TSX 37-10** proposent cinq configurations de base possibles. Ces configurations diffèrent par leur tension d'alimentation et par le type de module d'entrées/sorties discrètes installé dans le premier emplacement. Ces automates peuvent accueillir une mini-extension d'entrées/sorties, ce qui permet d'étendre le nombre d'entrées/sorties locales jusqu'à 192. De plus, les automates TSX 37-10 sont équipés d'une horloge en temps réel.

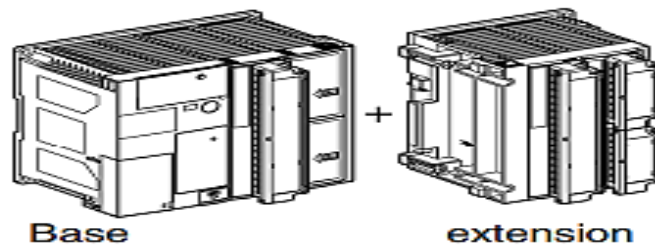


Figure II. 13: L'automate TSX 37-10

L'automate **TSX 37-21** est disponible en 2 configurations qui se différencient par le type d'alimentation électrique. Ces automates n'intègrent pas de modules d'entrées/sorties discrets dans leur configuration de base. Leur capacité d'entrées/sorties maximale est de 256 points lorsqu'une mini-extension d'entrées/sorties est ajoutée. Ils sont équipés d'une horloge en temps réel, permettent l'extension de la mémoire de l'application et peuvent recevoir un module de communication.

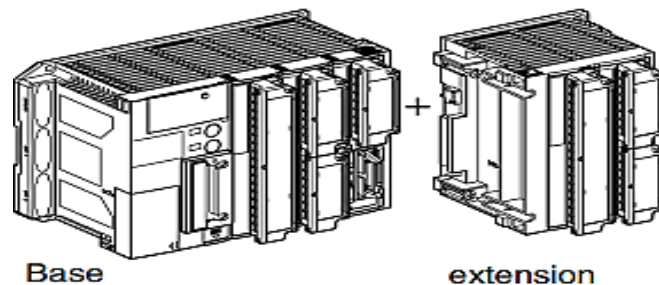


Figure II. 14 : L'automate TSX 37-21

L'automate modulaire **TSX 37-22** est identique à l'automate TSX 37-21 à tous égards, excepté qu'il intègre des fonctions de comptage rapide et d'entrées/sorties analogiques supplémentaires [19].

II.12.2 PL7 Pro

Le logiciel PL7 Pro est un outil de programmation pour les automates Modicon TSX Micro et Modicon Premium de Schneider Electric. Il permet de structurer une application pour automate Premium en modules fonctionnels. Un module fonctionnel est un regroupement d'éléments de programme destinés à réaliser une fonction spécifique. [20]

Autrement dit, c'est un outil qui permet de créer et de modifier les programmes qui contrôlent le fonctionnement de ces automates. On l'utilise pour définir la logique de commande des machines et des processus industriels.

Voici quelques informations clés sur PL7 Pro :

- **Programmation API avancée** : PL7 Pro est utilisé pour la conception et la mise en œuvre de l'automatisation des unités flexibles de robot.

- **Compatibilité** : Le logiciel est compatible avec les systèmes d'exploitation Windows 7 et Windows XP.
- **Version** : La version la plus récente du logiciel est la version 4.5.
- **Langues** : PL7 Pro est disponible en anglais et en français.
- **Arrêt de commercialisation** : Veuillez noter que le produit a été arrêté et n'est plus commercialisé. Cependant, Schneider Electric continue de fournir un support pour ce produit.

Pour plus de détails et voir une idée comment créer un nouveau projet sur PL7 Pro voir l'annexe A.

II.13 Présentation de la gamme SIMATIC de SIEMENS

SIMATIC constitue le noyau de Totally Integrated Automation, un concept industriel développé par Siemens qui englobe l'ensemble de la chaîne de production. Son architecture système se distingue par des caractéristiques partagées entre le matériel et le logiciel, telles qu'une gestion de données cohérente, le respect des normes internationales et des interfaces uniformes.

La gamme SIMATIC englobe un éventail complet de produits éprouvés pour les applications les plus diverses des industries manufacturières et de processus. [21]

II.13.1 Présentation de l'automate programmable S7-1200

Afin d'automatiser notre machine, nous avons opté pour l'utilisation de l'automate S7-1200, un contrôleur programmable conçu par Siemens. Il fait partie de la gamme de produits SIMATIC S7, qui est couramment employée dans le domaine de l'automatisation industrielle. Il nous convient pour cette application en raison de son compteur rapide et de son nombre d'E/S adéquat.

SIMATIC S7-1200 est un mini-contrôleur modulaire destiné aux performances réduites. Les trois classes de performances des CPU du système SIMATIC S7-1200 sont les CPU 1211C, CPU 1212C et CPU 1214C, chacune d'elles pouvant être élargie en fonction des exigences de la machine. On peut intégrer une platine d'extension (Signal Board) sur chaque CPU afin d'ajouter des E/S TOR ou analogiques supplémentaires sans avoir à modifier la taille de l'automate. Il est possible d'inclure des modules de signaux à droite de la CPU afin d'augmenter la capacité d'E/S TOR ou analogiques. La 1212C peut accueillir deux modules d'E/S supplémentaires, tandis que la 1214C en accueille huit. De plus, il est possible de monter jusqu'à trois modules de communication à gauche de l'automate sur toutes les CPU SIMATIC S7-1200 afin d'obtenir une communication optimale.

La communication optimale avec le système d'ingénierie intégré SIMATIC STEP 7 Basic et les périphériques PROFINET IO est assurée par une interface Industriel Ethernet/PROFINET intégrée au SIMATIC S7-1200. Elle offre la possibilité de programmer et de communiquer avec les pupitres SIMATIC HMI Basic Panels pour la visualisation, avec des automates supplémentaires pour la communication entre CPU et CPU, ainsi qu'avec des appareils d'autres fabricants pour une intégration élargie.

La connexion RJ45 insensible aux perturbations de l'interface de communication SIMATIC S7-1200 est équipée d'une fonction d'auto croisement, supportant de multiples connexions Ethernet et offrant une vitesse de transmission allant jusqu'à 10/100 Mbit/s.

En résumé Le S7 1200 est un contrôleur puissant qui combine un microprocesseur, une alimentation intégrée, des circuits d'entrée et de sortie, un PROFINET intégré, des E/S rapides de commande de mouvement et des entrées analogiques dans un boîtier compact.

Caractéristiques clés de l'automate S7-1200 :

- **Programmation** : Il est programmé à l'aide du logiciel STEP 7 Basic, qui offre une interface conviviale pour le développement et la configuration des applications. Le langage de programmation utilisé est le langage de programmation graphique par blocs fonctionnels (FBD), ainsi que le langage d'instructions structurées (LADDER) et d'autres langages.
- **Puissance de traitement** : L'automate S7-1200 dispose d'un processeur intégré puissant qui permet d'exécuter rapidement les programmes et de traiter les entrées/sorties.
- **Entrées/sorties (E/S) intégrées** : Il est équipé d'une gamme d'interfaces d'E/S numériques et analogiques intégrées qui permettent de se connecter à différents capteurs, actionneurs et autres équipements sur le terrain.
- **Communication** : L'automate S7-1200 prend en charge plusieurs protocoles de communication, tels que PROFIBUS, PROFINET, Modbus et Ethernet TCP/IP. Cela permet d'établir des connexions avec d'autres automates, systèmes de supervision, périphériques et réseaux industriels.
- **Flexibilité modulaire** : L'automate S7-1200 peut être étendu avec des modules d'extension, tels que des modules d'E/S supplémentaires, des modules de communication ou des modules de fonction spécifique, afin de répondre aux besoins spécifiques de l'application.
- **Fonctions de sécurité** : Il intègre des fonctionnalités de sécurité intégrées, telles que la surveillance des entrées de sécurité, les arrêts d'urgence et les fonctions de sécurité

programmables, permettant de répondre aux exigences de sécurité des machines et des processus industriels.

- **Maintenance et diagnostic** : L'automate S7-1200 dispose de fonctionnalités avancées de maintenance et de diagnostic, telles que la surveillance en temps réel, les journaux d'événements, la gestion des alarmes et la possibilité de mise à jour du firmware.

L'automate S7-1200 représenté dans la figure II.15 est utilisé dans une variété d'applications industrielles, telles que le contrôle des processus, l'automatisation des machines, la surveillance et la gestion des systèmes. Sa taille compacte, sa flexibilité et ses performances en font un choix populaire pour les petites et moyennes applications d'automatisation [22].



Figure II. 15 : Automate programmable S7-1200 1214C

II.13.2 Caractéristiques techniques de l'APIS7-12001214C

Les caractéristiques techniques sont :

- 14 entrées et 10 sorties TOR
- 2 entrées analogiques
- Une mémoire de travail 100 kilo-octets
- Une mémoire de chargement de 4 Méga-octets
- Processeur intégré avec une fréquence de fonctionnement de 100 MHz
- Temps de traitement par la CPU 0.08 μ s pour opérations sur bits et 2.3 μ s pour opérations à virgule flottante
- Interface de communication PROFINET de type RJ45(Ethernet), d'une vitesse de transmission de 100 Mbit/s
- 6 compteurs rapides (HSC) incrémentation/décrémentation avec une fréquence de comptage maximale de 100 kHz

II.13.3 Caractéristiques de la CPU 1214C DC/DC/RELAY (6ES7214-1HG40XB0)

La CPU utilisée dans notre projet est de type S7-1200 DC/DC/RELAY de référence 6ES72141HG40-0XB0, elle se caractérise par sa compacité avec des entrées sorties embarquées et on en trouve: 14 entrées TOR 24VCC;10 sorties TOR RELAIS 2A; 2 entrées analogiques 0 - 10V CC, son alimentation est en CC: 20,4 - 28,8 V CC, sa mémoire programme/données est de 100 KO, mémoire de chargement 4 MO avec possibilité d'insertion d'une carte de mémoire SIMATIC [23].

II.13.4 Logiciel de programmation TIA Portal

TIA Portal est un environnement d'automatisation intégré qui simplifie la configuration des processus de production les plus complexes à partir d'un seul écran d'ordinateur. Il optimise la planification et la production.

Avec une interface intuitive et une navigation facile, l'apprentissage des fonctions de programmation est rapide. Le TIA Portal est divisé en une "vue portail" pour guider l'utilisateur à travers les différentes étapes de l'ingénierie, et une "vue projet" pour un accès rapide aux outils pertinents. Il aide les utilisateurs, qu'ils soient novices ou expérimentés, à travailler de manière productive.

La conception des éditeurs logiciels du TIA Portal repose sur une présentation et une navigation communes. Que ce soit pour la configuration du matériel, la programmation logique, le réglage d'un variateur ou la conception d'une image IHM, chaque environnement utilise le même design pour les éditeurs. Les fonctions, caractéristiques et bibliothèques sont affichées de manière intuitive en fonction de l'activité souhaitée, y compris pour tous les composants de sécurité.

La configuration de tout le matériel et de son réseau se fait dans une vue graphique complète des appareils et du réseau. En effet, le réseau du contrôleur, des IHM, du PC et des entraînements est réalisé par une simple configuration graphique des connexions [24].

II.13.4.1 Description du logiciel TIA Portal

La plateforme de développement TIA Portal « Totally Intergrated Automation Portal » est le nouvel environnement de travail Siemens qui permet de faire un gain important en temps lors du développement de systèmes d'automatisation. C'est une plateforme tout en un comportant le logiciel Step 7 pour la programmation d'automates et WinCC Flexible pour les interfaces homme-machine.

II.13.4.1.1 SIMATIC STEP 7

SIMATIC STEP 7 est reconnu mondialement comme le logiciel d'ingénierie le plus populaire et le plus largement utilisé pour l'automatisation industrielle. Il fonctionne dans le cadre du logiciel Totally Integrated Automation Portal (TIA Portal). SIMATIC STEP 7 offre à l'utilisateur la possibilité de configurer, programmer, tester et diagnostiquer tous les automates SIMATIC.

II.13.4.1.2 SIMATIC WinCC

WinCC (TIA portal) est un logiciel d'ingénierie pour la configuration de pupitres SIMATIC, de PC industriel SIMATIC et de PC standard par le logiciel de visualisation. Le SIMATIC WinCC dans le TIA portal fait partie d'un nouveau concept d'ingénierie intégré qui offre un environnement d'ingénierie homogène pour la programmation et la configuration de solution de commande, de visualisation d'entraînement, c'est le logiciel pour toutes les applications IHM allant de solutions de commande simples avec basic panels aux applications SCADA pour système multipostes basé sur PC [36].

II.13.4.2 Vues du TIA Portal

Lorsqu'on lance TIA Portal, l'espace de travail est divisé en deux vues :

- **Vue du portail** : Fournit un aperçu de toutes les étapes de planification de projet et un accès orienté tâche, et divers portails ("Test Run", "Devices and Networks", "PLC Programming", etc....) montrent toutes les étapes nécessaires à l'exécution de l'automatisation d'une façon claire et organisé. La figure suivante montre la vue du portail



Figure II. 16 : Vue de portail

- **La vue du projet :** elle comporte une arborescence avec les différents éléments du projet. Les éditeurs requis s'ouvrent en fonction des tâches à réaliser. Données, paramètres et éditeurs peuvent être visualisés dans une seule et même vue. La figure suivante montre la vue du projet TIA :

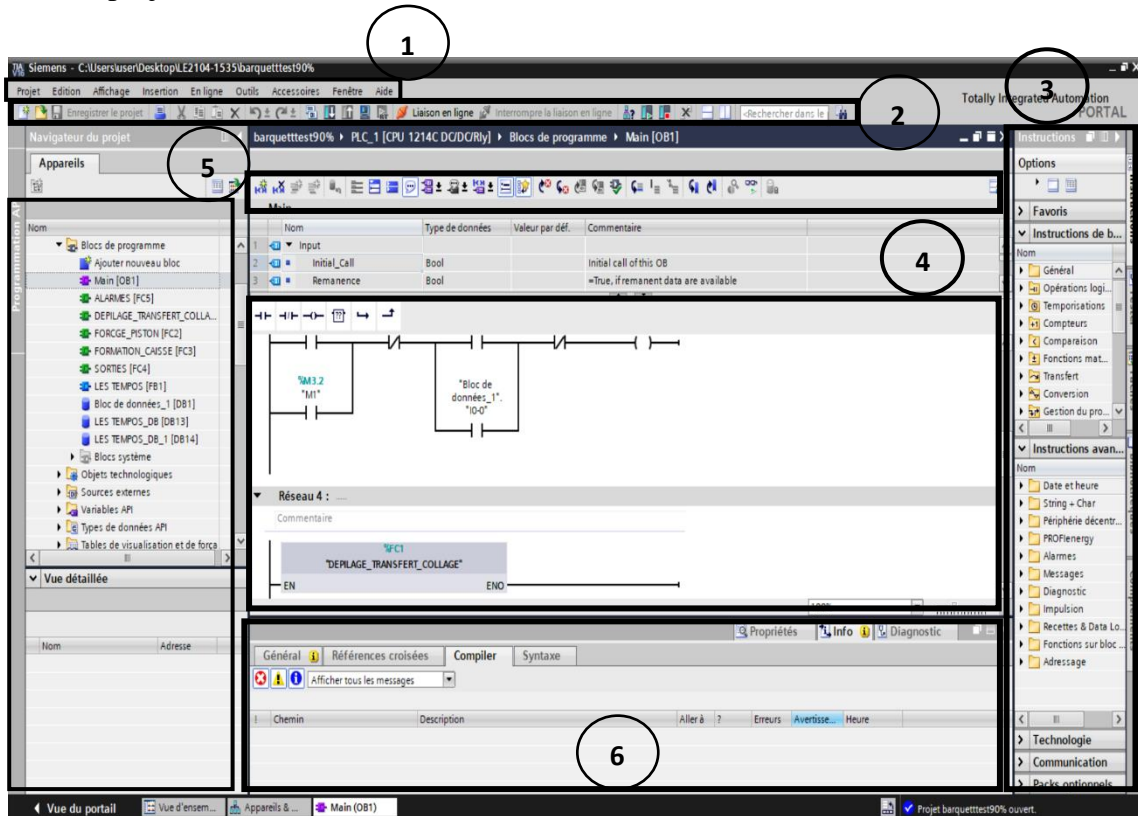


Figure II. 17 : Vue du projet

- 1 : Barre de menu ; 2 : Barre d'outil ; 3 : Catalogue/Bibliothèque ; 4 : Editeurs ;
- 5 : Navigateur du projet ; 6 : Fenêtre d'inspection.
- **La fenêtre de travail** permet de visualiser les objets sélectionnés dans le projet pour être traités. Il peut s'agir des composants matériels, des blocs de programme, des tables des variables, des HMI.
- **La fenêtre d'inspection** permet de visualiser des informations complémentaires sur un objet sélectionné ou sur les actions en cours d'exécution (propriété du matériel sélectionné, messages d'erreurs lors de la compilation des blocs de programme).
- Les onglets de sélection de tâches sont un contenu qui varie en fonction de l'objet sélectionné (configuration matérielle, bibliothèques des composants, bloc de programme, instructions de programmation).
- **Les avantages du logiciel TIA portal**
 - Programmation intuitive et rapide : grâce aux éditeurs de programmation développés SCL, CONT, LOG, LIST et GRAPH.

- Efficacité accrue grâce aux innovations linguistiques de STEP 7 : programmation symbolique uniforme, Calculate Box, ajout de blocs durant le fonctionnement, et bien plus encore.
- Performances améliorées avec des fonctionnalités intégrées : simulation avec PLCSIM, télémaintenance avec TeleService, diagnostic système cohérent.
- Technologie flexible : Fonctionnalité motion control évolutive et efficace pour les automates S7-1500 et S7-1200.
- Sécurité renforcée avec « Security Integrated » : Protection du savoir-faire, protection contre la copie, protection d'accès et protection contre la falsification.

II.13.4.3 Simulateur PLCSIM

S7-PLCSIM permet d'exécuter et de tester le programme, et de remédier à d'éventuelles erreurs avant son implantation dans l'automate, ainsi il dispose d'une interface simple permettant de visualiser et de modifier ou forcer les différents paramètres utilisés par le programme (d'activer ou de désactiver des entrées). La figure ci-dessous montre la vue du simulateur PLCSIM du TIA V16

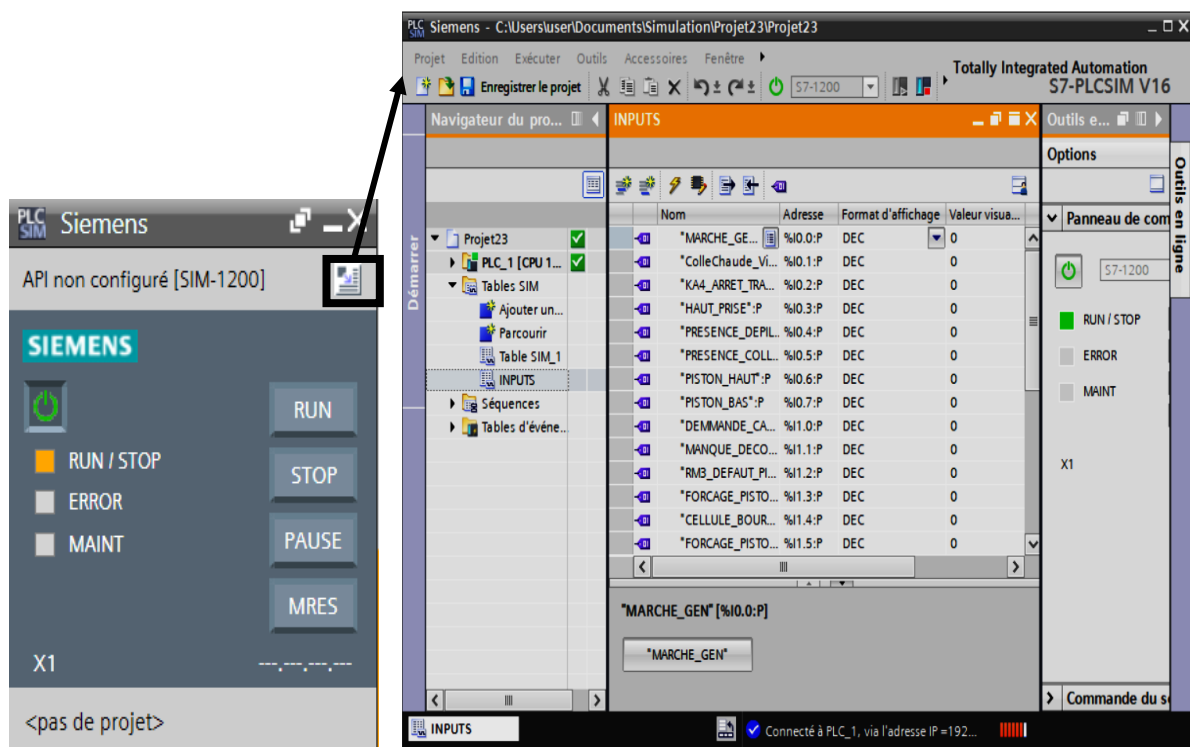


Figure II. 18 : S7 PLCSIM

II.13.5 Définition de la supervision

Technique industrielle de suivi et de pilotage informatique de procédés de fabrication automatisés. La supervision concerne l'acquisition de données (mesures, alarmes, retour d'état de fonctionnement) et des paramètres de commande des processus généralement confiés à des automates programmables [26].

Les fonctions de la supervision sont multiples, voici quelques exemples :

- Assure la communication entre les équipements d'automatisation et les outils informatiques pour la planification et la gestion de la production.
- Assiste l'opérateur dans les opérations de diagnostic et de maintenance.
- Coordonne le fonctionnement d'un ensemble de machines enchaînées constituant une ligne de production, en assurant l'exécution d'ordres communs tels que la marche, l'arrêt, et en gérant des tâches telles que la synchronisation.
- Fournit en temps réel une visualisation de l'évolution des paramètres du processus, ce qui permet à l'opérateur de prendre rapidement des décisions adaptées à ses objectifs.
- Répond à des besoins qui nécessitent généralement une puissance de traitement importante.

II.13.5.1 Pourquoi Superviser ?

La supervision d'un système automatisé présente plusieurs avantages significatifs :

- Contrôler la disponibilité des services/fonctions.
- Contrôler l'utilisation des ressources.
- Amélioration de la maintenance : La supervision facilite la maintenance préventive en fournissant des informations sur l'état des équipements et des composants du système.
- Diagnostic des pannes.
- Prévenir les pannes/défauts/débordements (pannes latentes).
- Prévoir les évolutions.
- Surveillance continue à distance : La supervision permet une surveillance continue de l'état et des performances du système automatisé, sans devoir être proche au système.

II.13.5.2 Interface Homme Machine

Les Interfaces Homme-Machine (IHM) ou en anglais (HMI) sont des dispositifs qui permettent aux utilisateurs de communiquer avec des machines, des logiciels ou des appareils électroniques. L'une des fonctions principales des IHM est de présenter des informations de manière conviviale et compréhensible. Les IHM sont conçues pour présenter des informations de manière à ce que les utilisateurs puissent interagir efficacement avec les systèmes informatiques ou les dispositifs électroniques. Une conception appropriée de l'IHM est essentielle pour garantir que les informations sont présentées de manière claire, intuitive et conviviale pour les utilisateurs [27].

II.13.5.3 Types d'IHM de supervision

- **IHM de supervision basique** : Ce type d'IHM de supervision est assez simpliste dans sa conception, au niveau de ses graphismes et de ses fonctionnalités.
- **IHM de supervision par menus** : Dans ce type d'IHM de supervision, les menus de navigation ont une importance prépondérante. L'opérateur peut effectuer rapidement un maximum d'actions ou accéder, en un minimum de temps, à différents écrans successivement [27].

II.13.5.4 Choix d'IHM

Le choix de vos IHM sera principalement basé sur un besoin d'automatisation de votre projet. Vous voulez avant tout garantir une meilleure productivité. Dans un second temps, il vous sera demandé d'examiner l'atmosphère dans laquelle vous travaillerez tel que : l'ensemble de vos logiciels, la connectivité et la sécurité de vos données et l'extensibilité et la flexibilité de vos processus.

II.13.5.5 Création d'un nouveau Panel dans le projet

Pour créer un nouveau Panel dans le projet, ouvrir la fenêtre de sélection en cliquant sur (1) Ajouter un appareil, après en cliquant sur (2) HMI et (3) en choisissant notre Panel TP1200 Comfort, pour confirmer le choix en cliquant sur le bouton (4) Ajouter.

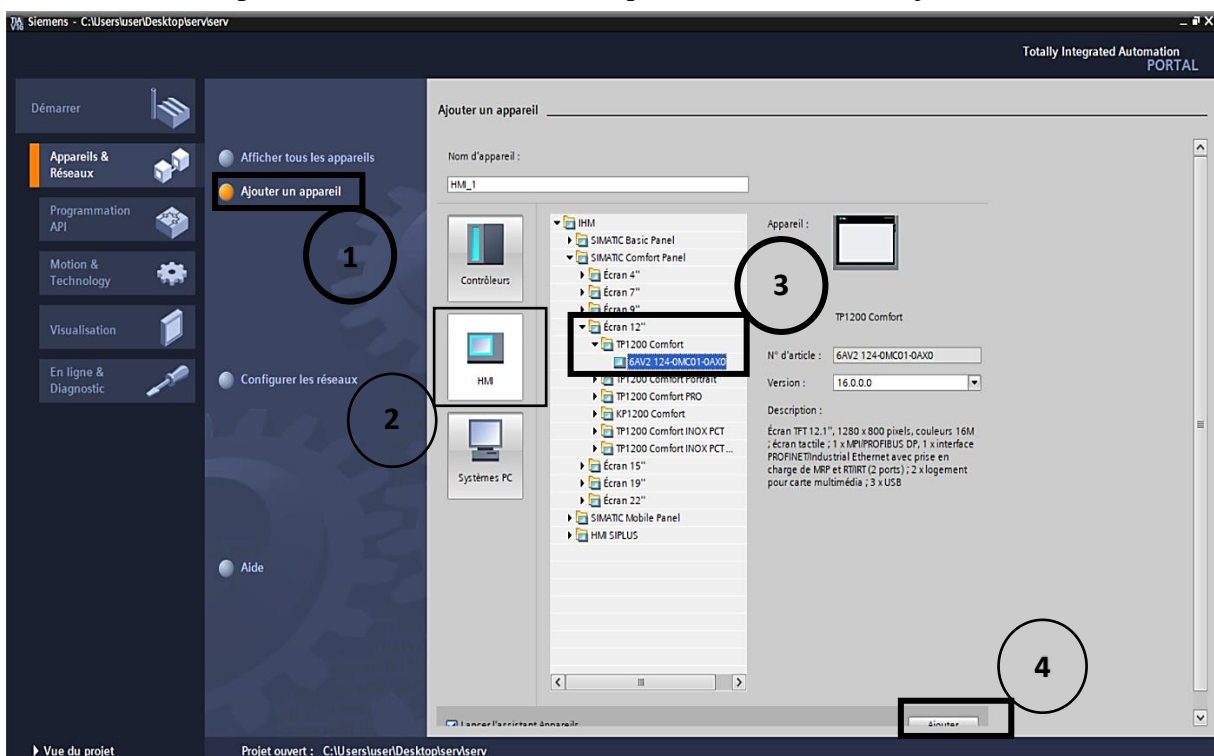


Figure II. 19 : Etapes pour ajouter un nouveau panel

En clique sur Parcourir et en sélectionne notre PLC

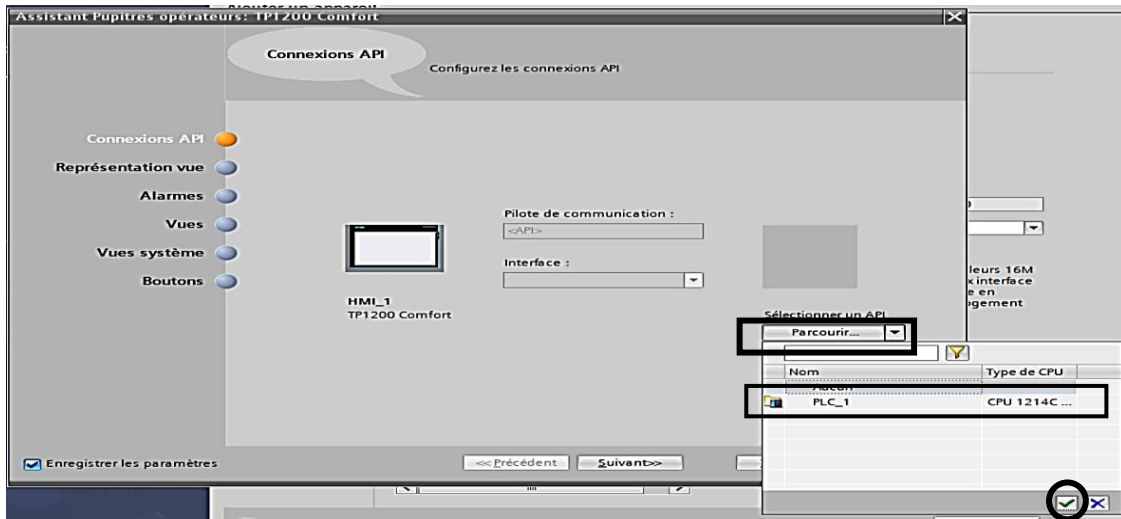


Figure II. 20 : Parcourir et sélectionne d'API

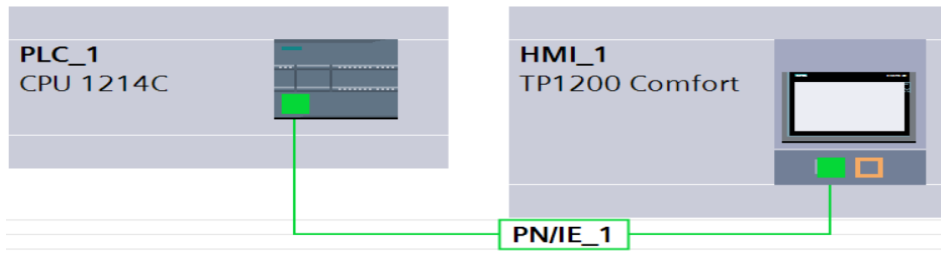


Figure II. 21 : Connexion entre API et IHM

En clique sur Suivant pour tous ce qui suit, après Terminer et notre vue IHM s'affiche.

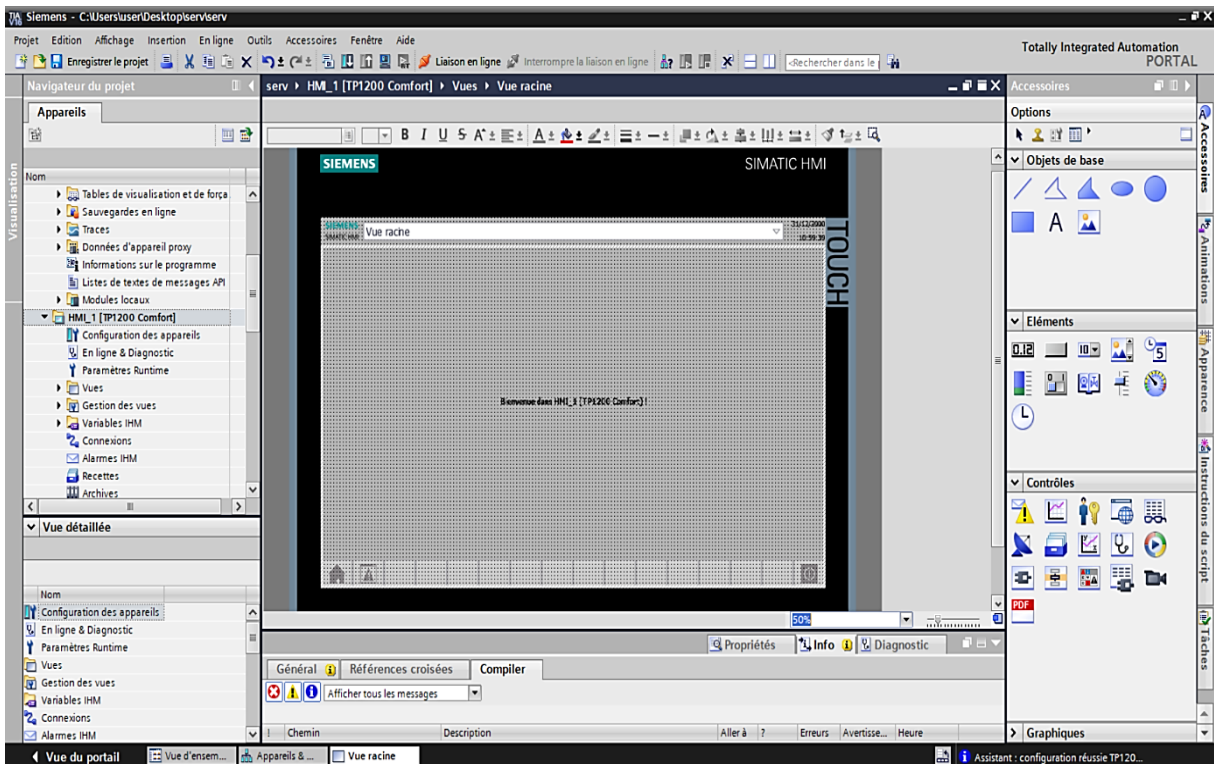


Figure II. 22: Vue IHM

II.14 Conclusion

Ce chapitre présente une description approfondie de l'automate programmable, de ses composants, de ses fonctions essentielles, de son rôle dans le système automatisé, et de l'avantage que cet appareil offre à l'industrie. Par la suite on a donné une description de l'automate programmable Schneider Modicon TSX micro et son logiciel de programmation PL7 Pro, l'API S7-1200 1214C utilisée dans ce projet a été exposée, ainsi que ses caractéristiques techniques. En fin nous avons arrivé à voir comment configurer une vue IHM.

Nous allons entamer l'étude de cahier de charges la programmation de l'automate à partir du logiciel TIA Portal V16 en langage Ladder (langage à contacts) dans le chapitre suivant.

**Chapitre III : Cahier de charges de la machine et
programmation de l'API**

III.1 Introduction

Avant d'entamer la partie programmation on va rédiger le cahier de charges de la barquetteuse et le traduire en GRAFCET, par la suite afin d'automatiser la machine, on a opté pour l'utilisation de l'automate programmable S7-1200(on va justifier la migration de l'automate Schneider TSX micro vers l'automate Siemens S7 1200). Le logiciel TIA Portal V16 de programmation offre un environnement convivial pour concevoir, modifier et superviser la logique requise pour contrôler une application. Le projet permet d'associer le programme à toutes les informations nécessaires pour communiquer avec l'API et charger le programme dans ce dernier.

Tout d'abord, on décrira la procédure à suivre pour créer et configurer le matériel d'un projet d'automatisation, ainsi que la structure d'un projet dans le logiciel TIA Portal, ensuite on expliquera la logique derrière le programme.

III.2 Cahier de charges

III.2.1 Dépilage_Transfert_Collage

Avant que l'opérateur clique sur le bouton « marche générale » qui déclenche le début de cycle, il faut qu'il vérifie les conditions initiales (la découpe est présente, le convoyeur à taquets est à l'état initial, la station de colle est prête, le piston de formation de caisse est en position haute et les deux vérins de pressage sont à l'état initial). Si sont satisfaites, le démarrage de cycle se fait directement après 2 secondes de l'appui sur le bouton « marche générale ». Mais si ces conditions initiales ne sont pas satisfaites, on trouve deux cas possibles : soit il y a un défaut au niveau du piston, un manque de découpe et le convoyeur est à l'état initial, dans ce cas l'opérateur doit régler le « défaut piston » et charger le magasin par des découpes en cartons qui sont détectées par le capteur de présence à proximité photoélectrique « présence dépilage » et l'étape 20 est activée, maintenant il peut démarrer le cycle. Le deuxième cas à vérifier est si on a une découpe en carton sur le convoyeur à taquets et ce dernier n'est pas à sa position initiale dans ce cas il doit le ramener à son état initial grâce à un moteur électrique et un codeur qui mesure la vitesse et la position exacte de ce convoyeur « Transfert_Taquet », le « Transfert de taquet » s'arrête lorsque le taquet arrive à son état initial qui sera détecté par le capteur « Arrêt transfert ». Après la vérification des conditions initiales l'opérateur peut maintenant démarrer le cycle quand il clique sur le bouton « marche générale » une temporisation « TM0 » se déclenche pour réinitialiser les paramètres de la barquetteuse.

Après la temporisation « TM0 » atteinte 2 seconds et la station de collage est prête « détecté par un capteur « Colle_chaude » et la découpe est présentée ou elle est absente mais il faut qu'au même temps la temporisation TM9 sera désactivée cela fait si la découpe est présentée, maintenant on garde les mêmes conditions précédentes et on vérifie si la découpe est sur le convoyeur à taquets ou non (cela fait par un capteur à proximité inductif « présence dépilage »), **si non** le piston de margeur et la pompe à vide actionnent, le premier pour déplacer le margeur vers le magasin, la deuxième pour alimenter les ventouses qui ont un rôle de prendre une découpe, un capteur « haut prise » détecte que le margeur est arrivé au magasin et prend une découpe, le margeur dépose la découpe sur le convoyeur à taquets(ce margeur revient à son état initial grâce au piston qui est simple effet doc le retour est fait par ressort), la détection de la découpe sur le convoyeur se fait par un capteur à proximité « Présence dépilage », mais si la découpe est déjà présentée l'étape suivante sera activer directement ce qui fait déclencher une temporisation « TM1 » qui dure 0.2 s, maintenant l'étape actuelle est activée et l'étape 10 aussi est activé(est une autre étape initiale), pour passer à l'étape suivante il faut vérifier aussi que le convoyeur à taquets est à l'état initial et que la temporisation « TM10 » qui dure 2s sera désactiver (cette temporisation sera désactiver si seulement si « Demande caisse » est activé pour éviter le bourrage au niveau de convoyeur de sortie), cela provoque à actionner le moteur électrique « transfert taquet » et la station de colle commence à éjecter la colle chaude « EV_Colle » et on vérifie que le convoyeur à taquets(le taquet qui fait transférer la première découpe) n'atteint pas son état initial, le collage maintient ce qui déclenche une temporisateur « TM6 » qui dure 100ms et on vérifie qu'il n'y a pas le dépilage, cela permet de redémarrer le cycle de dépilage et au même temps le collage est terminé (le collage est contrôlé par une station de colle ITW Challenger QUATTRO) et le convoyeur à taquets arrive à son état initial et la dernière étape s'active, pour revenir à l'étape initial 10 il faut que le piston de formation de caisse sera en position haute, les vérins de pressage aussi revient à l'état initial et bouton « Marche générale » sera toujours actif.

III.2.2 Formation de caisse

L'opérateur a déjà vérifié les conditions initiales de la partie précédente, dans cette partie il faut qu'il vérifie au même temps que le piston de formation de caisse est en position haute (au même temps avec « marche générale » et qu'il n'y a pas défaut piston et l'étape 20 est active), si le piston n'est pas dans sa position initiale (position haute) le "moteur piston" actionne et le ramène à sa position initiale, sinon l'étape suivante sera active.

La présence de colle et l'appui sur le bouton "marche générale" qui déclenche une temporisateur "TM0" permet d'activer l'étape suivante, après si la dernière étape de "Dépilage_Transfert_Collage" est atteint (le dépilage et le collage sont terminés et la découpe présentée devant la cavité), la plaque où la découpe se situer remonte et le "moteur piston" actionne pour descendre le piston, si le piston arrive à la position basse qui sera détectée par le capteur à proximité inductif "piston bas" les électrovannes régulatrices s'activent donc les vérins poussent les équerres pour bien positionner et fixé la découpe collé cela pendant 0.2s (TM3 qui sera déclenche) la fin de la temporisation "TM3" permet de passer vers l'étape suivante, mais si la temporisateur "TM2 " qui s'active (elle dure 2s) cela signifie que le piston n'arrive pas à la position souhaité et pour passer à l'étape suivante il faut désactiver le bouton "marche générale" après activer le à nouveau.

Si l'un des deux cas précédents est réalisé les électrovannes de pressage actionnent les deux vérins associés à chaque électrovanne, cela permet de maintenir et garantir le collage de la découpe et la formation de caisse cela dure "TM4" = 0.7s, quand cette temporisation termine et il y a demande caisse (cela permet de désactiver la temporisation "TM10") et il n'y a pas de bourrage au niveau de convoyeur de sortie, le "moteur piston" actionne qui permet au piston de remonter et la plaque "rampe" descend donc le piston n'est pas à la position basse ce qui provoque la continuité de remonte de piston, la descente de la plaque "rampe" et le déclenchement de la temporisation "TM5" qui dure 0.05s (cette temporisation sert à l'anticipation transfert), si le piston arrive à la position haute ou "TM5" termine le capteur à contact de bourrage détecte une caisse et le "moteur piston" actionne aussi pour assurer que le piston revient à sa position initiale. Si le bouton "marche générale" reste appui le cycle redémarre.

III.2.3 Forçage Piston

Pour passer de l'étape initiale "X20" vers l'étape suivant il faut que les deux étapes précédentes soient en début de cycle et l'opérateur doit appui sur l'un des boutons poussoirs "Forçage piston bas ou Forçage piston haut".

Si l'opérateur choisir de forcer le piston vers le bas il faut que le bouton poussoir "Forçage piston bas" reste activer et le bouton "Forçage piston haut" soit désactivé et le piston n'est pas à la position basse, le piston sera forcé vers le bas. Pour qu'il revient à sa position initiale il faut relâcher le bouton poussoir "Forçage piston bas" ou il arrive à la position basse.

Si l'opérateur choisit de forcer le piston vers le haut il faut que le bouton poussoir "Forçage piston haut" reste activer et le bouton "Forçage piston bas" soit désactivé et le piston n'est pas

à la position haute, le piston sera forcé vers le haut, pour qu'il revienne à sa position initiale il faut relâcher le bouton poussoir "Forçage piston haut" ou il arrive à la position haute. Ce fonctionnement est utilisé en cas de problème avec le moteur principal ou le système de bielle-manivelle. Pourrait être utilisé pour effectuer des ajustements précis de la position du piston pendant des phases spécifiques du processus. Par exemple, les boutons poussoirs pourraient permettre de surélever légèrement le piston pour faciliter le chargement ou le déchargement des barquettes.

III.3 GRAFCETs

Après avoir étudié le cahier de charges de la barquetteuse, on a trouvé 14 entrées et 10 sorties qui sont mentionnées dans les deux tableaux suivants :

Les entrées	Les adresses
Marche Générale	I0.0
Colle chaude	I0.1
Arrêt transfert	I0.2
Haut prise	I0.3
Présence dépilage	I0.4
Présence colle	I0.5
Piston Haut	I0.6
Piston Bas	I0.7
Demande caisse	I1.0
Présence découpe	I1.1
Piston prêt	I1.2
Forçage piston bas	I1.3
Cellule bourrage	I1.4
Forçage piston haut	I1.5

(1)

Les sorties	Les adresses
Transfert taquet	Q0.0
EV_Prise	Q0.1
EV_Piston	Q0.2
EV_Colle	Q0.3
EV_Rampe	Q0.4
EV_Equerrage	Q0.5
EV_Pressage	Q0.6
Moteur Piston	Q0.7
Forçage Piston Haut	Q1.0
Forçage Piston Bas	Q1.1

(2)

Tableau III.1 : Table des Entrées/Sorties de l'API

(1) Table des Entrées d'API

(2) Table des Sorties d'API

Pour traduire le cahier de charges en GRAFCET, il est essentiel de suivre les règles d'évolution d'un GRAFCET tel que la bipartie qui définit la relation entre les étapes et les transitions. Un Grafcet est dit **biparti** si l'ensemble de ses étapes et transitions peut être divisé en deux groupes distincts, ne contenant aucun arc reliant des éléments du même groupe.

Cette étude nous a permis de donner un GRAFCET structuré et fonctionnel.

III.3.1 Dépilage_Transfert_Collage

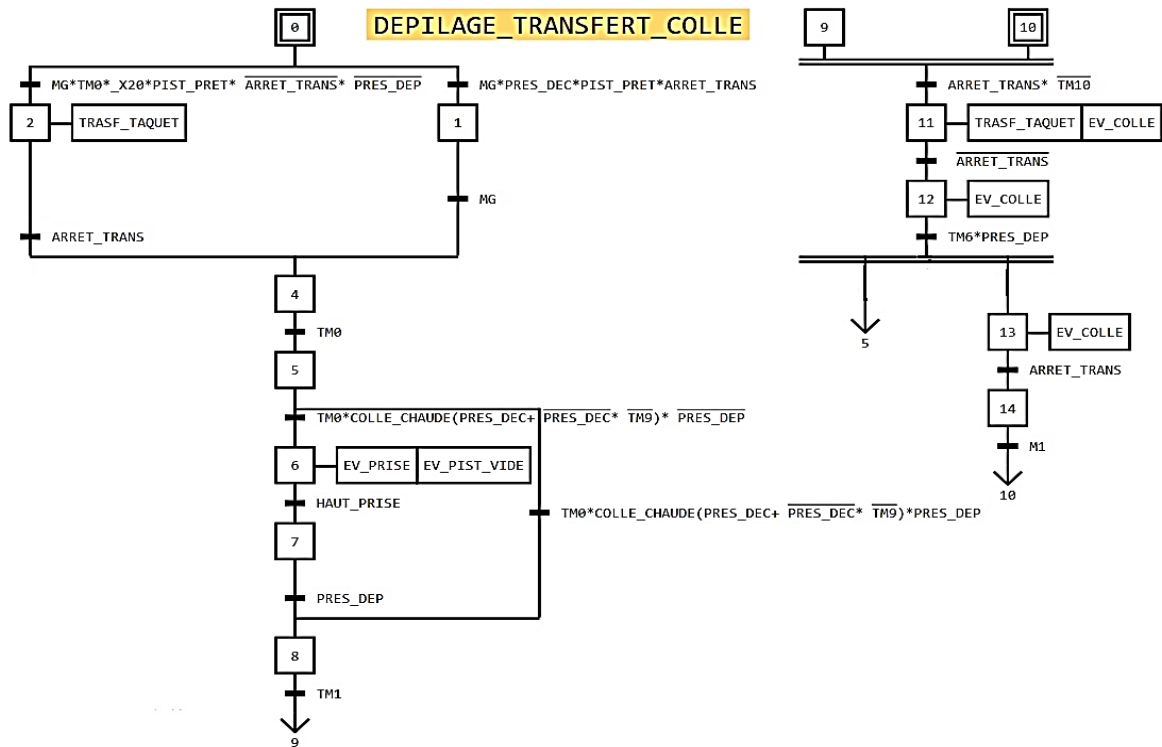


Figure III. 1: Grafcet de "Dépilage_Transfert_Collage"

III.3.2 Formation de Caisse

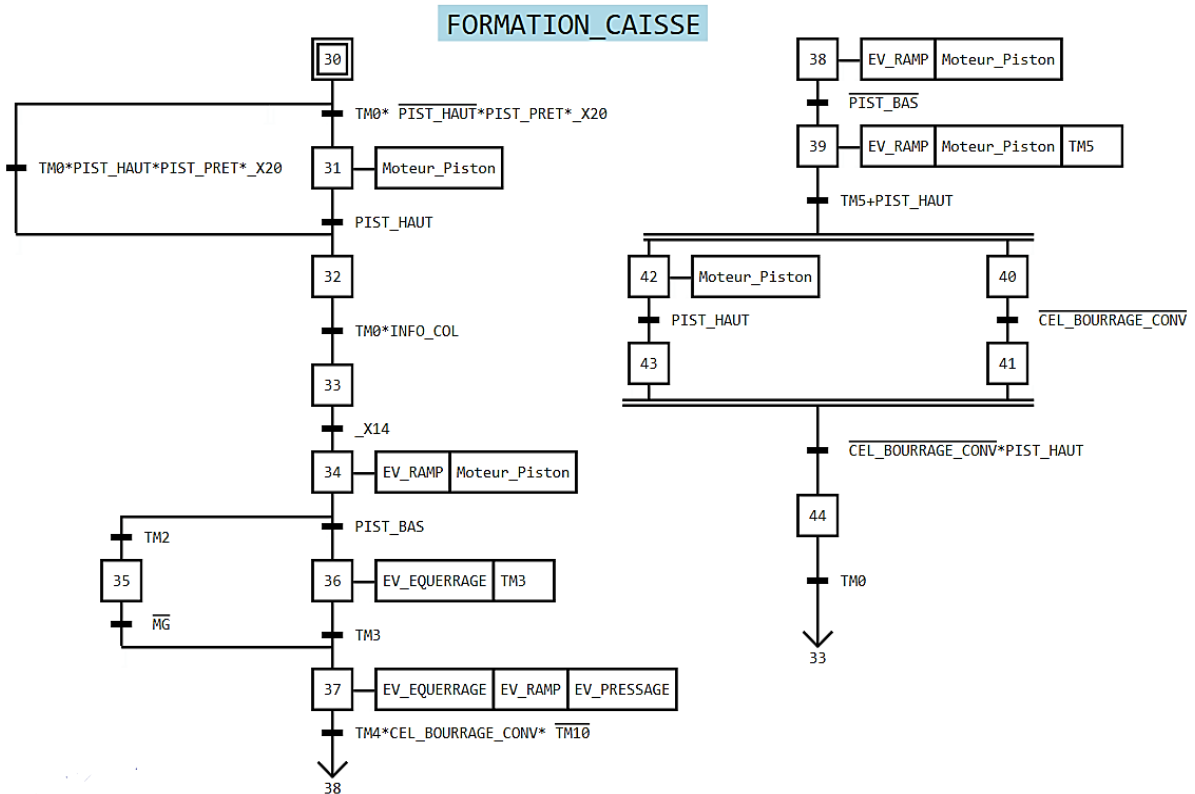


Figure III. 2 : Grafcet de "Formation de caisse"

III.3.3 Forçage piston

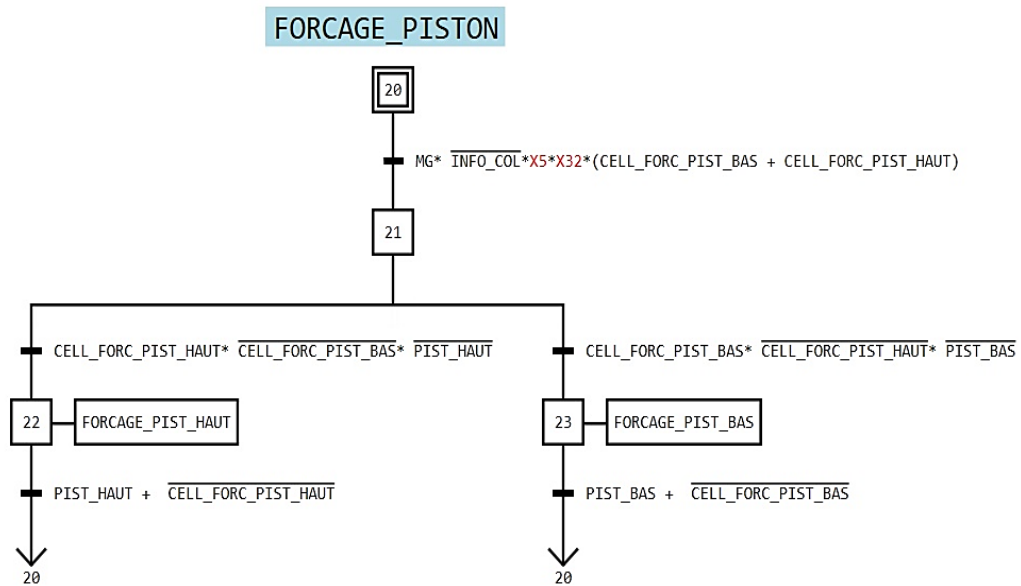


Figure III. 3 : Grafcet de "Forçage piston"

Les GRAFCETs réalisés sur PL7 PRO sont mentionnés dans l'annexe A.

III.4 Migration vers SIEMENS S7 1200 ?

- **Les inconvénients de l'automate Schneider TSX micro :**

Dans cette section nous allons exposer les raisons de la migration de l'automate Schneider TSX Micro vers SIEMENS S7 1200.

- **Capacité de traitement :** Par rapport aux automates plus récents, le TSX micro est moins puissant en vue de traitement des données, ce qui peut limiter la vitesse de la machine ou la complexité des programmes qu'il peut exécuter. Cela peut être un inconvénient si la machine doit gérer de gros volumes de produits ou des tâches complexes.
- **La programmation** utilisée par Schneider, le langage de programmation Codesys, peut-être moins intuitif ou convivial que d'autres langages de programmation utilisés par des automates concurrents. La maintenance et les futures modifications du programme de la machine peuvent être rendues plus complexes, surtout pour les utilisateurs non expérimentés.
- La machine avec automate TSX Micro peut avoir un coût initial plus élevé que les solutions similaires qui utilisent des automates plus récents. Il est essentiel de faire une comparaison des coûts totaux d'achat, de maintenance et de mise à niveau réels sur la durée de vie de la machine.

- **Obsolescence technologique:** Comme c'est le cas pour tout appareil technologique, l'automate TSX Micro peut devenir obsolète au fil du temps. Les avancées technologiques peuvent proposer des fonctionnalités améliorées, une efficacité accrue et une meilleure compatibilité avec d'autres systèmes.

La disponibilité des pièces détachées et du support technique pour le TSX Micro peut également devenir plus limitée avec le temps. L'assistance technique pour le TSX Micro peut devenir plus difficile à obtenir avec le temps, car Schneider concentre ses efforts sur les produits plus récents [28].

La migration vers l'automate Siemens S7 1200 nous a permis d'éviter les inconvénients cités précédemment de l'automate Schneider TSX micro, on peut citer autres raisons peuvent motiver cette migration :

- **Technologie plus récente :** Le S7-1200 utilise une technologie plus récente que le TSX Micro, ce qui lui permet d'offrir des performances améliorées et une fiabilité accrue. Le S7 1200 propose de nombreuses fonctionnalités avancées que le TSX Micro ne possède pas, comme la communication Ethernet, la communication maître-esclave et la compatibilité avec des langages de programmation plus récents.
- **Performances accrues:** Le S7 1200 propose des capacités de traitement et de mémoire accrues, ce qui permet une exécution plus rapide des programmes et un stockage de données plus élevé.
- **Support technique et disponibilité des pièces:** Par rapport à TSX Micro, Siemens propose un soutien technique et une disponibilité de pièces de rechange supérieures à Schneider Electric, ce qui facilite la maintenance et l'évolution à long terme.
- **Sécurité améliorée :** Le S7-1200 propose des technologies de sécurité intégrées afin de garantir la protection de votre système et de vos données.
- **Intégration avec TIA Portal :** Le S7-1200 est complètement compatible avec ce logiciel. Cela facilite et accélère la programmation, la mise en service et le diagnostic.

III.5 Choix de la CPU

Comme on a 14 entrées et 10 sorties, et après avoir analysé la fiche technique de la barquetteuse on a trouvé que sa partie commande est alimentée par une tension de **24V DC**, on a décidé de choisir la CPU **1214c DC/DC/RLY** comme on peut choisir aussi la CPU 1214c DC/DC/DC mais comme ils ont des caractéristiques communes : sont alimentées avec 24 V DC, possèdent une mémoire de 100 KB, 14 entrées TOR, 10 sorties TOR, 2 sorties analogiques

et une interface de communication PROFINET, et offrant l'option du Web server. La CPU 1214 DC/DC/RLY a des caractéristiques supplémentaires tel que l'alimentation (24 V DC ou **85-264 V AC**), et au niveau d'application (Commande d'acteurs à **courant alternatif** ou continu). Ces caractéristiques supplémentaires nous ont poussé à choisir la CPU **1214c DC/DC/RLY**.

III.6 Programmation

Comme notre automate Siemens s7 1200 ne prend pas en charge GRAFCET, on a choisi de traduire les GRAFCETs précédents en langage Ladder.

- ❖ **Solution 1** : Le GRAFCET est composé d'étapes qu'on va les représenter en utilisant des mémoires internes de type bool %M où on choisit une variable pour chaque étape dont les équations d'activations et désactivations sont comme suit :

Tableau 5 : Table des CAXn et CDXn

Etapes	CAXn	CDXn
0	Init	(x1+x2). \overline{Init}
1	x0.TM0.x20.Présence_découpe.Piston_Pret.Arret_transfert. \overline{Init}	x4+Init
2	x0.TM0.x20.Présence_découpe. Piston_Pret. $\overline{Arret_transfert.Présence_dépilage.Init}$	x3+Init
4	(x1.Marche_Générale+x2.Arret_transfert). \overline{Init}	x5+Init
5	(x4.TM0+x12.TM6. $\overline{Présence_dépilage}$). \overline{Init}	x6+x8+Init
6	(x5.TM0.Colle_chaude(Présence_découpe+ $\overline{Présence_dépilage.TM9}$) $\overline{Présence_dépilage}$) \overline{Init}	x7+Init
7	x6.Haut_prise. \overline{Init}	x8+Init
8	(x7.Présence_dépilage+x5.TM0.Colle_chaude(Présence_découpe+ $\overline{Présence_dépilage.TM9}$) $\overline{Présence_dépilage}$) \overline{Init}	x9+Init
9	x8.TM1. \overline{Init}	x11+Init
10	x14.M1+Init	x11. \overline{Init}
11	x9.x10.Arret_transfert. $\overline{TM10.Init}$	x12+Init
12	x11. $\overline{Arret_transfert.Init}$	x13.x5+Init
13	x12.TM6. $\overline{Présence_dépilage.Init}$	x14+Init
14	x13.Arret_transfert. \overline{Init}	x10+Init
20	x22(Piston_Haut+ $\overline{Forçage_piston_haut}$)+x23(Piston_Bas+ $\overline{Forçage_piston_bas}$)+Init	x21. \overline{Init}
21	x20.Marche_Générale. $\overline{Colle_chaude}$.x5.x32($\overline{Forçage_piston_haut}$ + $\overline{Forçage_piston_bas}$) \overline{Init}	x22+x23+Init
22	x21. $\overline{Forçage_piston_haut}$. $\overline{Forçage_piston_bas}$. $\overline{Piston_Haut}$. \overline{Init}	x20+Init
23	x21. $\overline{Forçage_piston_bas}$. $\overline{Forçage_piston_haut}$. $\overline{Piston_Bas}$. \overline{Init}	x20+Init
30	Init	(x31+x32) \overline{Init}
31	x30.TM0. $\overline{Piston_Haut}$.Piston_pret.x20. \overline{Init}	x32+Init
32	(x30.TM0.Piston_Haut. Piston_pret.x20+x31. Piston_Haut) \overline{Init}	x33+Init
33	(x32.TM0.Colle_chaude+x44.TM0) \overline{Init}	x34+Init
34	x33.x14. \overline{Init}	x35+x36+Init
35	x34.TM2. \overline{Init}	x37+Init
36	x34.Piston_bas. \overline{Init}	x37+Init

37	$(x35.\overline{Marche_Générale}+x36.TM3)\overline{Init}$	x38+Init
38	$x37.TM4.\overline{Cellule_bourage.TM10.Init}$	x39+Init
39	$x38.\overline{Piston_Bas.Init}$	x40.x42+Init
40	$x39(TM5+Piston_Haut)\overline{Init}$	x41+Init
41	$x40.\overline{Cellule_bourage.Init}$	x44+Init
42	$x39(TM5+Piston_Haut)\overline{Init}$	x43+Init
43	$x42.Piston_Haut.\overline{Init}$	x44+Init
44	$x41.x43.\overline{Cellule_bourage.Piston_Haut.Init}$	x33+Init

III.6.1 La conception du programme sur TIA PORTAL V16

Pour créer un nouveau projet sur TIA PORTAL V16 il faut suivre les étapes suivantes :

- Créer un nouveau projet
- Sélectionner le matériel à utiliser d'après les références (configuration matériels)
- Ajouter les blocs de données et les blocs fonctionnels
- Remplir la table des variables et les blocs de données
- Construire le programme dans le bloc main OB1 et les blocs fonctionnels
- Charger la configuration après compilation
- Exécuter le programme et effectuer les tests nécessaires, pour confirmer la bonne exécution de l'opération.

La figure suivante représente l'organigramme pour la création d'un projet sous TIA PORTAL V16 :

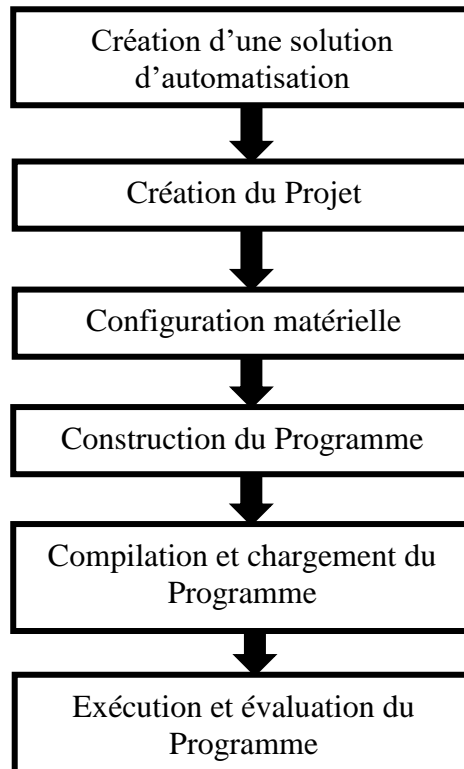


Figure III. 4 : L'organigramme pour création d'un projet sous TIA PORTAL V16

III.6.2 Configuration matérielle

Après la création du projet, il est possible de configurer la station de travail. La première étape implique de déterminer le matériel déjà disponible. Pour cela, il est possible de se rendre dans la vue du projet et de sélectionner l'option « ajouter un appareil » dans le navigateur du projet, une autre fenêtre apparaît au centre de l'écran comme sur la figure III.5, qui affiche tous les appareils qu'on peut ajouter à notre projet.

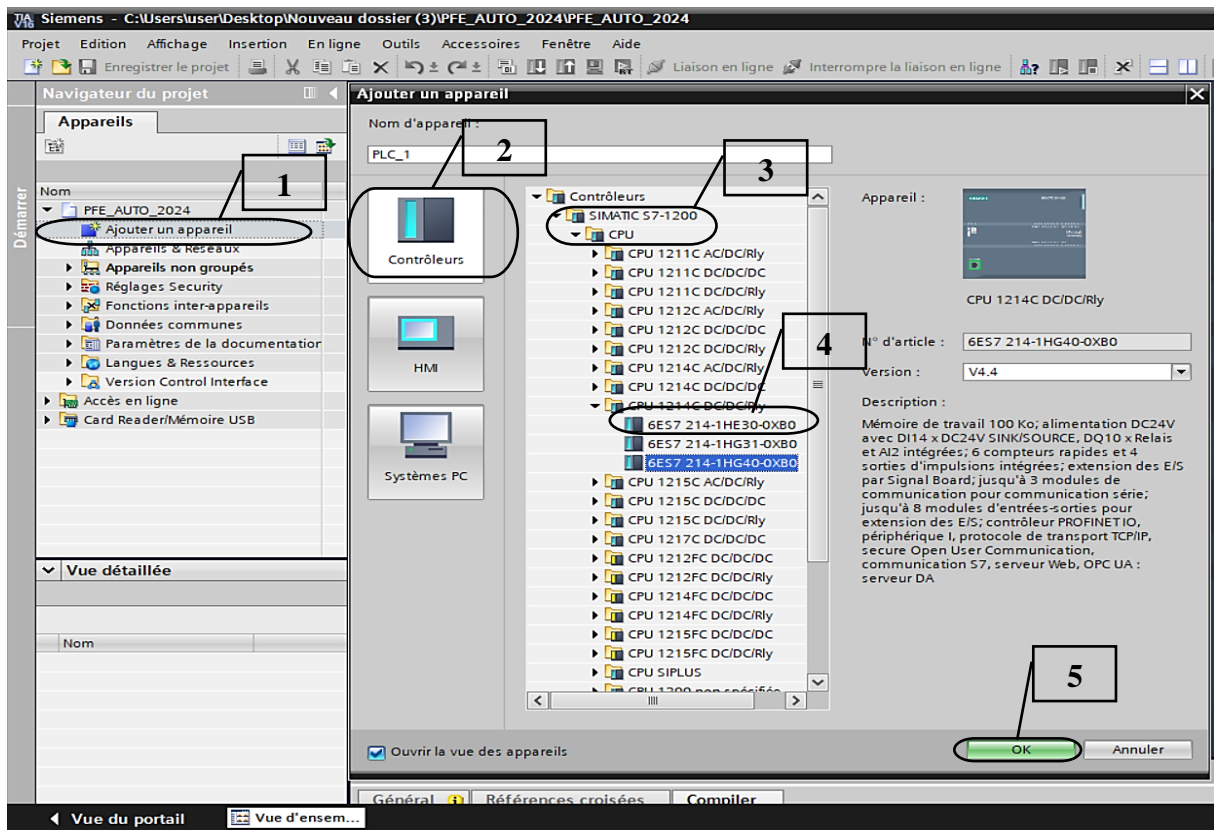


Figure III. 5 : Configuration matérielle

- (1) on sélectionne "Ajouter un appareil"
- (2) on sélectionne "contrôleurs"
- (3) on ouvre le dossier "SIMATIC S7-1200" puis le dossier "CPU 1214C DC/DC/Rly"
- (4) on choisit la référence "6ES7 214-1HG40-0XB0"
- (5) enfin on clique sur "OK" et la fenêtre "configuration de l'appareil" s'ouvre, comme indique la figure III.6.

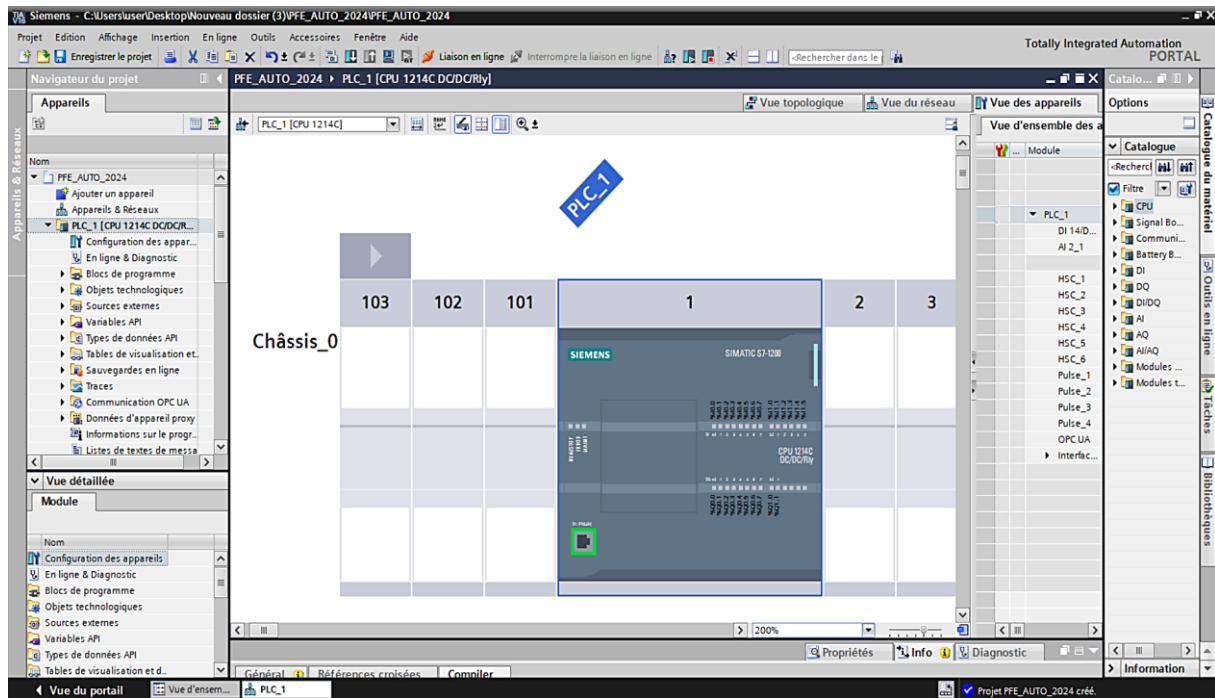


Figure III. 6: Configuration de l'appareil

III.6.3 Table des variables API

Pour une meilleure organisation des variables API, nous les avons sectionnées en différents tableaux, chaque tableau correspond à des variables spécifiques pour faciliter la localisation et la consultation des variables, la figure III.7 montre ces tableaux.

Pour TIA Portal, la table de variables est l'outil qui permet de créer un adressage symbolique. C'est dans cette table qu'on va pouvoir déclarer toutes les variables et constantes utilisées dans le programme. Lorsqu'on définit une variable API, il faut définir :

- Un nom : c'est l'adressage symbolique de la variable
- Le type de donnée : REAL, BOOL, INT... etc
- L'adresse absolue : par exemple %Q0.6, %I1.0.

On peut aussi insérer un commentaire pour chaque variable, ce commentair peut etre visible dans chaque réseau qui utilise cette variable.



Figure III. 7: Les tableaux des variables d'API

Comme on a 14 entrées TOR, dans le tableau des entrées « INPUTS » on a utilisé un adressage %I qui joue un rôle clé dans l'interaction avec les entrées numériques des automates S7 dans TIA Portal, offrant ainsi une programmation claire, efficace et facilement maintenable.

La syntaxe complète de cette adressage est : **%I<Numéro d'entrée digitale>**

INPUTS				
		Nom	Type de données	Adresse
1		Marche_Générale	Bool	%I0.0
2		Colle_chaude	Bool	%I0.1
3		Arrêt_transfert	Bool	%I0.2
4		HAUT_PRISE	Bool	%I0.3
5		PRESENCE_DEPILAGE	Bool	%I0.4
6		PRESENCE_COLLE	Bool	%I0.5
7		PISTON_HAUT	Bool	%I0.6
8		PISTON_BAS	Bool	%I0.7
9		DEMANDE_CAISSE	Bool	%I1.0
10		Présence_découpe	Bool	%I1.1
11		Piston_prêt	Bool	%I1.2
12		FORCAGE_PISTON_BAS	Bool	%I1.3
13		Cellule_bourrage	Bool	%I1.4
14		FORCAGE_PISTON_HAUT	Bool	%I1.5

Figure III. 8 : La table des variables des entrées d'API

Comme on a 10 sorties TOR, dans le tableau des sorties « OUTPUTS » on a utilisé un adressage %Q qui joue un rôle essentiel dans la gestion des sorties numériques des automates S7 dans TIA Portal.

La syntaxe complète de cette adressage est : **%Q<Numéro de sortie digitale>**

OUTPUTS				
		Nom	Type de données	Adresse
1		Transfert_Taquet	Bool	%Q0.0
2		EV_Prise	Bool	%Q0.1
3		EV_Piston	Bool	%Q0.2
4		EV_Colle	Bool	%Q0.3
5		EV_Rampe	Bool	%Q0.4
6		EV_Equerrage	Bool	%Q0.5
7		EV_Pressage	Bool	%Q0.6
8		Moteur_piston	Bool	%Q0.7
9		Forçage_Piston_Haut	Bool	%Q1.0
10		Forçage_Piston_Bas	Bool	%Q1.1

Figure III. 9: La table des variables des sorties d'API

Comme on a considéré les étape du GRAFCET comme des mémoires interne donc on doit utilisé l'adressage %M qui est un outil essentiel pour gérer la mémoire de marque des automates S7 dans TIA Portal. La syntaxe complète de cette adressage est :

%M <Numéro de mémoire de marque>

On peut utiliser l'adresse %M dans diverses instructions du langage STEP 7, telles que :

- **Lecture d'une valeur:** Lire la valeur stockée dans une mémoire de marque.
- **Écriture d'une valeur:** Affecter une valeur à une mémoire de marque.
- **Comparaison de valeurs:** Comparer la valeur d'une mémoire de marque à une autre valeur ou à une constante.
- **Stockage d'états de programme:** Signaler l'état d'une opération ou d'un processus.

ETAPES				
		Nom	Type de données	Adresse
1		X0	Bool	%M10.0
2		X1	Bool	%M10.1
3		X2	Bool	%M10.2
4		X3	Bool	%M10.3
5		X4	Bool	%M10.4
6		X5	Bool	%M10.5
7		X6	Bool	%M10.6
8		X7	Bool	%M10.7
9		X8	Bool	%M11.0
10		X9	Bool	%M11.1
11		X10	Bool	%M11.2
12		X11	Bool	%M11.3
13		X12	Bool	%M11.4
14		X13	Bool	%M11.5
15		X14	Bool	%M11.6
16		X20	Bool	%M11.7
17		X21	Bool	%M12.0
18		X22	Bool	%M12.1
19		X23	Bool	%M12.2
20		X30	Bool	%M12.3
21		X31	Bool	%M12.4
22		X32	Bool	%M12.5
23		X33	Bool	%M12.6
24		X34	Bool	%M12.7
25		X35	Bool	%M13.0
26		X36	Bool	%M13.1
27		X37	Bool	%M13.2
28		X38	Bool	%M13.3
29		X39	Bool	%M13.4
30		X40	Bool	%M13.5
31		X41	Bool	%M13.6
32		X42	Bool	%M13.7
33		X43	Bool	%M14.0
34		X44	Bool	%M14.1

Figure III. 10 : La table des étapes

LES TRANS				
		Nom	Type de données	Adresse
1		t0_1	Bool	%M0.0
2		t0_2	Bool	%M0.1
3		t2_3	Bool	%M0.2
4		t1_4	Bool	%M0.3
5		t3_4	Bool	%M0.4
6		t4_5	Bool	%M0.5
7		t5_6	Bool	%M0.6
8		t5_8	Bool	%M0.7
9		t6_7	Bool	%M1.0
10		t7_8	Bool	%M1.1
11		t8_9	Bool	%M1.2
12		t9_11	Bool	%M1.3
13		t11_12	Bool	%M1.4
14		t12_5	Bool	%M1.5
15		t13_14	Bool	%M1.6
16		t14_10	Bool	%M1.7
17		t12_13	Bool	%M2.0
18		t9-10_11	Bool	%M2.1
19		t20_21	Bool	%M2.2
20		t21_22	Bool	%M2.3
21		t21_23	Bool	%M2.4
22		t23_20	Bool	%M2.5
23		t22_20	Bool	%M2.6
24		t30_31	Bool	%M2.7
25		t30_32	Bool	%M3.6
26		t31_32	Bool	%M3.7
27		t32_t33	Bool	%M4.0
28		t44_t33	Bool	%M4.1
29		t33_t34	Bool	%M4.2
30		t34_t35	Bool	%M4.3
31		t34_t36	Bool	%M4.4
32		t35_t37	Bool	%M4.5
33		t36_t37	Bool	%M4.6
34		t37_t38	Bool	%M4.7
35		t38_t39	Bool	%M5.0
36		t39_t42-40	Bool	%M5.1
37		t40_t41	Bool	%M5.2
38		t42_t43	Bool	%M5.3
39		t41-43_t44	Bool	%M5.4
40		t10_11	Bool	%M5.5

Figure III. 11 : La table des transitions

PRL				
		Nom	Type de données	Adresse
1		S21	Bool	%M3.0
2		M1	Bool	%M3.2

Figure III. 12 : La table des variables de la partie PRL

III.6.4 Tableau d'affectation

L'utilisation du tableau d'affectation est extrêmement bénéfique pour connaître les entrées, sorties et mémoires déjà employées. Pour y accéder on click (click droit) d'abord sur (1) « PLC_1[CPU1214c DC/DC/Rly] » après on sélectionne (2) « Tableau d'affectation »

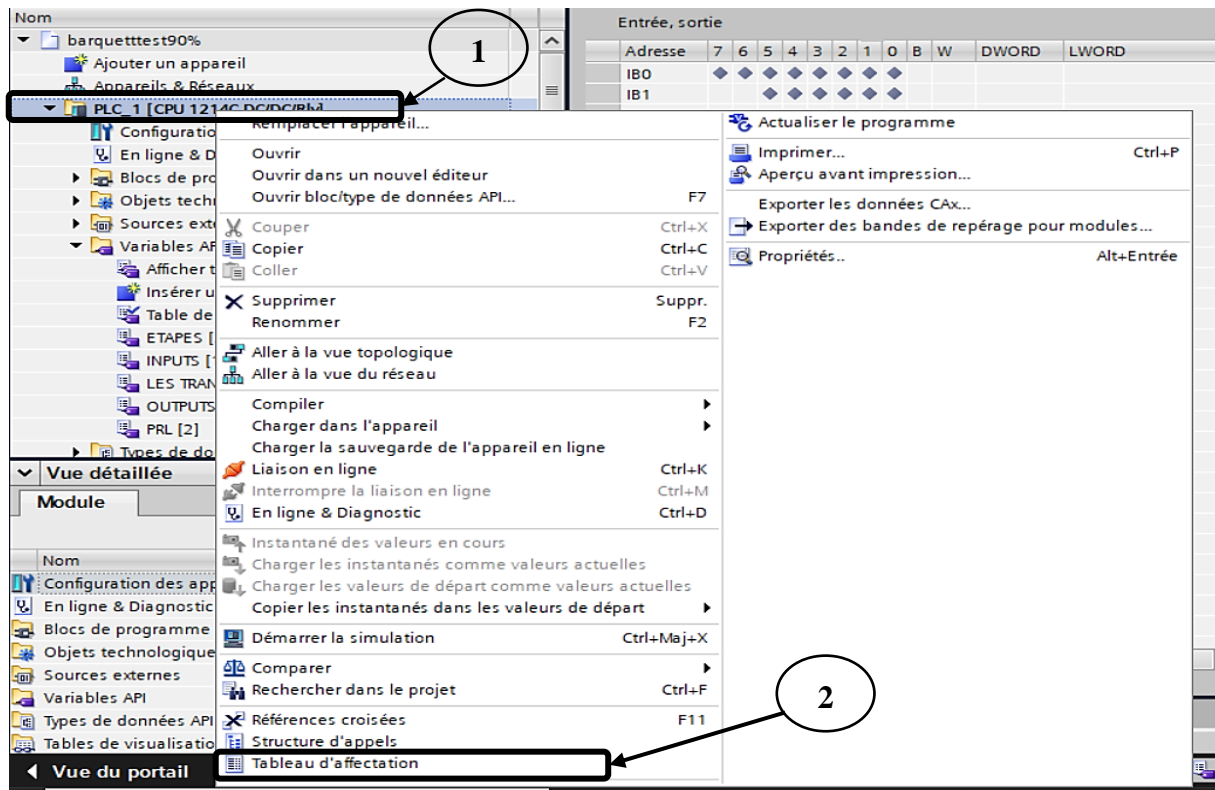


Figure III. 13: Etapes d'affichage du tableau d'affectation

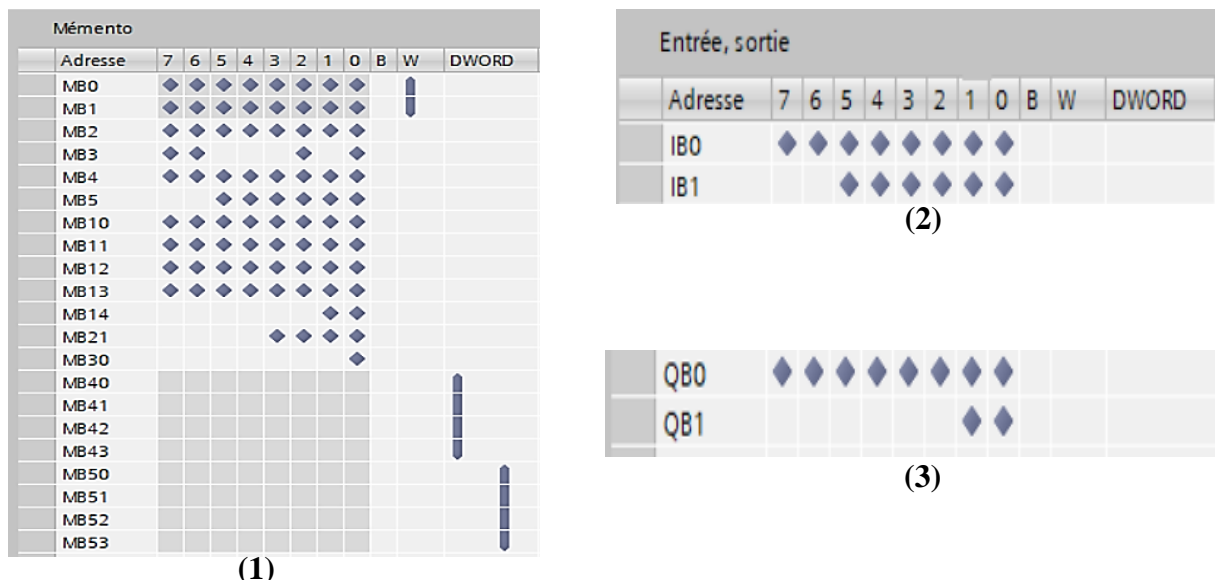


Figure III. 14 : Tableau d'affectation

- (1) Tableau d'affectation des mémoires
- (2) Tableau d'affectation des entrées
- (3) Tableau d'affectation des sorties

III.6.5 Les blocs dans TIA Portal

- **Les blocs de données-DB**

Dans notre projet les deux types de DB sont créés, la figure III.15 montre le bloc DB global ajouté « DB_INPUT » et la figure III.16 montre le DB d'instance qui été créé lors de la création du bloc FB « LES TEMPOS ».

DB_INPUT									
	Nom	Type de données	Valeur de départ	Rémanence	Accessible ...	Écritu...	Visible da...	Valeur de ..	Commentaire
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	I0-0	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	I0-1	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	I0-2	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5	I0-3	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6	I0-4	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
7	I0-5	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
8	I0-6	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
9	I0-7	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
10	I1-0	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
11	I1-1	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
12	I1-2	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
13	I1-3	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
14	I1-4	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
15	I1-5	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Figure III. 15 : DB global « DB_INPUT »

LES TEMPOS_DB									
	Nom	Type de données	Valeur de départ	Rémanence	Accessible ...	Écritu...	Visible da...	Valeur de ..	Commentaire
1	Input			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	Output			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
3	InOut			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
4	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figure III. 16 : DB d'instance « LES TEMPOS_DB »

- **Les fonctions – FC**

Dans notre projet on a créé cinq blocs FC :

- **DEPILAGE_TRANSFERT_COLLAGE[FC1]**

Pour organiser notre programme et nos fonctions on a créé d'abord les réseaux des transitions, ensuite on a créé les réseaux des étapes.

Les transitions :

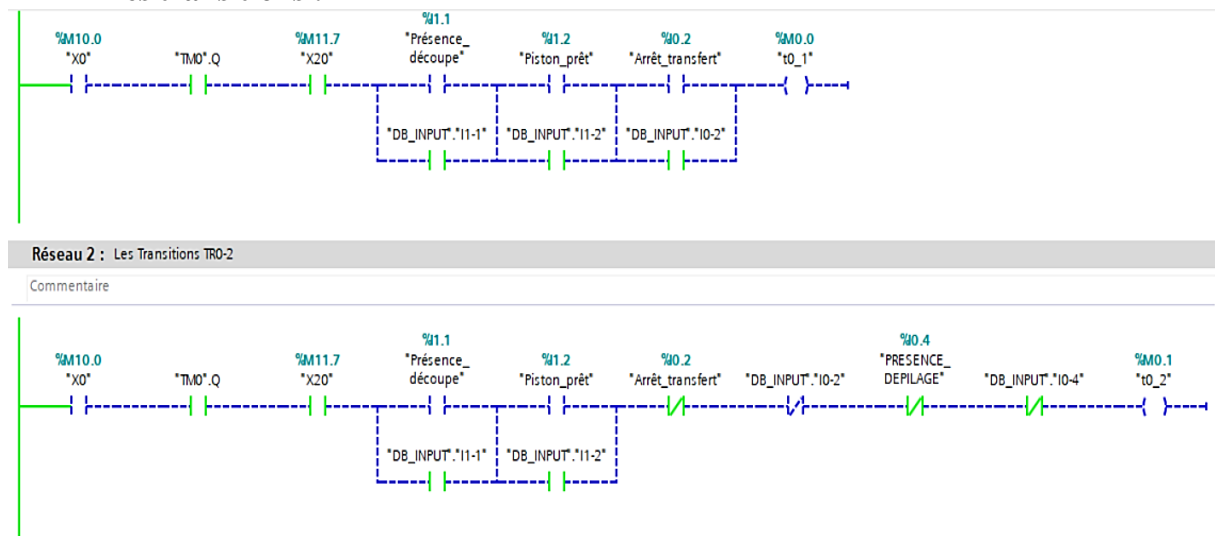


Figure III. 17: Les deux premiers transitions

On a considéré chaque transition comme un bit de mémoire interne et on a suivi la notation suivante pour chacune des transitions : tX_n-X_{n+1} cela signifie la transition qui exprime la possibilité d'évolution entre l'étape X_n et X_{n+1} .

Par exemple la transition "t0_1" est franchie lorsque elle est validée et la réceptivité associée à la transition est vraie, sachant qu'elle est validée lorsque l'étape x0 est active.

Pour chaque entrée physique on a ajouté une variable du bloc de donné "DB_INPUT", si on a un contact NO on ajoute une variable en parallèle, mais si on a un contact NC on ajoute une variable en série. Par exemple dans le premier réseau on a ajouté une variable "DB_INPUT"."IO-1" en parallèle avec l'entrée %IO.1 "Colle_chaude", dans le réseau 2 on a ajouté une variable "DB_INPUT"."IO-4" en série avec l'entrée %IO.4 "Présence_DEPILAGE" qui est sous forme d'un contact NC.

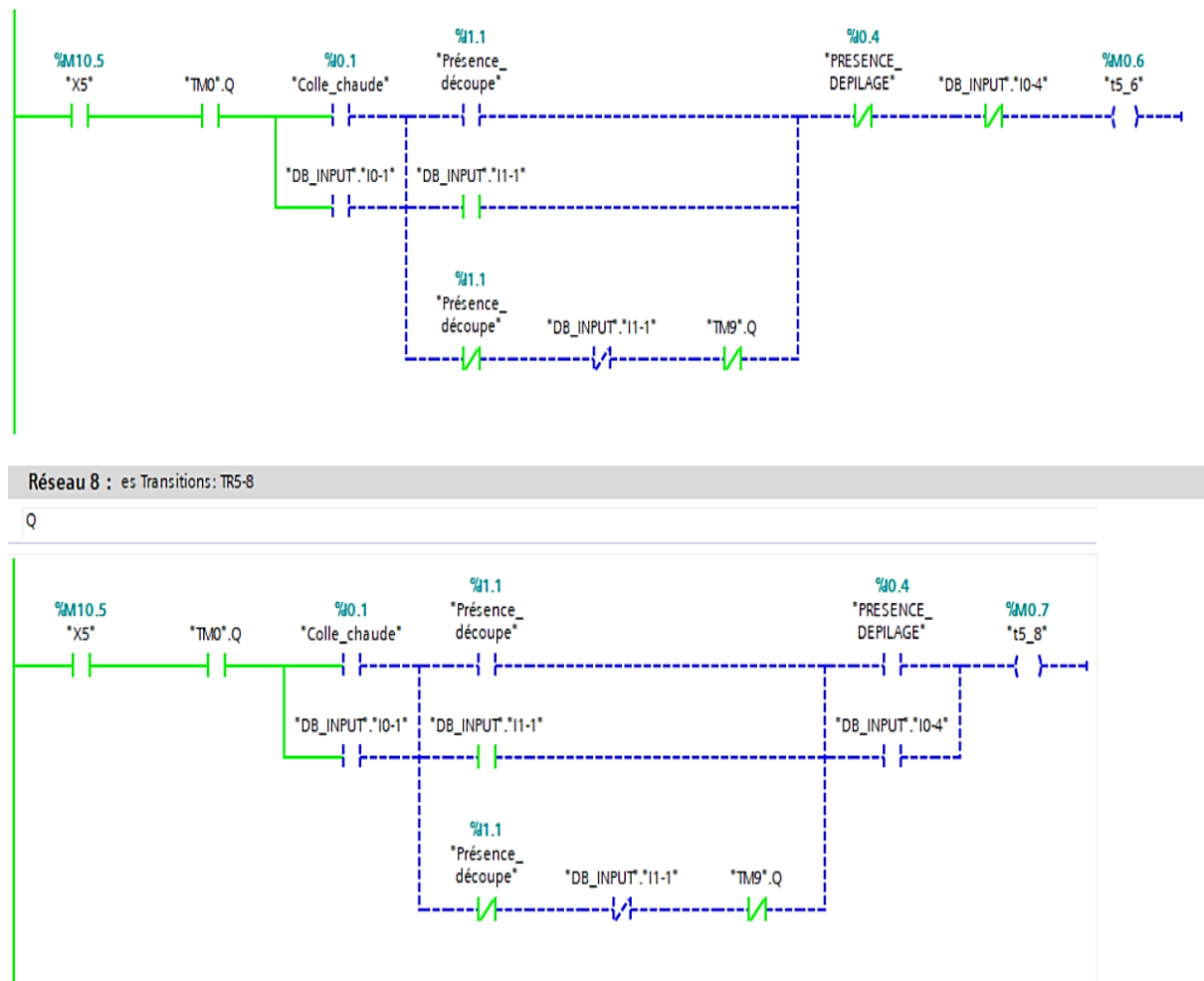


Figure III. 18: Les deux transitions t5_6 et t5_8

A partir des réseaux 7 et 8 on peut conclure qu'il y a une divergence en "OU" dans notre GRAFCET.

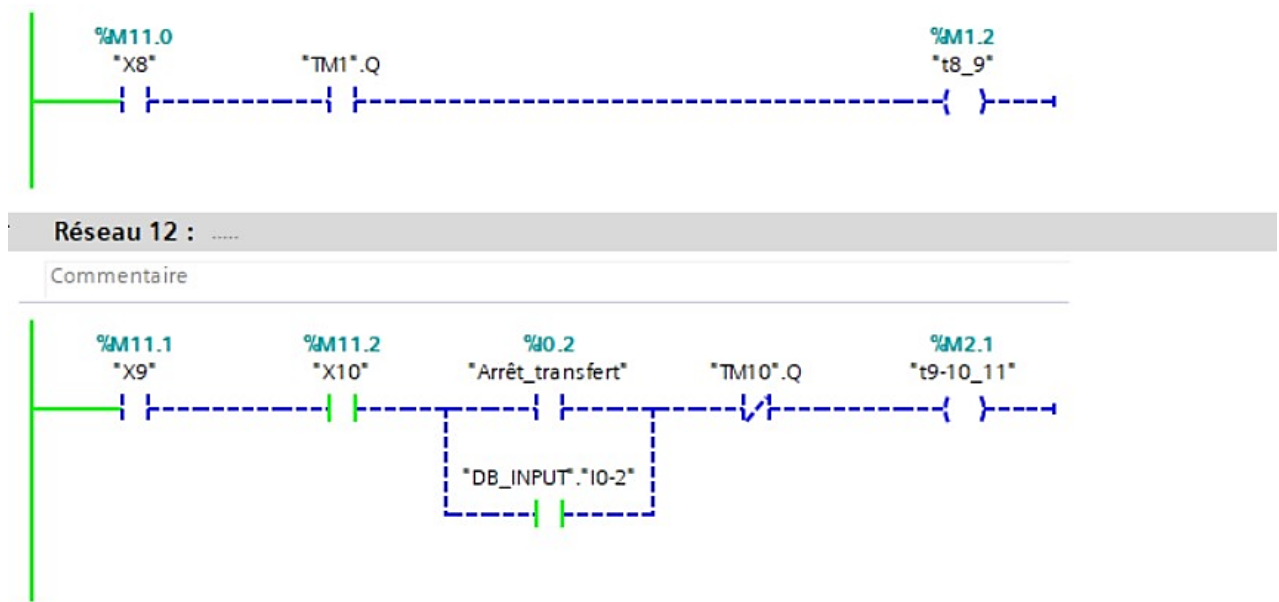


Figure III. 19 : Les deux transitions t8_9 et t9-10_11

Dans le réseau 12 on trouve la notation "t9-10_11" c'est une transition qui exprime l'évolution des étapes 9 et 10 vers l'étape 11, dans ce cas on a une convergence en "ET". On peut créer deux réseaux "t9_10" et "t10_11", mais ils ont la même réceptivité c'est pour cette raison qu'on a opté d'utiliser la notation "t9-10_11".

Pour voir la totalité des transition de ce GRAFCET voir l'annexe.

Les étapes :

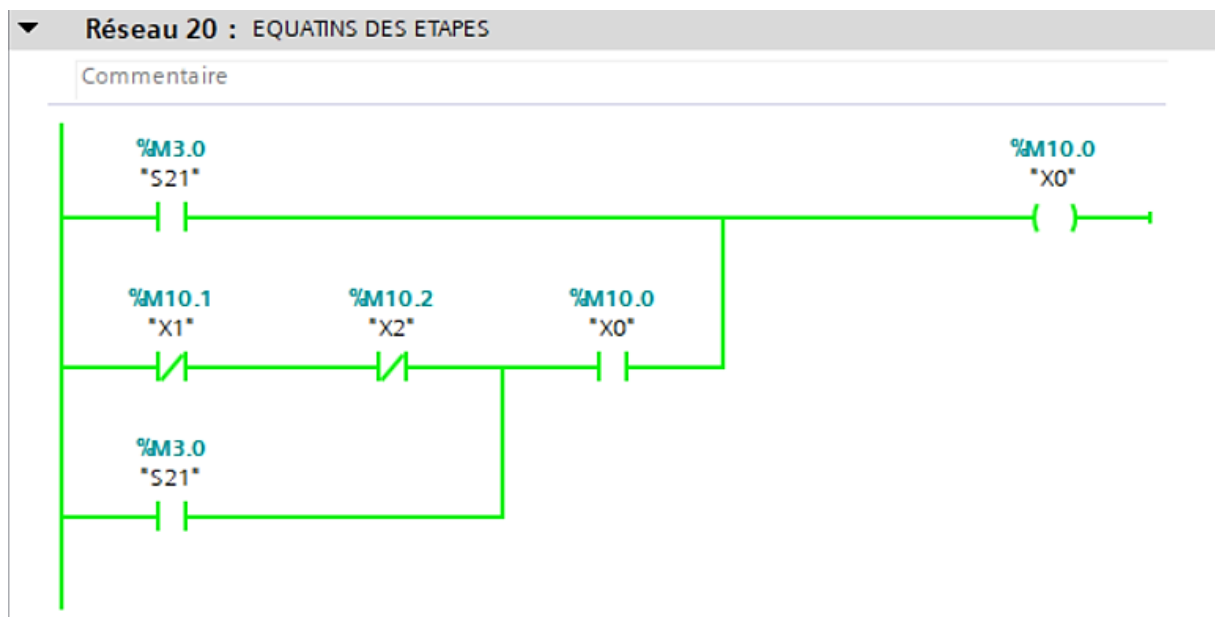


Figure III. 20 : L'étape initiale x0

L'étape initiale x0 est active à cause de bit d'initialisation S21.

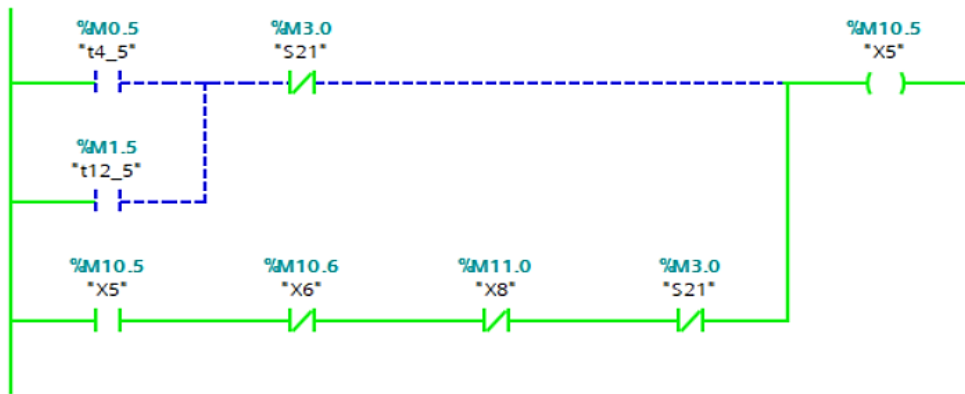


Figure III. 21 : L'étape x5

La convergence en "OU" se trouve dans les conditions d'activation de l'étape 5, la divergence en "OU" se trouve dans les conditions de désactivation de l'étape 5.

Pour voir la totalité des étapes de ce GRAFCET voir l'annexe.

- FORMATION_CAISSE[FC3]

Les Transitions :

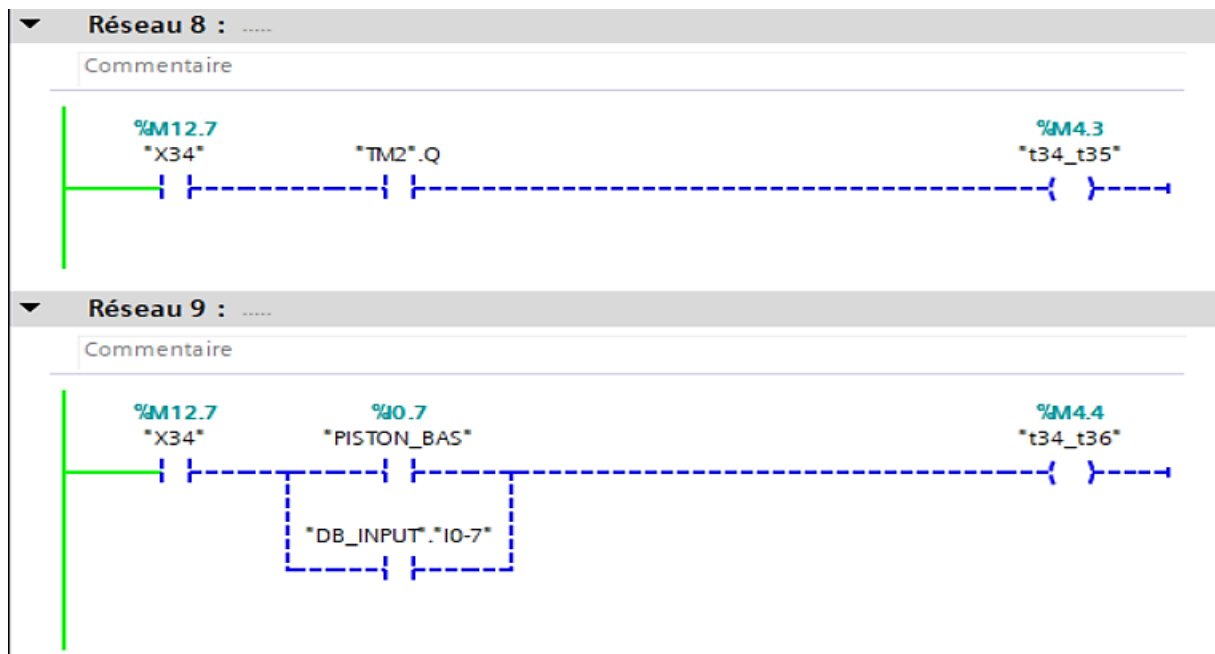


Figure III. 22 : Les deux transitions t34_35 et t34_36

A partir de ces deux réseaux (réseau 8 et 9) on peut conclure qu'il y a une divergence en "OU" après l'étape x34.

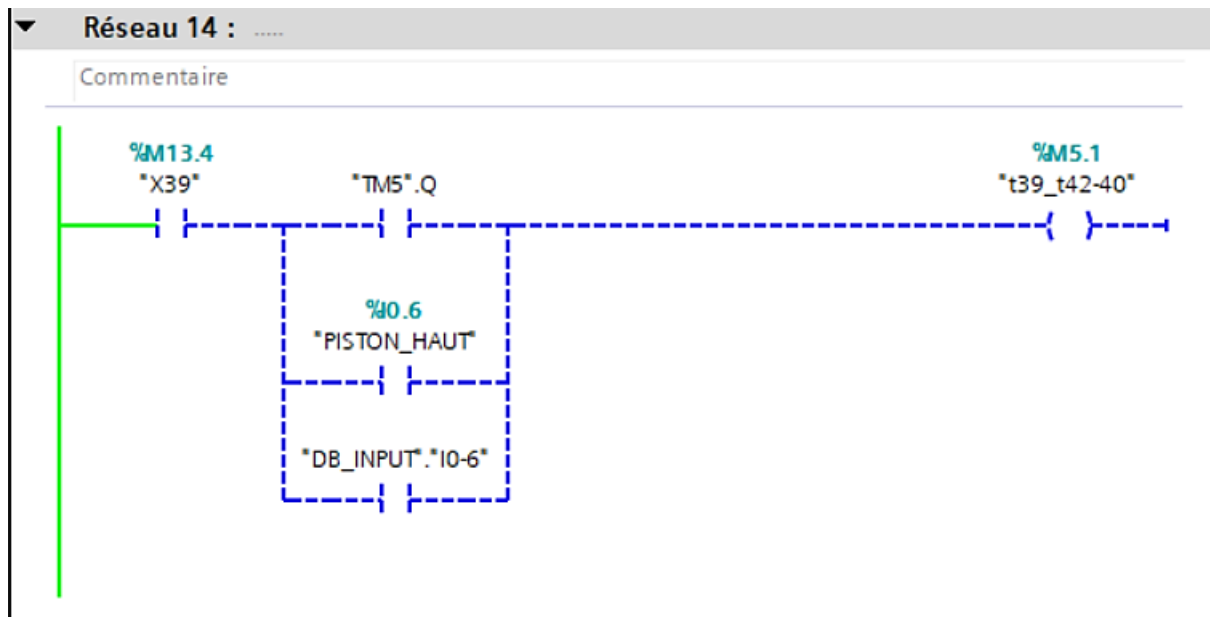


Figure III. 23 :La transition t39_42-40

Dans le réseau 14 on a préféré d'utiliser la notation "t39_t42-40" au lieu d'introduire deux réseaux (deux transitions "t39_40" et "t39_42"), car ces deux transitions ont la même réceptivité.

Pour voir la totalité des transition de ce GRAFCET voir l'annexe.

Les étapes :

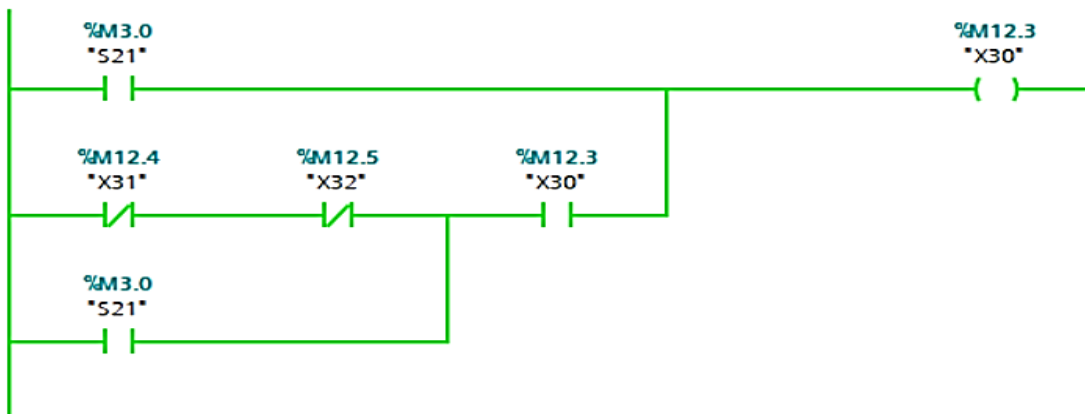


Figure III. 24 : L'étape initiale x30 de GRAFCET "FORMATION_CAISSE"

Avant de l'appui sur le bouton "Marche_Générale", l'étape initial est active car le bit d'initialisation "S21" est à 1, à partir du réseau 18 en remarque que les CAx30 et CDx30 sont satisfaites simultanément, le résultat est l'activation de x30 (Activation ET désactivation = Activation)

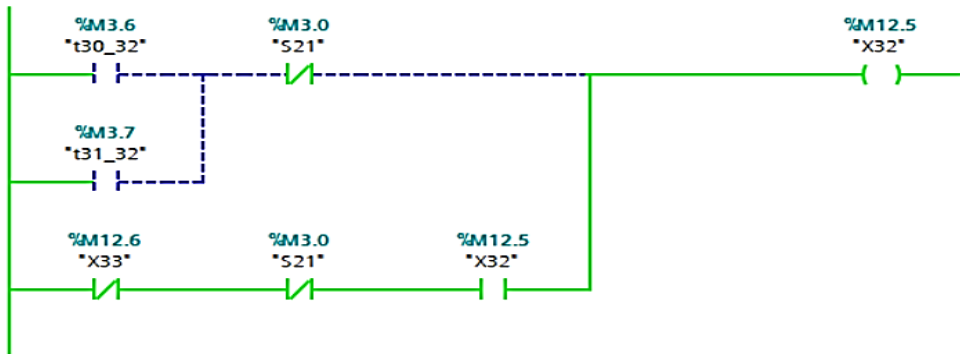


Figure III. 25 : L'étape x32

A partir du réseau 20, on peut remarquer que la convergence en "OU" se trouve dans les conditions d'activation de l'étape 32.

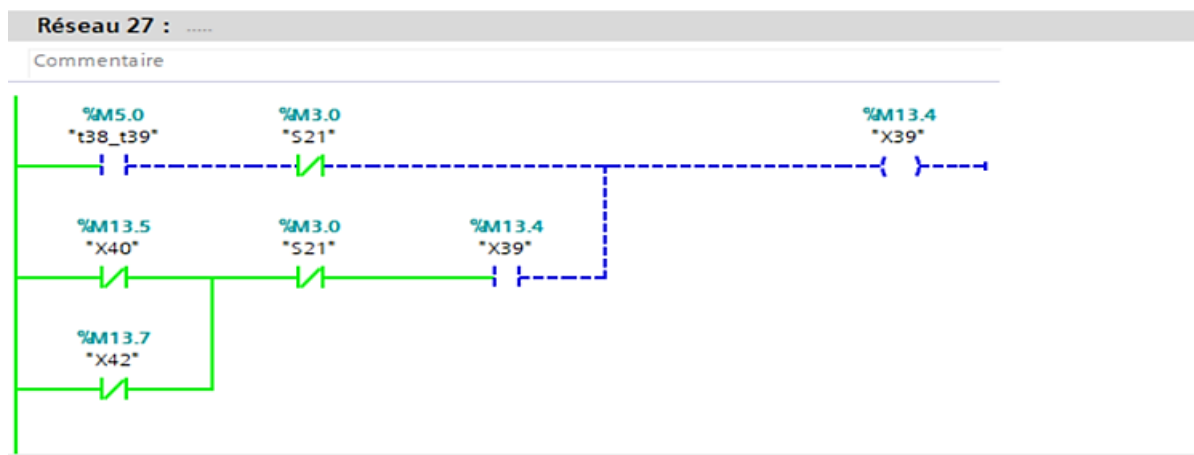


Figure III. 26 : L'étape x39

A partir de réseau 27, on peut remarquer que la divergence en "ET" se trouve dans les conditions de désactivation de l'étape 39.

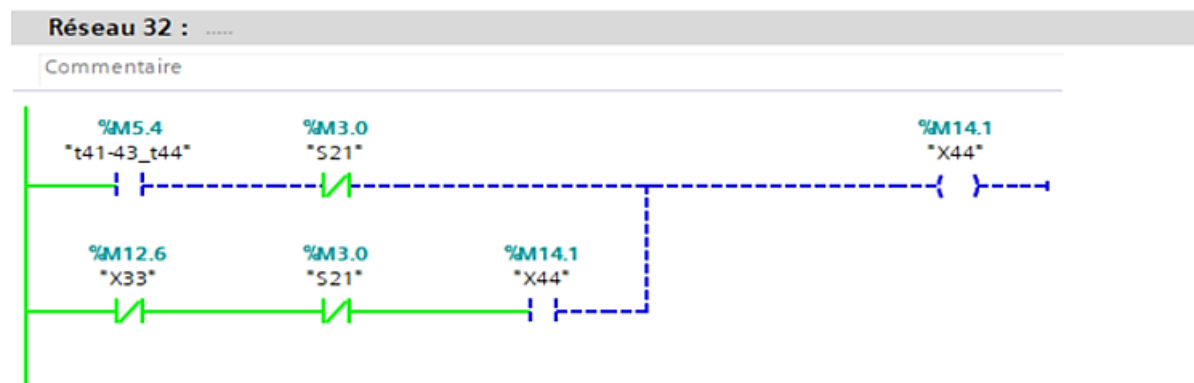


Figure III. 27 : L'étape x44

A partir de réseau 32, on peut remarquer que la convergence en "ET" se trouve dans les conditions de d'activation de l'étape 44.

Pour voir la totalité des étapes de ce GRAFCET voir l'annexe.

- FORCAGE_PISTON[FC2]

Les transitions :

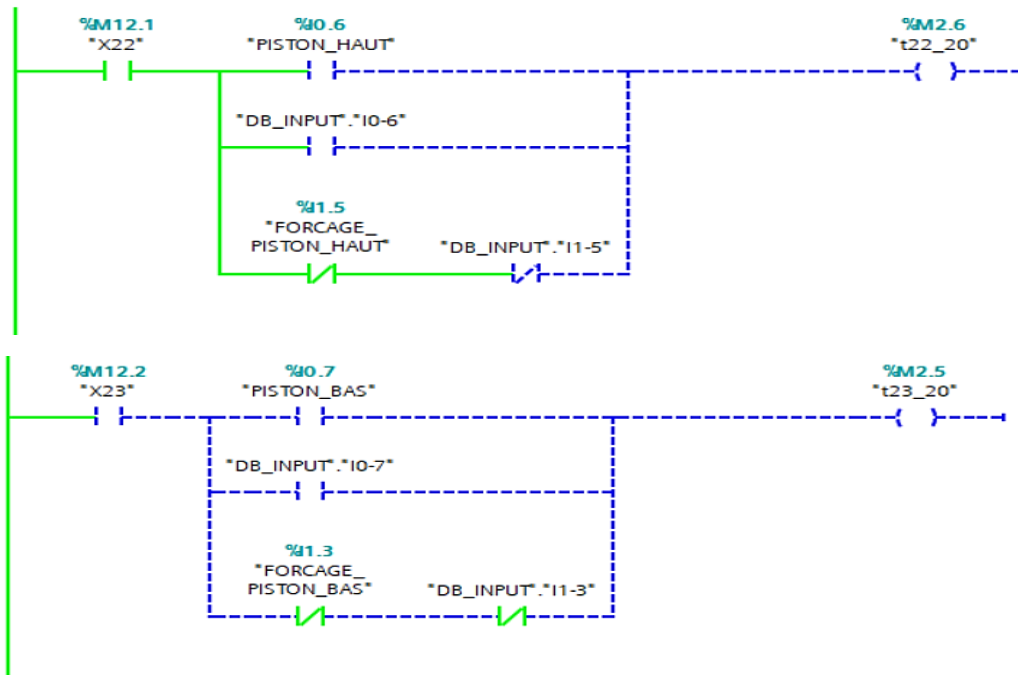


Figure III. 28 : Les transitions t22_20 et t23_20

A partir des deux transitions "t22_20" et "t23_20" de la figure III.28 on peut conclure qu'il y a une convergence en "OU".

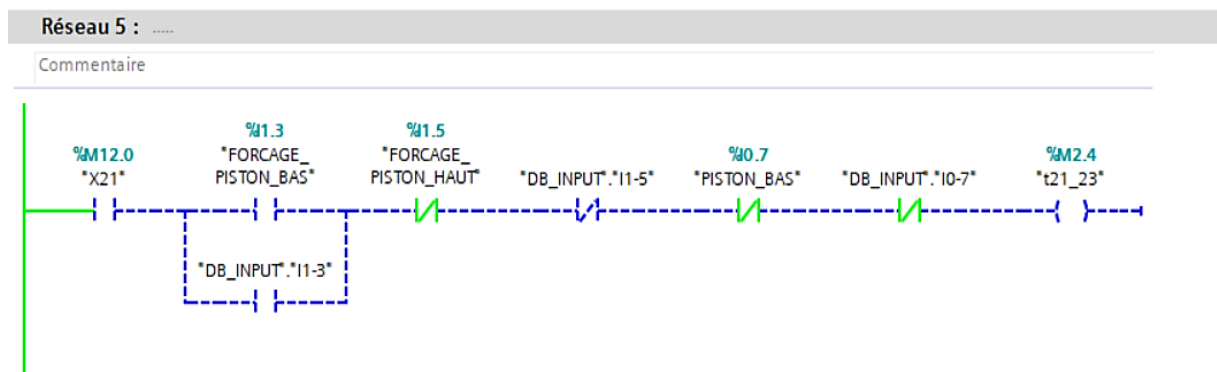
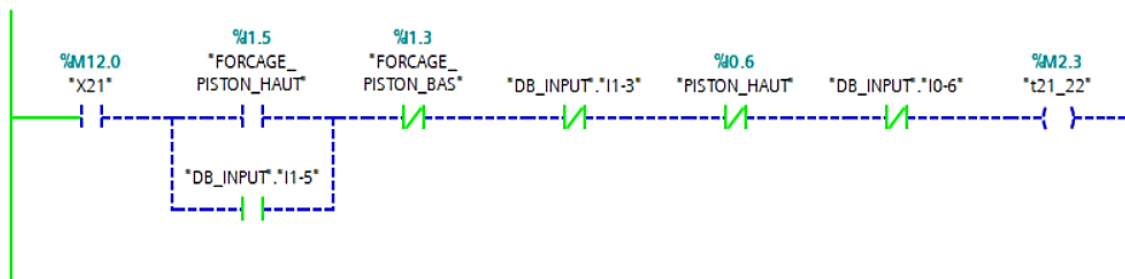


Figure III. 29 : Les t21_23 et t21_23

A partir des deux transitions "t21_23" et "t21_22" de la figure III.29 on peut conclure qu'il y a une divergence en "OU".

Pour voir la totalité des transition de ce GRAFCET voir l'annexe.

Les étapes :

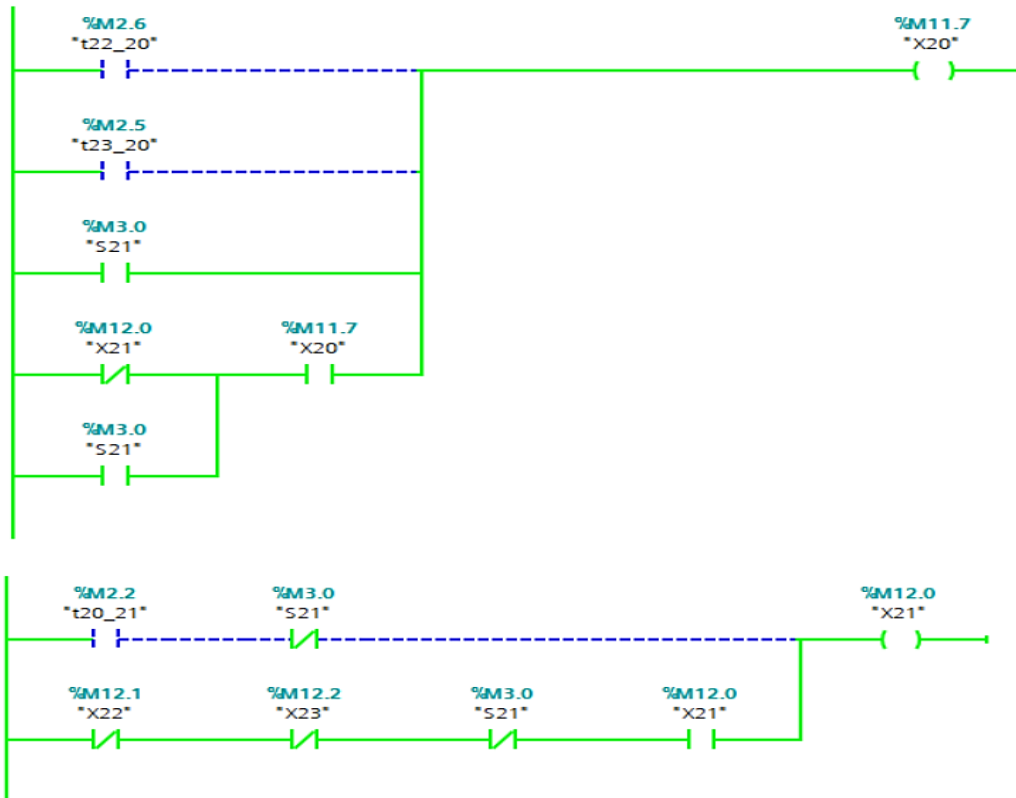


Figure III. 30 : L'étape initiale x20 et l'étape x21 de GRAFCET "FORCAGE_PISTON"

A partir de la figure III.30 on peut remarquer qu'il y a une convergence au niveau de l'étape 20, et une divergence se trouve dans les conditions de désactivation de l'étape 21.

- SORTIES[FC4] (les actions des grafcet)

On peut associer une ou plusieurs actions à chaque étape. Ces actions sont réalisées à chaque fois que nous activons l'étape à laquelle elles sont associées.

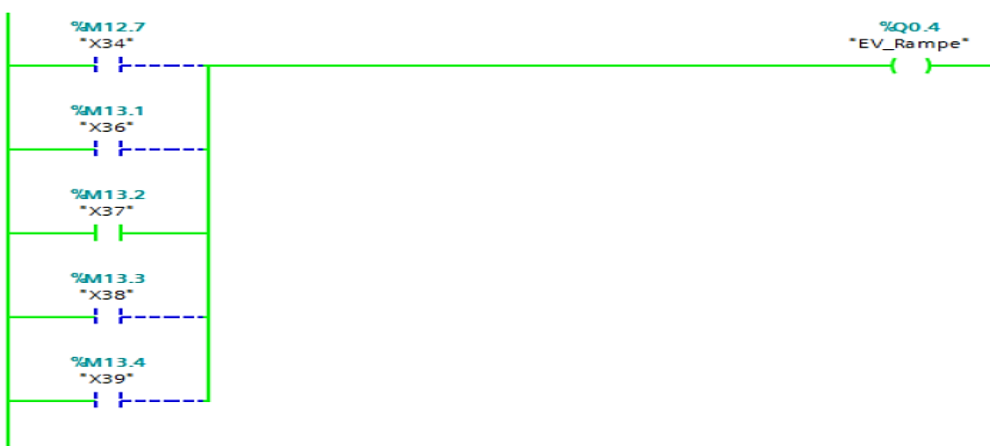


Figure III. 31 : Exemple d'un réseau Ladder d'une sortie

Pour voir la totalité des équations des sorties voir l'annexe.

- ALARMES[FC5]

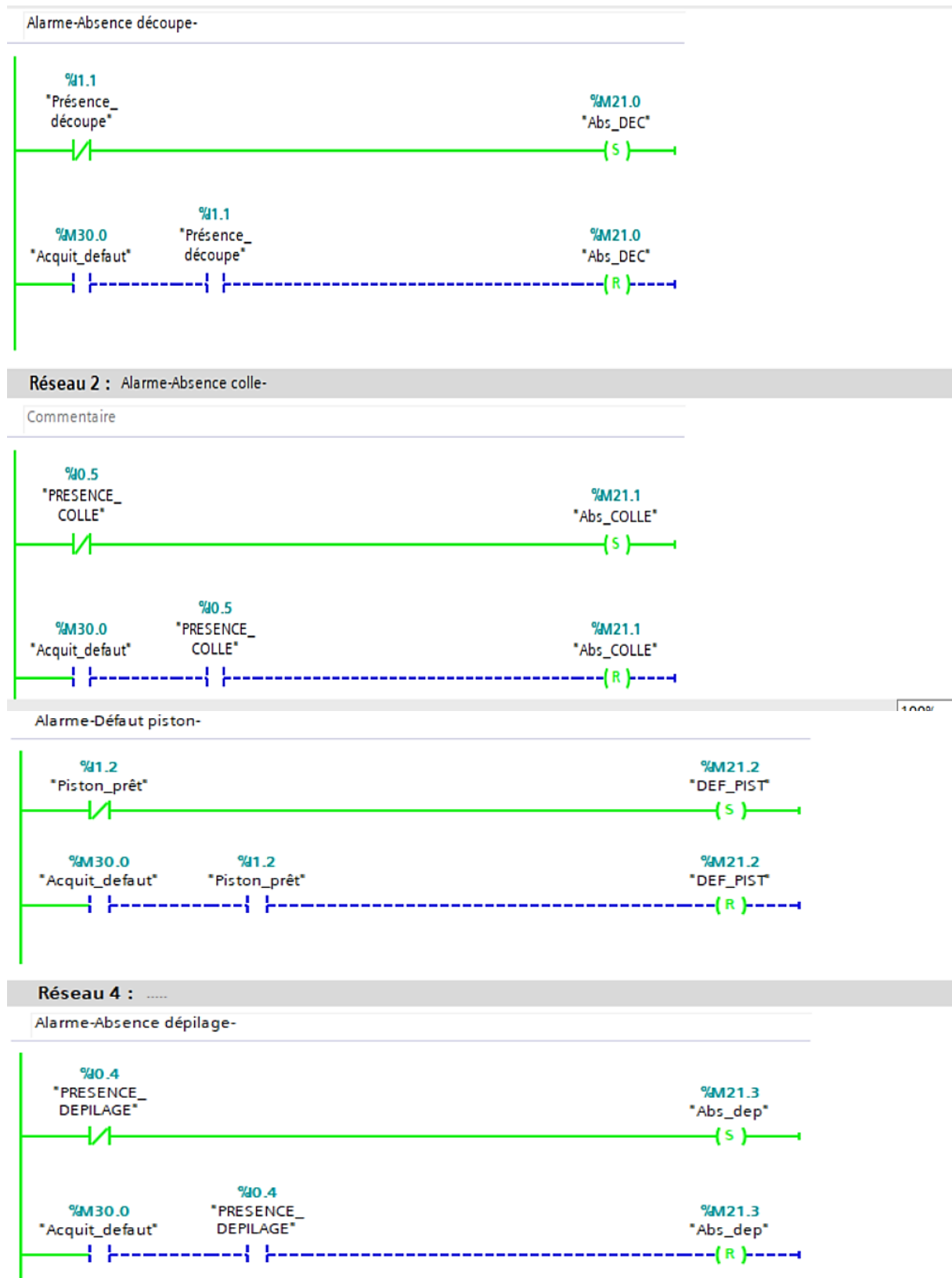


Figure III. 32 : Configuration des alarmes TOR

Quand une fonction de la machine échue une alarme se déclenche et persiste (car la sortie est mise à 1 grâce à un SET), en parallèle un deuxième réseau équipé d'un bouton d'acquiescement de défaut « Acquit_defaut » est ajouté à chaque alarme, si l'opérateur intervient et acquit l'alarme, cette dernière s'éteint mais le message reste affiché jusqu'au réglage du problème.

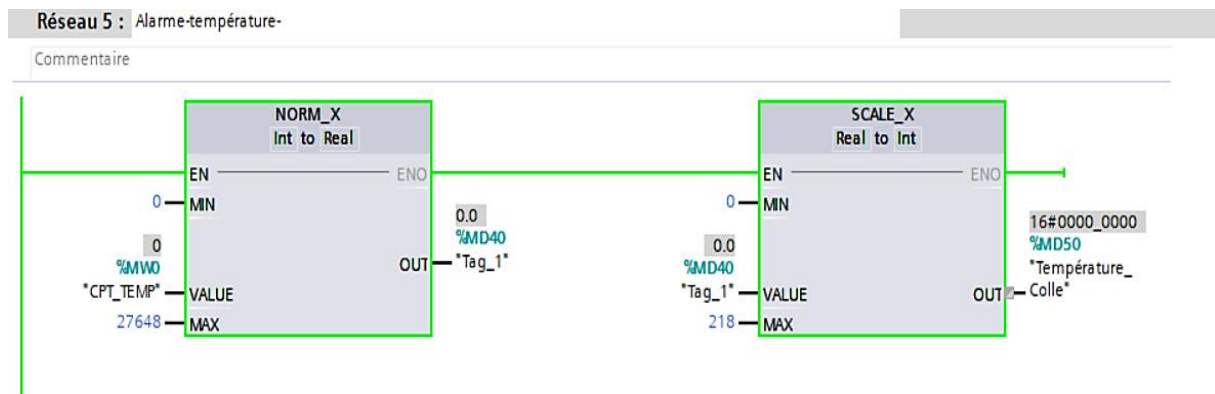


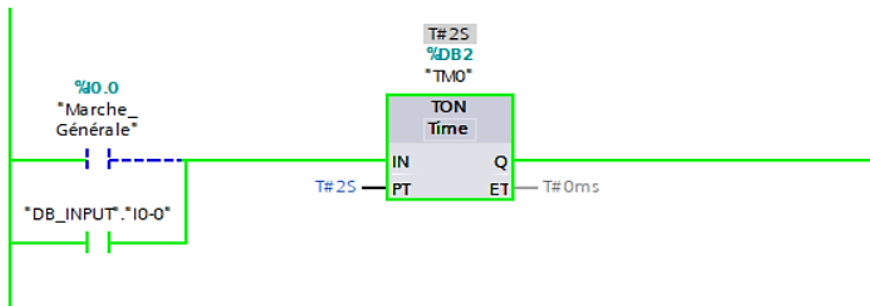
Figure III. 33 : Alarme analogique

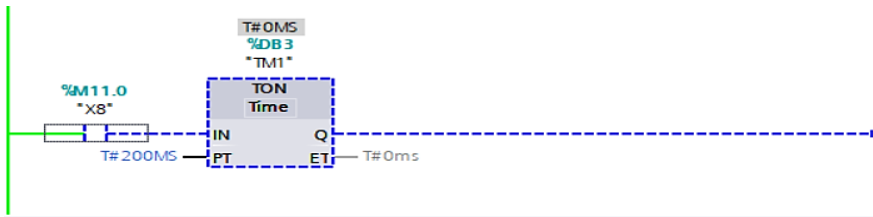
• **Les blocs fonctionnels – FB**

Dans notre projet on a utilisé des temporisateurs de type TON qui signifie Temporisateur à retard ON. L'utilisation du TON permet de produire un signal de sortie temporisé après un délai déterminé à partir de l'activation de l'entrée.

- TM0 : Initialisation pour le marche générale
- TM1 : Stabilisation découpe avant transfert
- TM2 : contrôle piston arrive en bas
- TM3 : Equerrage
- TM4 : Pressage

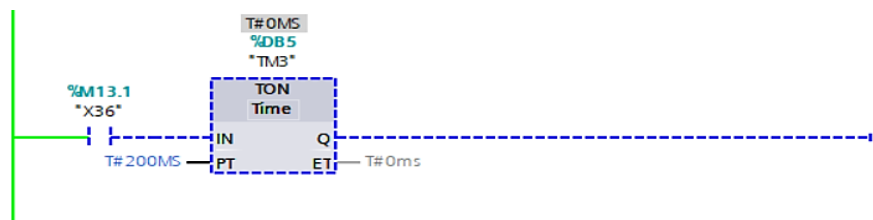
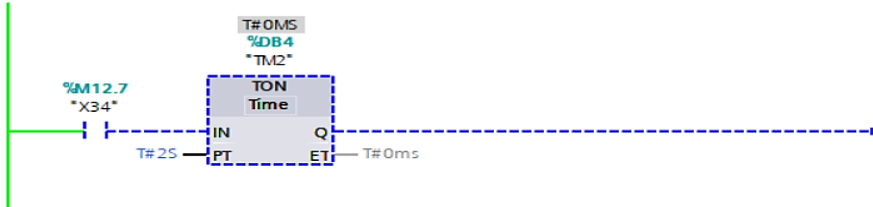
- TM5 : Anticipation transfert
- TM6 : Anticipation remonte vérin prise
- TM8 : Défaut prise découpe
- TM10 : Accumulation Convoyeur
- TM11 : Arrêt convoyeur





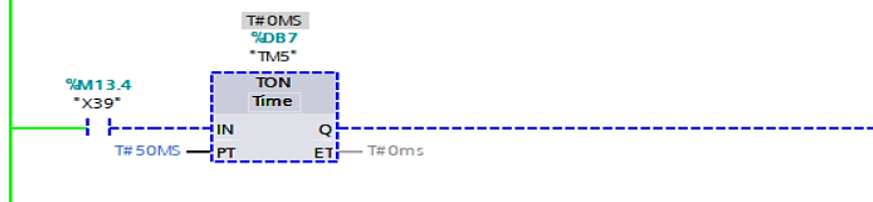
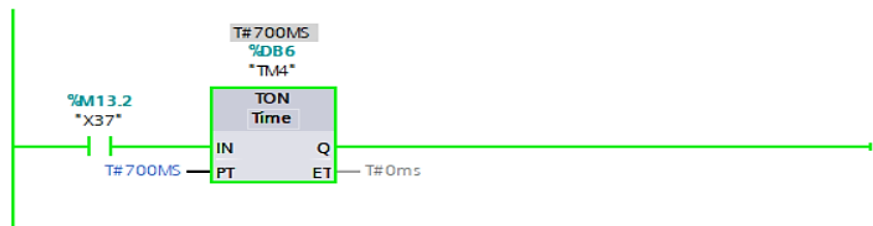
Réseau 3 : CONTROLE PISTON ARRIVE EN BAS

Commentaire



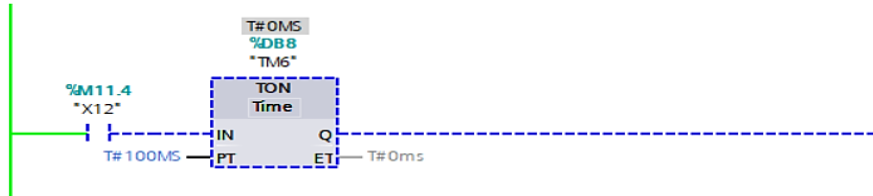
Réseau 5 : PRESSAGE

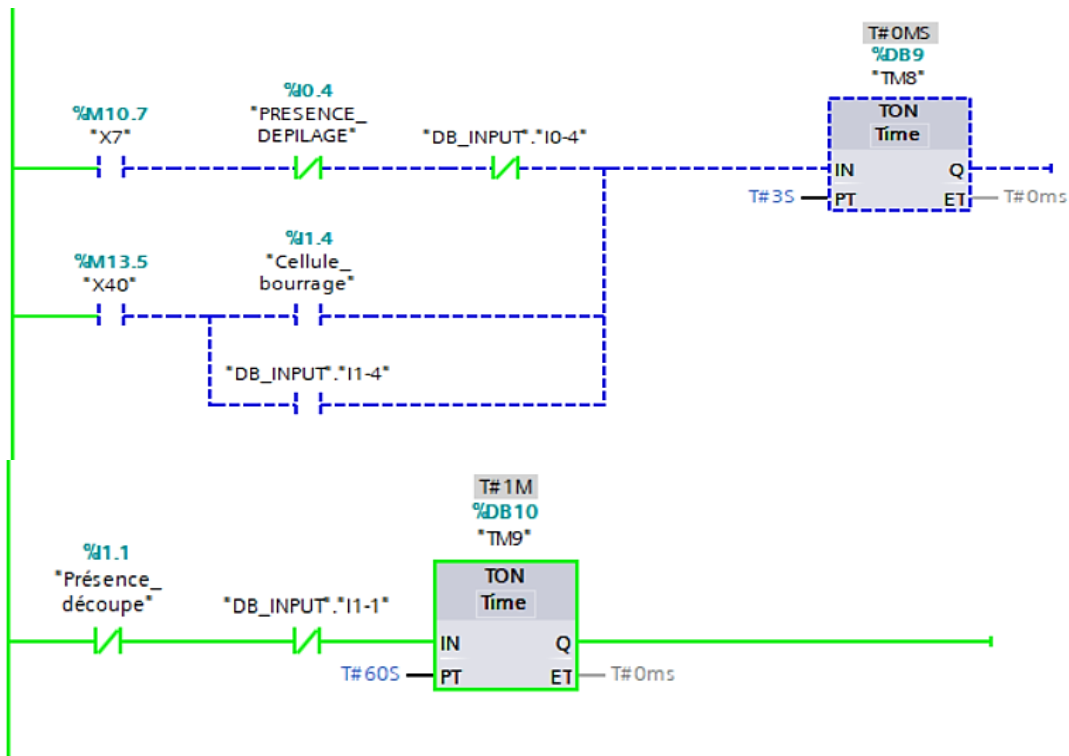
Commentaire



Réseau 7 : ANTICIPATION REMONTE VERIN PRISE

Commentaire





Réseau 10 : ACUMULATION CONVOYEUR

Commentaire

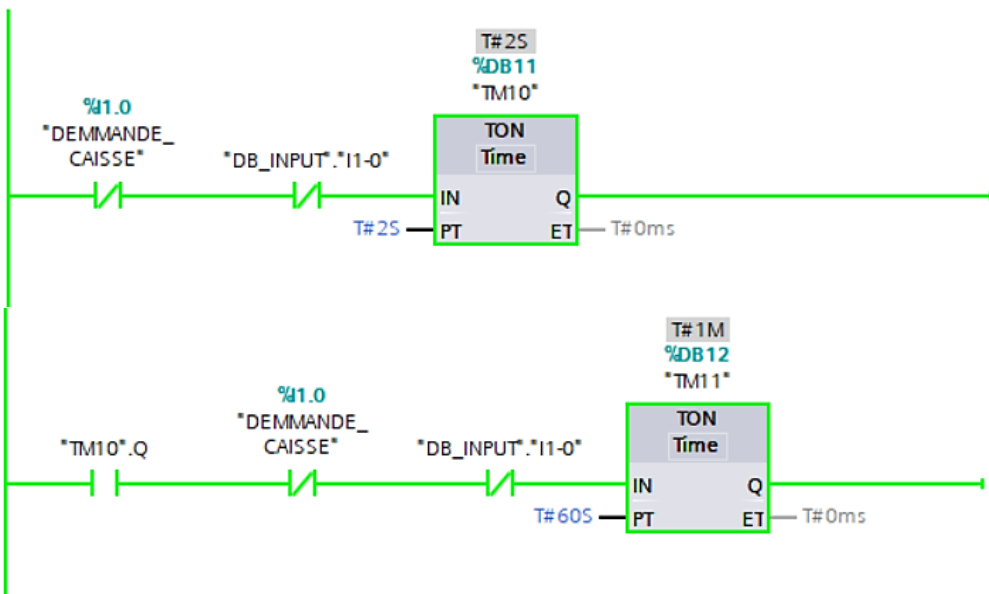


Figure III. 34 : Le bloc FB1 des temporisations utilisées

- **Le bloc d'organisation – OB1**

Dans notre projet pour exécuter les blocs qu'on a déjà déclarés on fait les appeler dans le bloc OB1 pour un traitement cyclique.

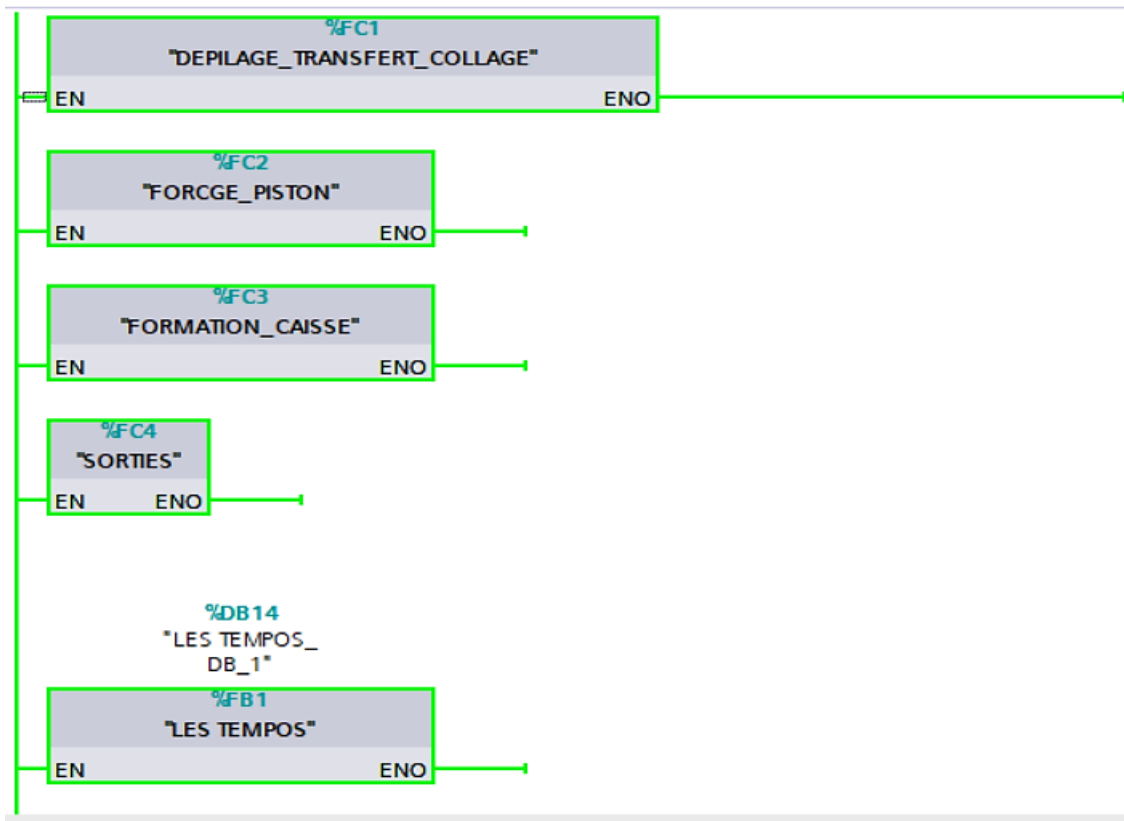


Figure III. 35 : Le bloc OB1

On a déclaré aussi dans ce bloc les réseaux ladder de la partie préliminaire, la figure suivante montre cette étape :

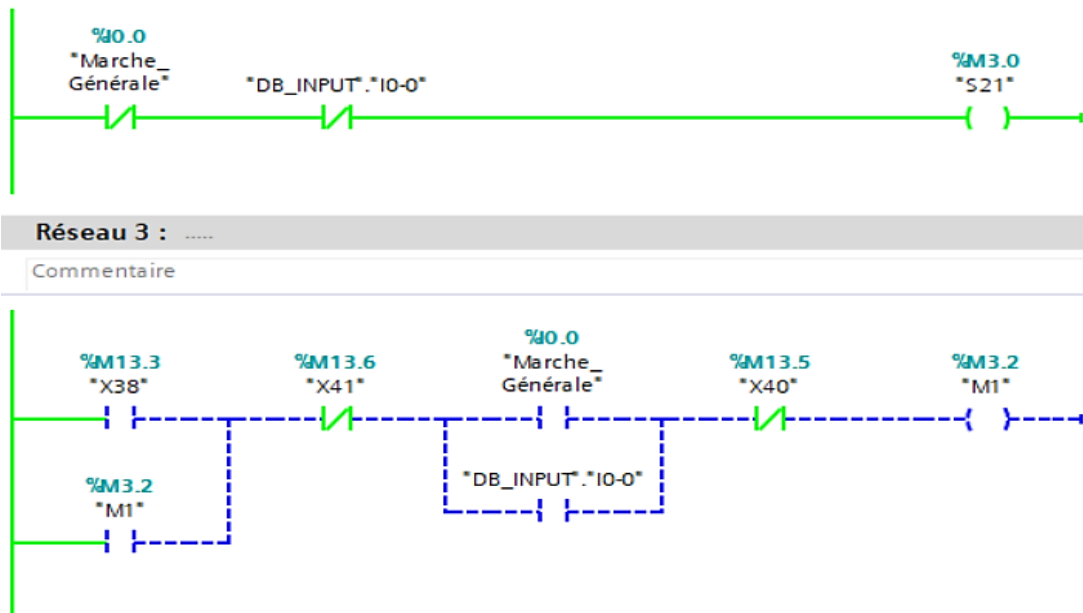
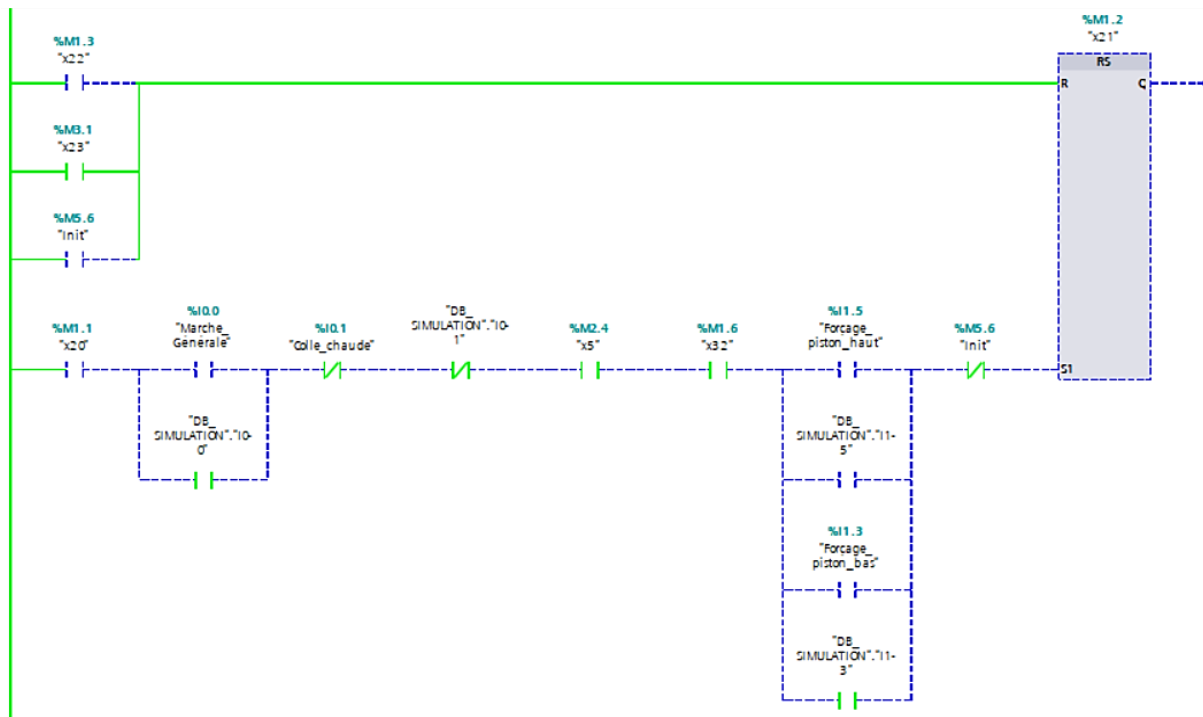
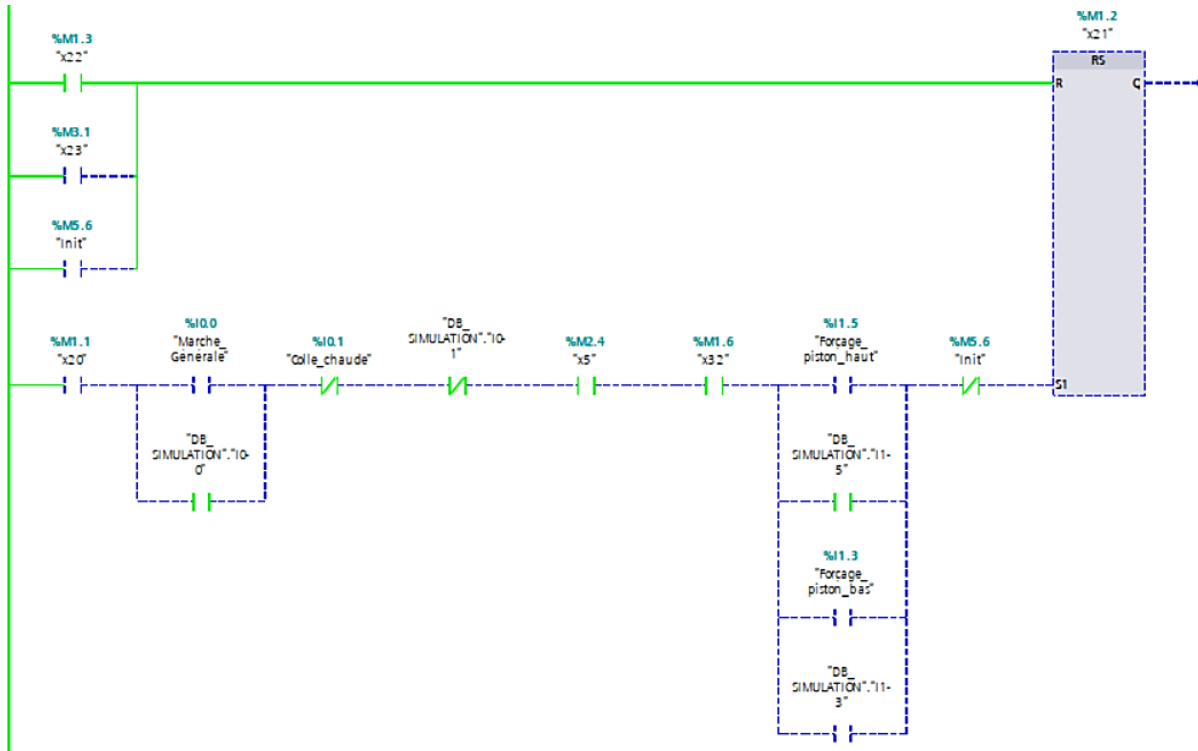
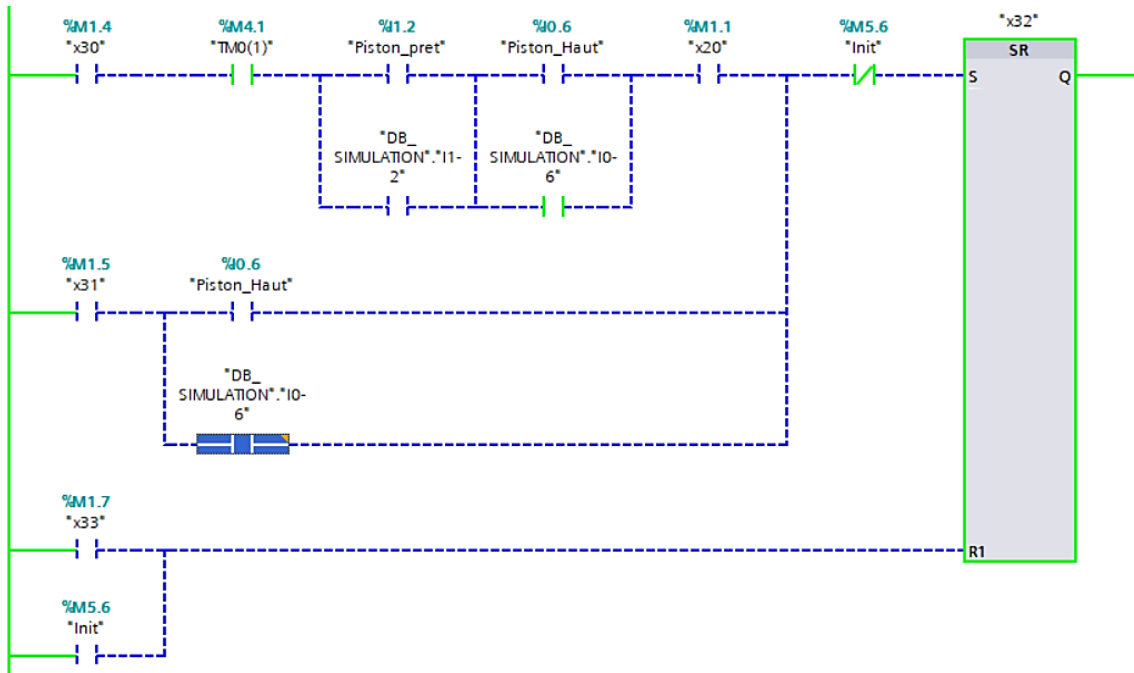


Figure III. 36 : La partie PRL

S21 : est un bit qu'est géré par l'utilisateur pour initialiser le GRAFCET (mise à 1 de préférence dans PRL). Il est repositionné à 0 par le système après initialisation du GRAFCET.

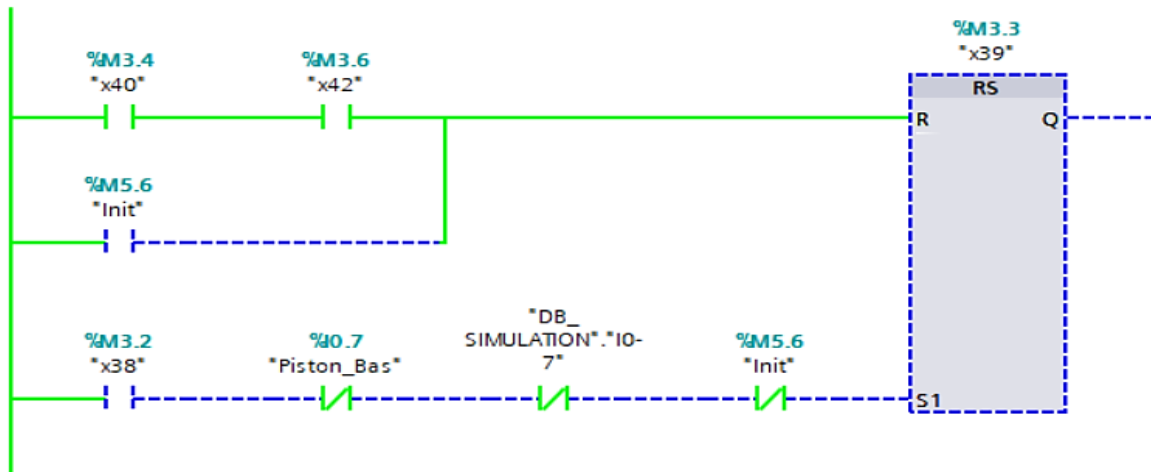
❖ **Solution 2 :** On a suivi le même raisonnement précédent tel que on considère chaque étape comme est représentée comme une mémoire interne, mais cette fois on utilise des bascules bascule RS qui nous a permet de faire facilement SET (activation) et RESET (désactivation) des étapes :





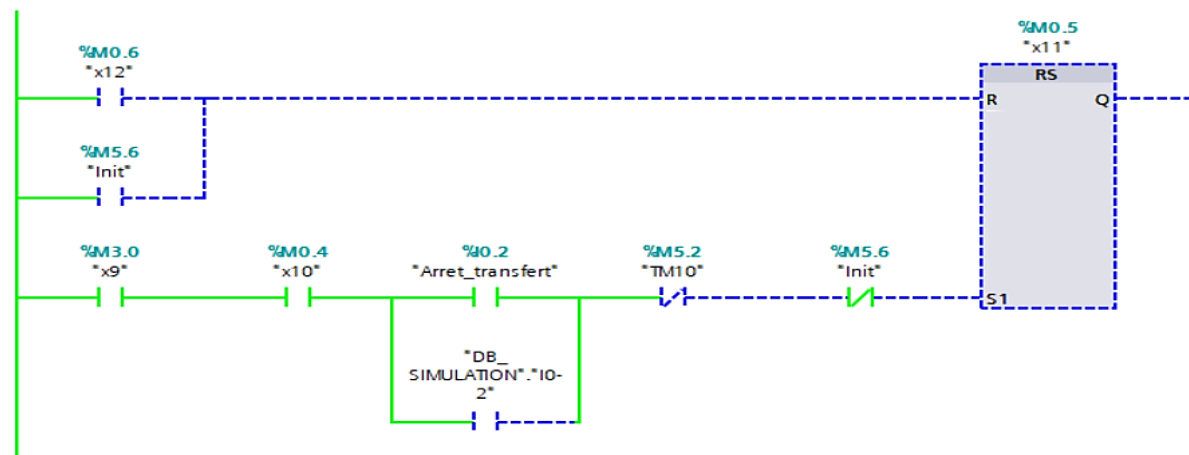
Réseau 10 : Etape 39. FORMATION_CAISSE. Solution 1

Commentaire



Réseau 12 : Etape 11. Dépilage_Transfert_Collage. Solution 1

Commentaire



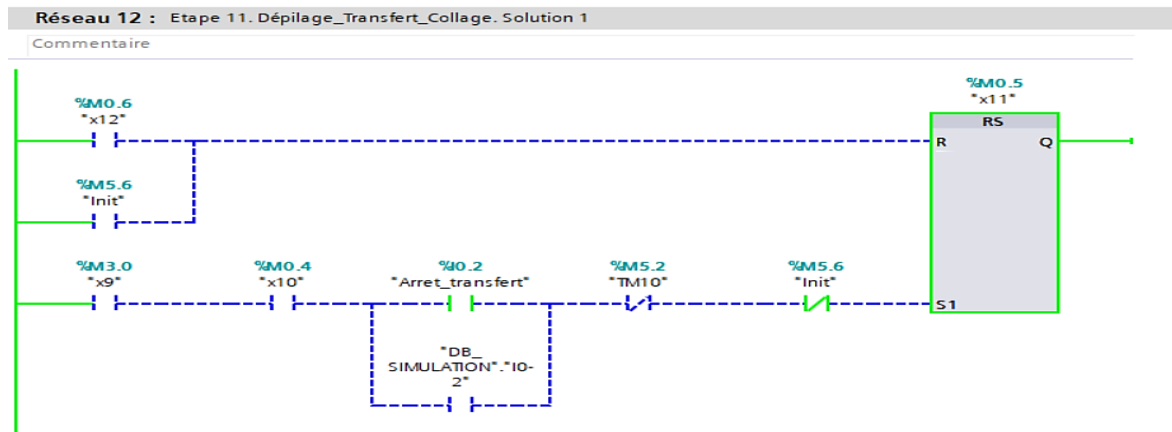


Figure III. 37 : Quelques exemples des étapes

Remarques :

- Une étape est une mémoire binaire.
- Les bascules RS permettent de répondre aux règles qui définissent le GRAFCET.
- L'utilisation des blocs RS (et non SR) répond à la règle 4.
- La divergence en OU se retrouve dans le Reset de l'étape 21.
- La convergence en OU se retrouve dans le Set de l'étape 32.
- La divergence en ET se retrouve dans le Reset de l'étape 39.
- La convergence en ET se retrouve dans le Set de l'étape 11.

❖ **Solution 3** : Avec le TIA Portale c'est possible aussi d'utiliser une seule variable interne (Etape numéro) type entier %MW qui prend le numéro d'étape active : si Etape numéro = i donc l'étape i est active. Comme on a trois GRAFCETs on a créé trois mots % MW0 "DEP_TRANS_COLL", %MW20 "FOR_PIST" et %MW30 "FORM_CAISSE".

48	DEP_TRANS_COLL	WORDS	Word	%MW0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
50	FOR_PIST	WORDS	Word	%MW20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
51	FORM_CAISSE	WORDS	Word	%MW30	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure III. 38 : La table des mots déclarés

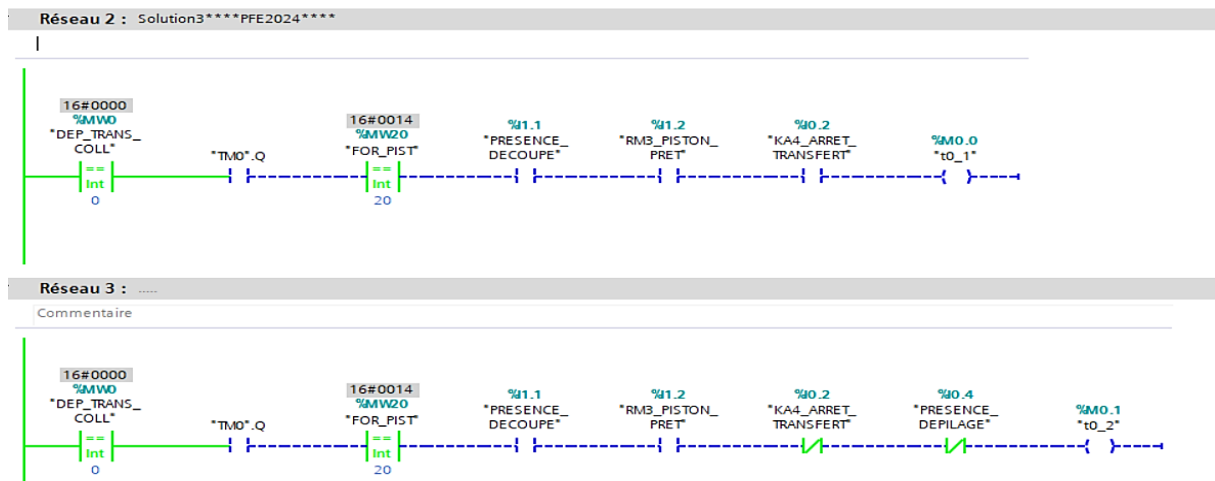


Figure III. 39 : Exemples des transitions

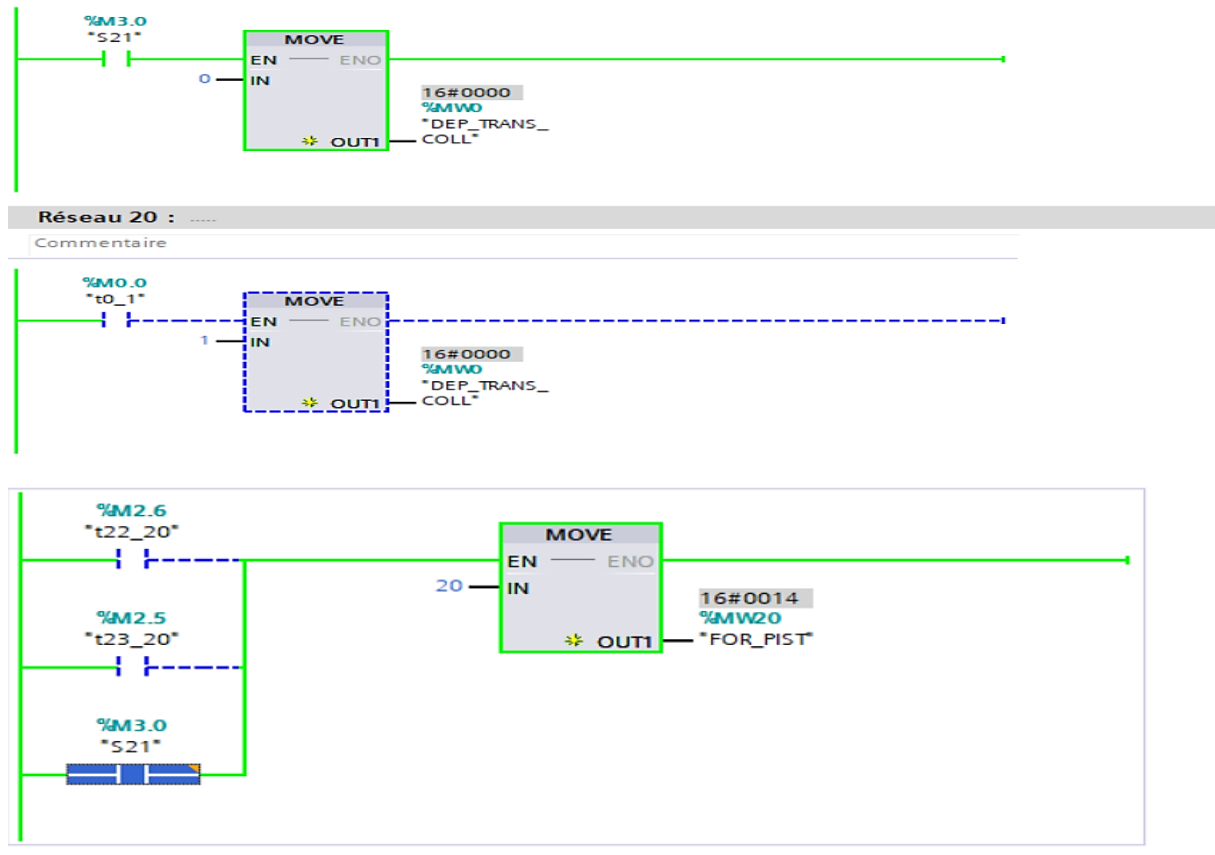
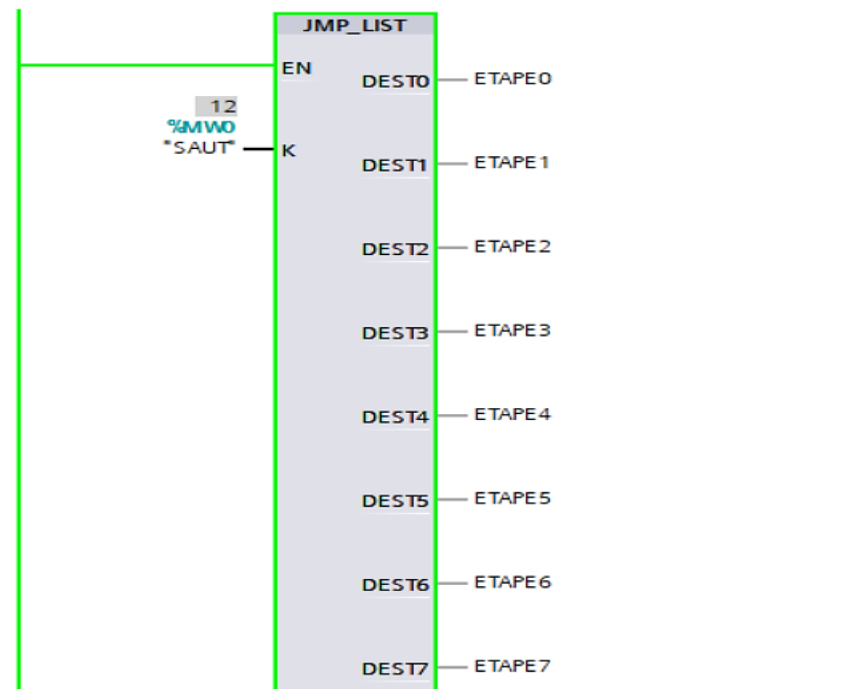


Figure III. 40 : Exemples des étapes

- ❖ **Solution 4 :** dans cette étape on garde le même raisonnement précédent, mais cette fois on ajoute des instructions de saut et des étiquettes, cette méthode nous a permet de réduire le temps de cycle, puisque le simulateur traite directement l'étape concernée.



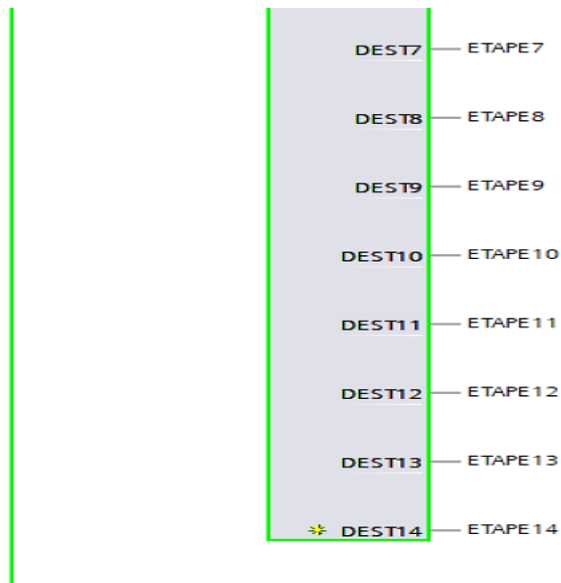
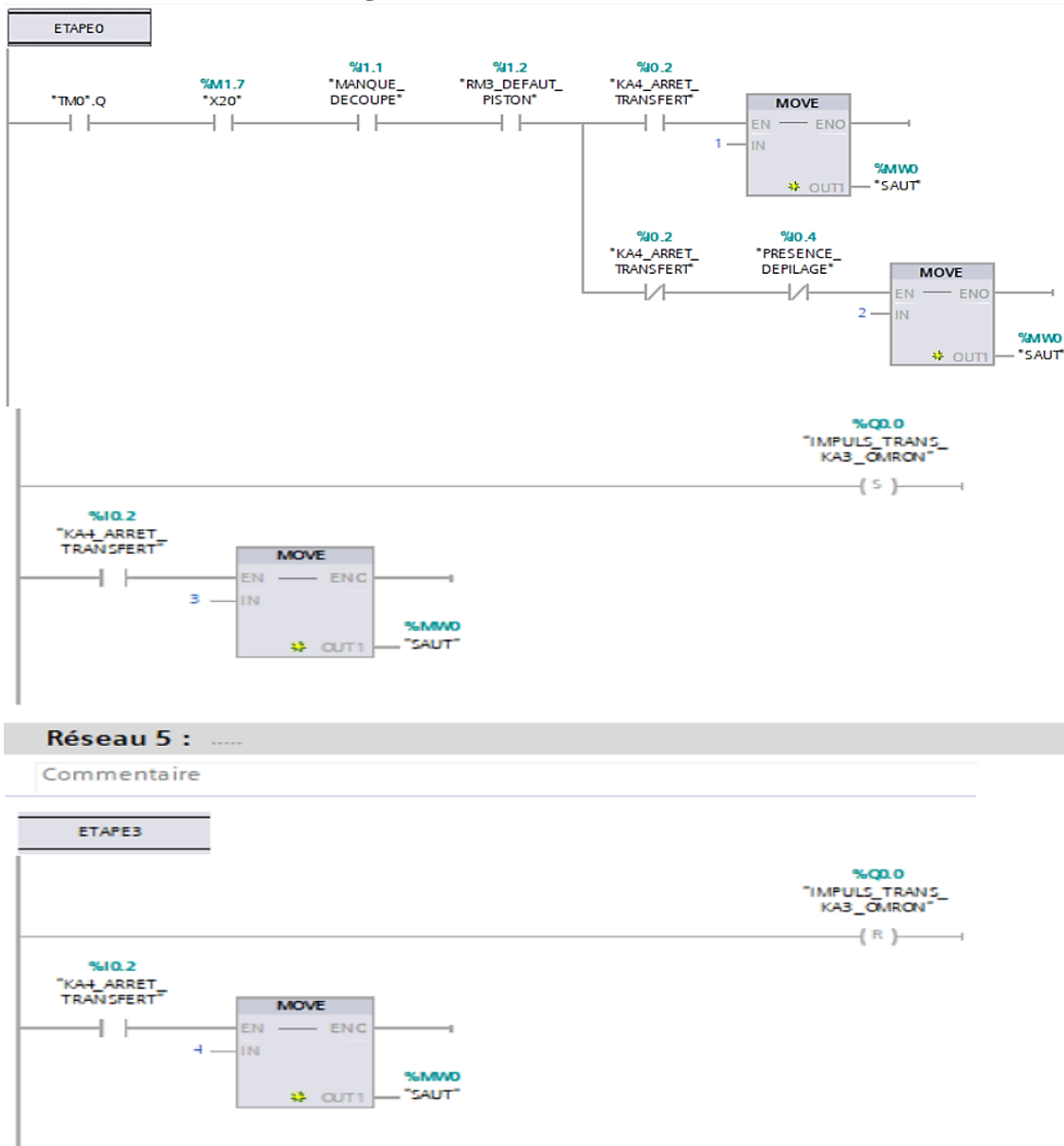


Figure III. 41 : Le bloc « JMP_LIST »



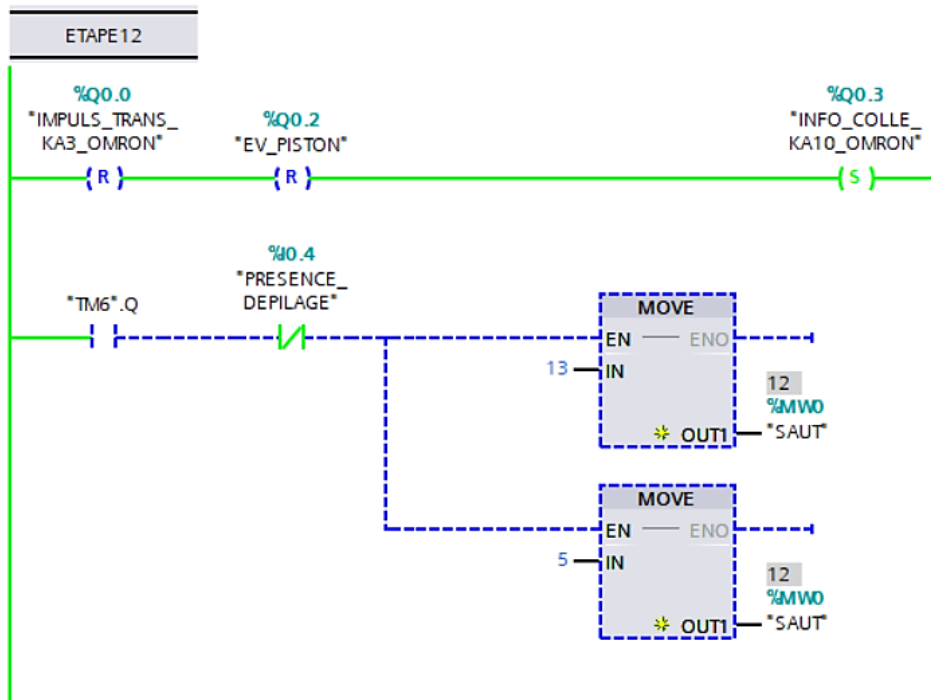


Figure III. 42 : Exemples des étapes

III.7 Conclusion

Dans ce chapitre après avoir rédigé le cahier de charges de la machine et le traduire en GRAFCET, on a justifié la migration de l'automate Scheider TSX Micro vers l'automate SIEMENS S7 1200 et on a vu son importance.

Avant d'entamer la partie programmation, nous avons vu les critères de choix de la CPU, ainsi que la configuration matérielle sur TIA PORTAL, ensuite nous avons traité quatre solutions possibles pour traduire nos GRAFCETs en langage LADDER vu que notre automate SIEMENS S7 1200 ne prend pas en charge le GRAFCET comme un langage de programmation, lors de la simulation nous avons rencontré des difficultés de suivre l'évolution des étapes du GRAFCET à cause de la complexité du programme d'une part, et des temporisations courtes.

Face à ces contraintes nous proposons de visualiser notre GRAFCET dans une interface IHM.

Chapitre IV : Interfaces de supervision

IV.1 Introduction

Dans le cas où les machines et les installations doivent répondre à des exigences de fonctionnalité de plus en plus rigoureuses, il est crucial de garantir à l'opérateur un niveau élevé de confiance. On obtient cette transparence en utilisant une Interface Homme-Machine (IHM). Une interface utilisateur (IHM) est un ensemble de vues préalablement générées et configurées à l'aide d'un logiciel adéquat, qui sont ensuite exposées à l'utilisateur.

Au cours de ce chapitre, on va créer une plateforme de supervision en utilisant le logiciel WinCC de TIA PORTAL. Cette plateforme permettra de visualiser le fonctionnement de la barquetteuse et l'évolution des trois GRAFCETs, ainsi détecter d'éventuels défauts survenant pendant son fonctionnement.

IV.2 Configuration de l'IHM

Dans le chapitre II on a vu la configuration matérielle et la création d'une vue IHM, il reste à vérifier que l'adresse Ethernet de l'IHM ajouté dans notre programme, est compatible avec celle de l'interface actuelle qu'on va utiliser en temps réel, cela peut se faire à partir de la fenêtre des propriétés de l'IHM, qu'on peut sélectionner dans "Appareils et réseaux", dans le navigateur du projet, comme indique la figure IV.1.

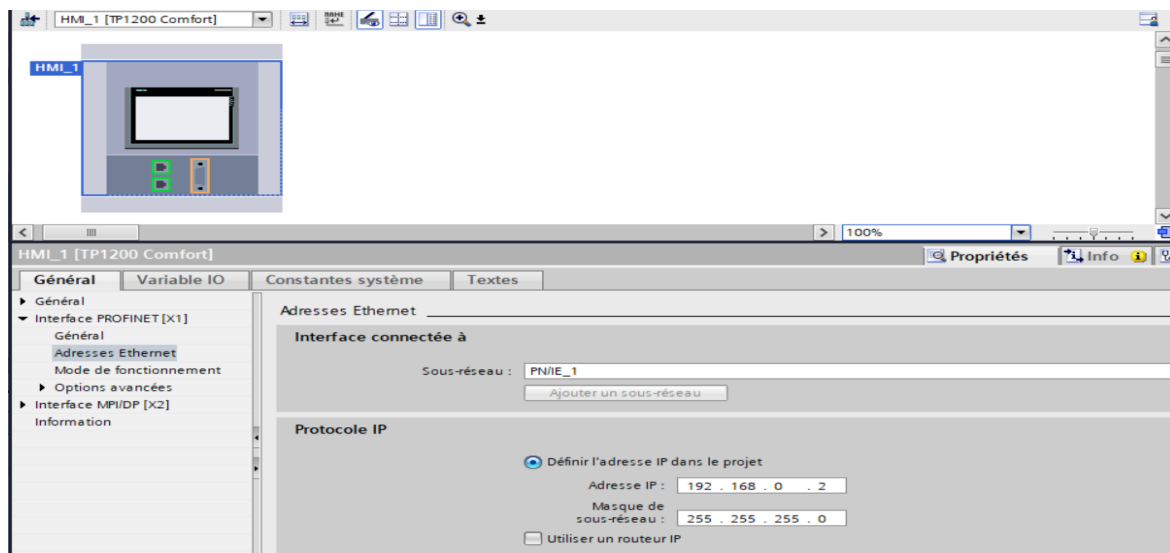


Figure IV. 1 : Adresse Ethernet

IV.3 Création d'un nouveau modèle

Dans notre projet on a décidé de créer un nouveau modèle, pour le créer (1) on clique sur le dossier « HMI_1[TP1200 Comfort] », ensuite (2) on choisit le dossier « Gestion des vues » et (3) on clique sur « Modèles » après (4) on clique sur « Ajouter un nouveau modèle », la figure IV.2 montre ces étapes.

Pour créer ce nouveau modèle on utilise les outils intégrés dans WINCC de TIA PORTAL, et on ajoute des outils graphiques, (1) on clique sur le dossier « Langages & Ressources », (2) ensuite on clique sur « Graphics », (3) l'espace d'ajouter des Graphics s'ouvre, la figure IV.3 montre ces étapes.

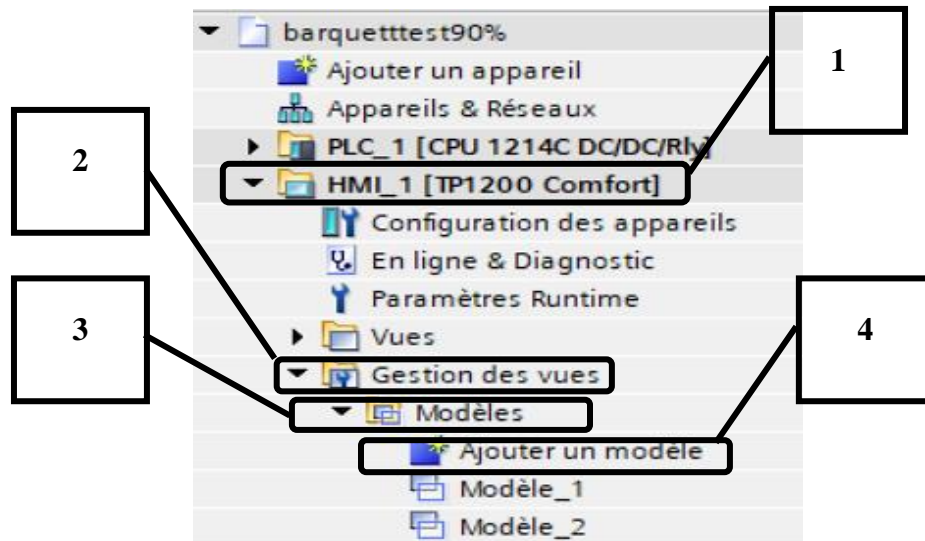


Figure IV. 2 : Création d'un nouveau modèle

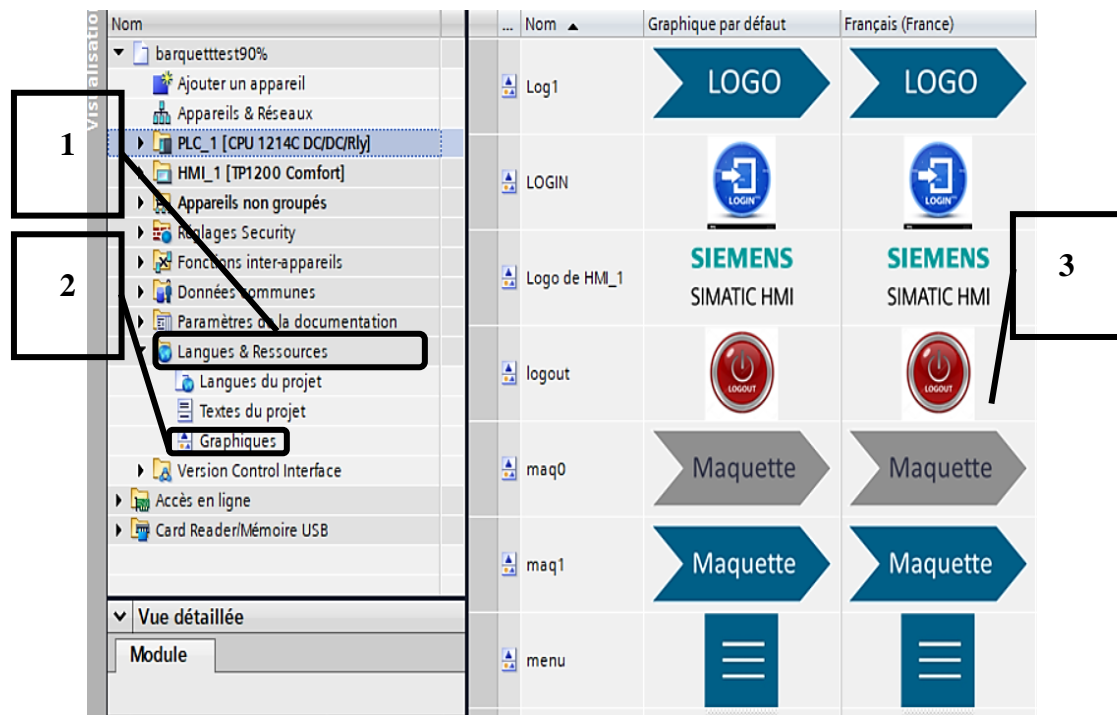


Figure IV. 3 : Eléments graphiques

La figure IV.4 montre le modèle final obtenu, ce modèle contient des icônes pour arrêter la simulation, gestion d'utilisateur, description & schéma électrique de la machine (sous forme d'un document pdf, cette vue contient tous les boutons nécessaires pour un pdf tel que le saut

de page, zoom in, zoom out...), ainsi une icône pour accéder au menu, à partir de la vue de ce modèle on peut aussi accéder à la vue d'accueil, la machine avec ces caractéristiques et au logo.

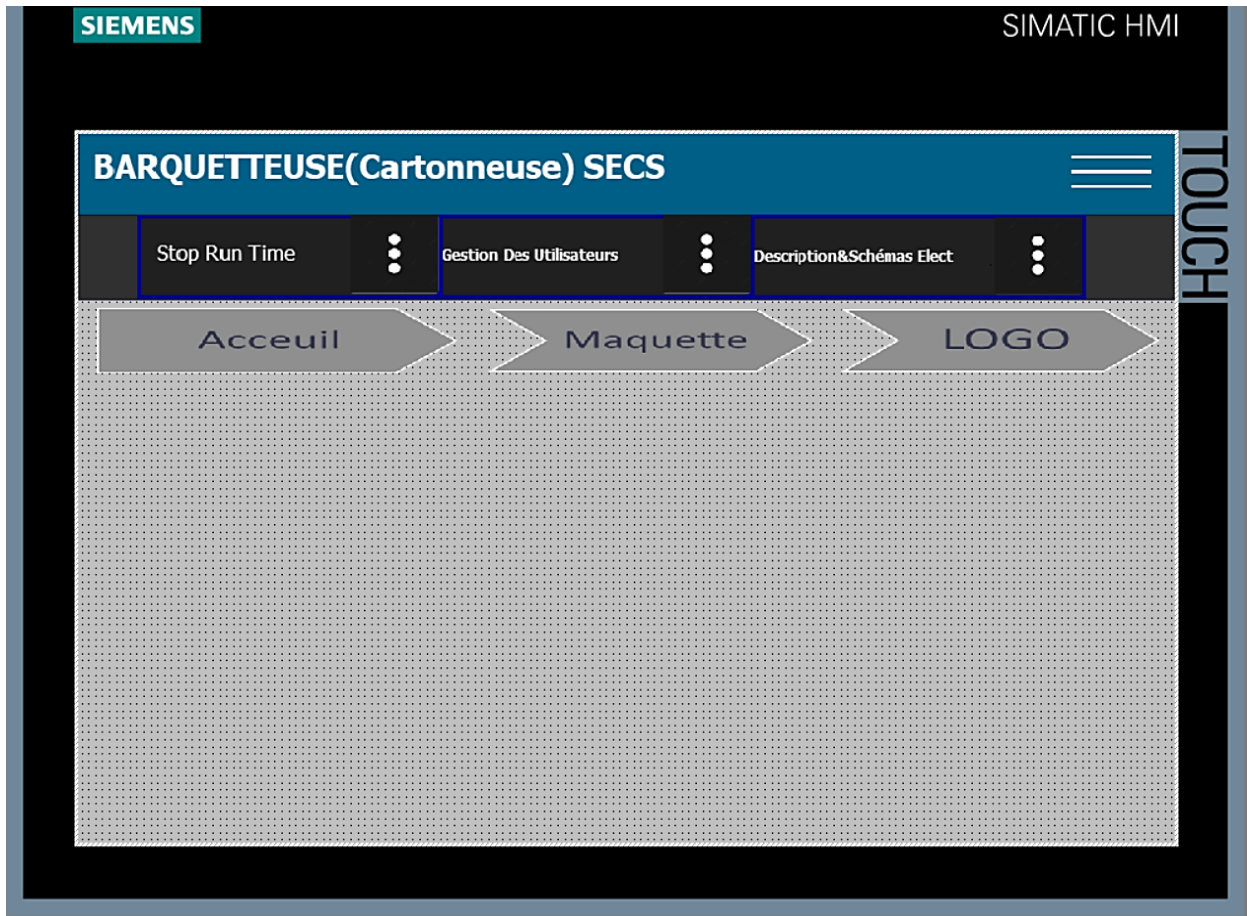


Figure IV. 4 : Modèle créé

Pour le bouton Menu on a utilisé (1) « Vues Slide », (2) exactement « Vues Slide-in droite », la figure suivante montre ces étapes :

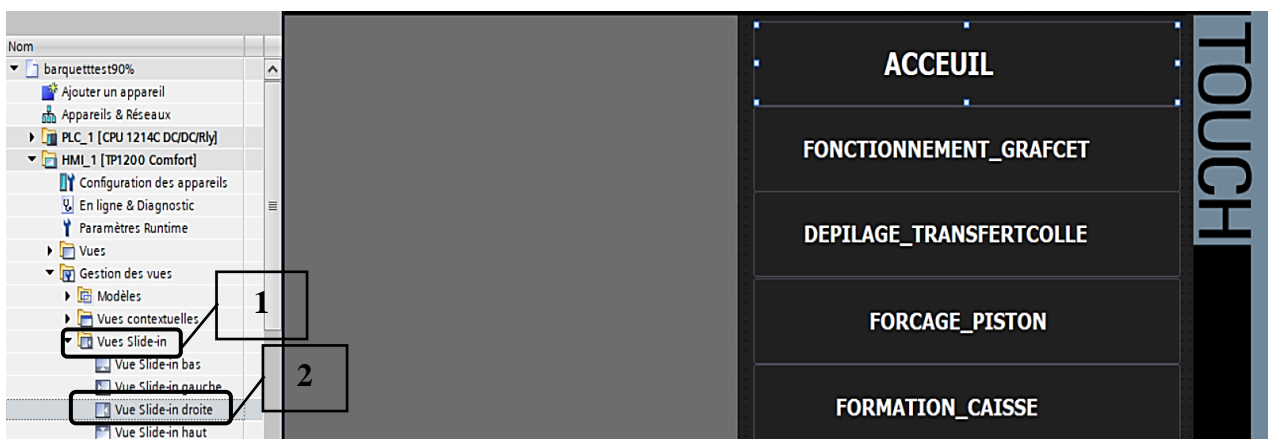


Figure IV. 5 : Vue Slide-in droite

IV.4 Vue d'accueil

C'est la première qui s'affiche, elle montre l'intitulé du projet, une photo réelle de la station ainsi qu'une petite description de cette dernière

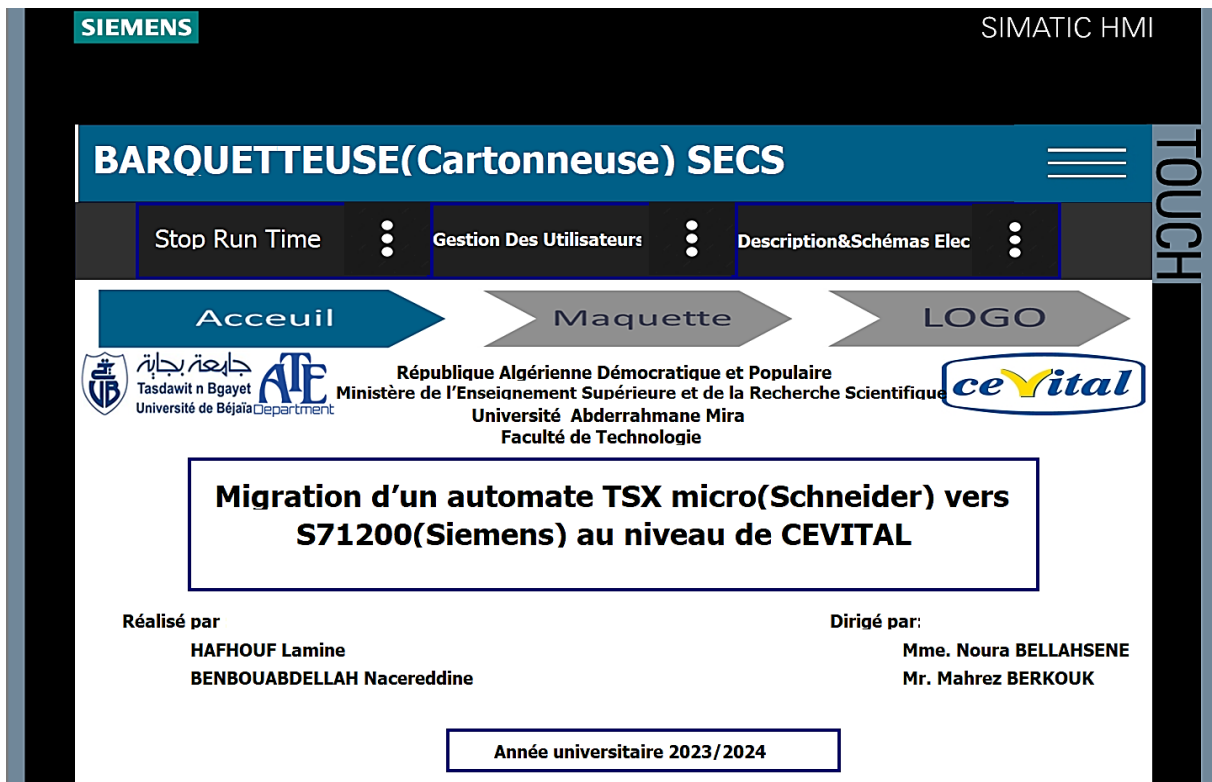


Figure IV. 6 : Vue d'accueil

- En appuyant sur le bouton “Maquette” on passe à la vue suivante.

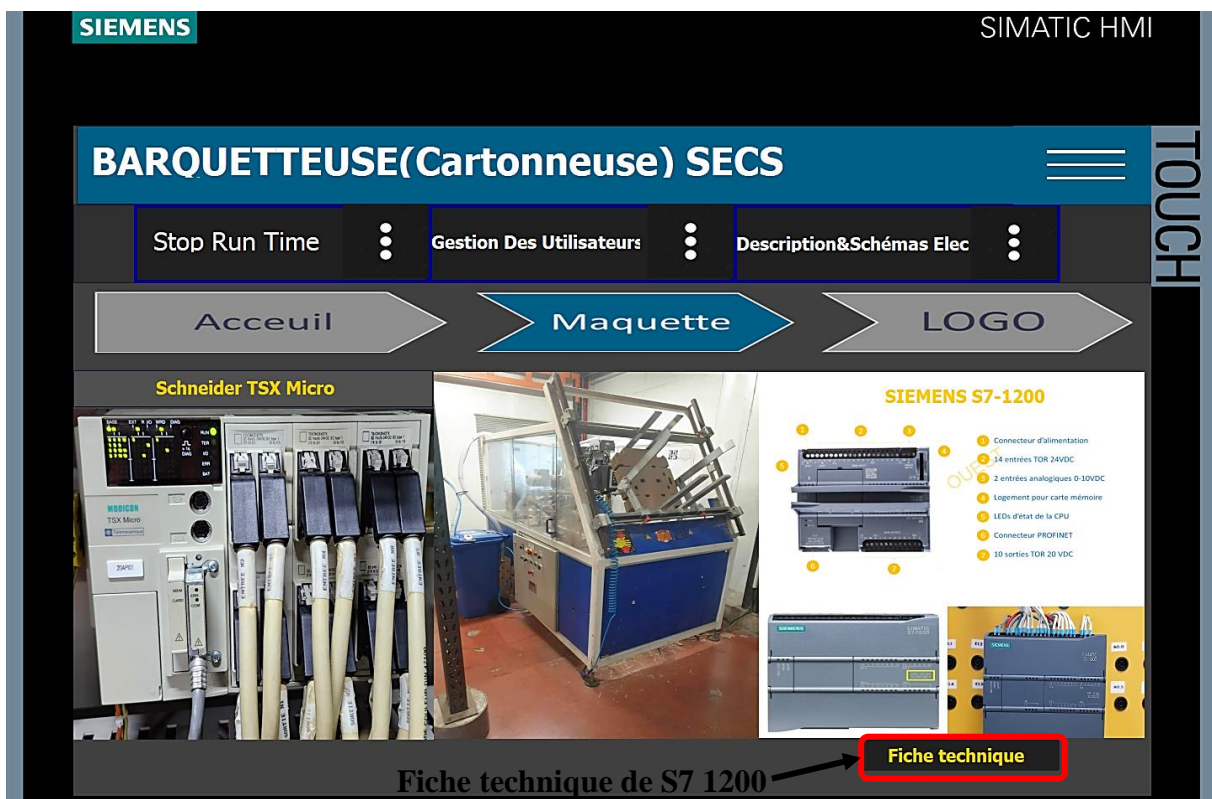


Figure IV. 7 : Vue de la machine

Si on clique sur le bouton « Fiche technique » dans la figure précédente un document pdf s'affiche, ce document contient les caractéristiques de l'automate S7 1200.

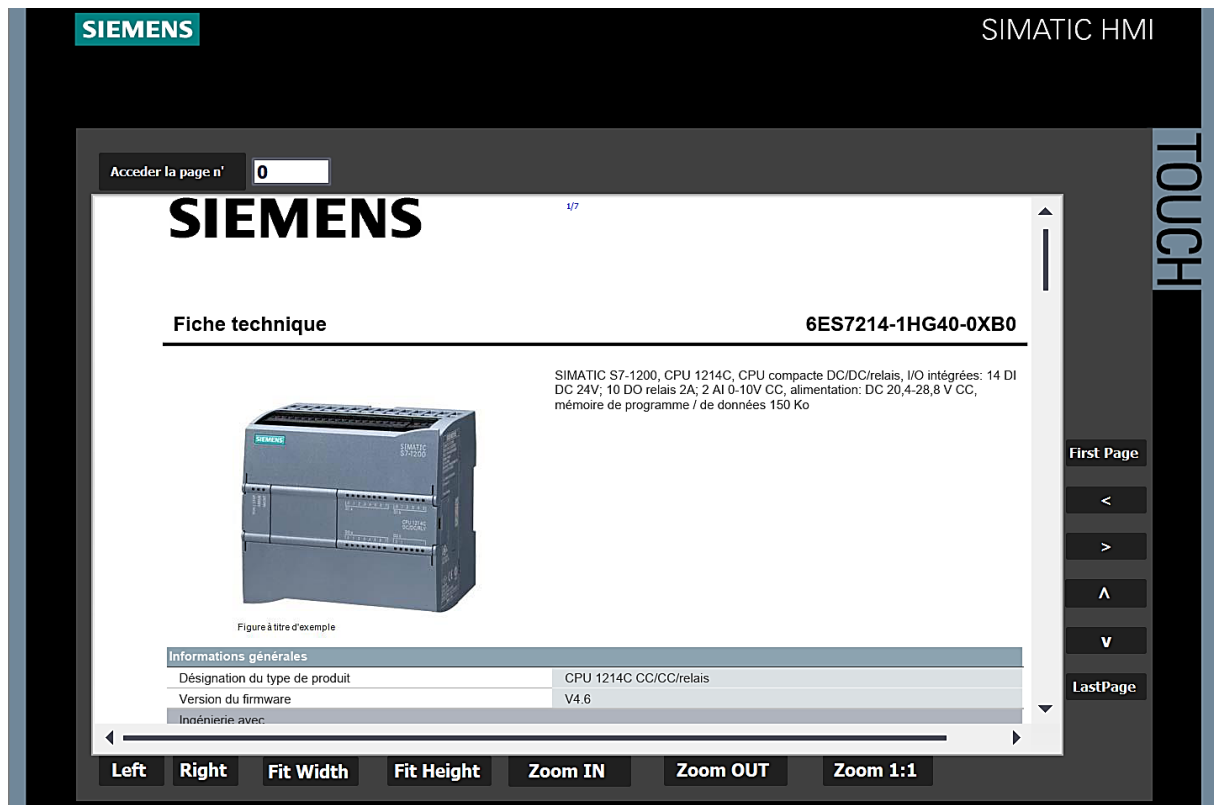


Figure IV. 8 : Fiche technique de l'automate SIEMENS S7 1200

Configuration des boutons

- Le bouton « First Page » : Après avoir ajouté un bouton, (1) on clique sur « Presser », ensuite on clique sur (2) « Evènement » et (3) on choisit la fonction « PDFVerslaPremièrePage », la figure suivante montre ces étapes :

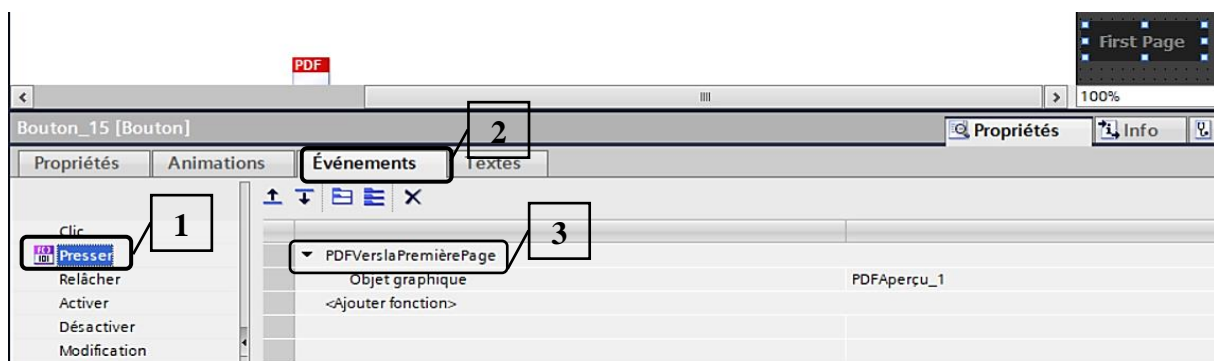
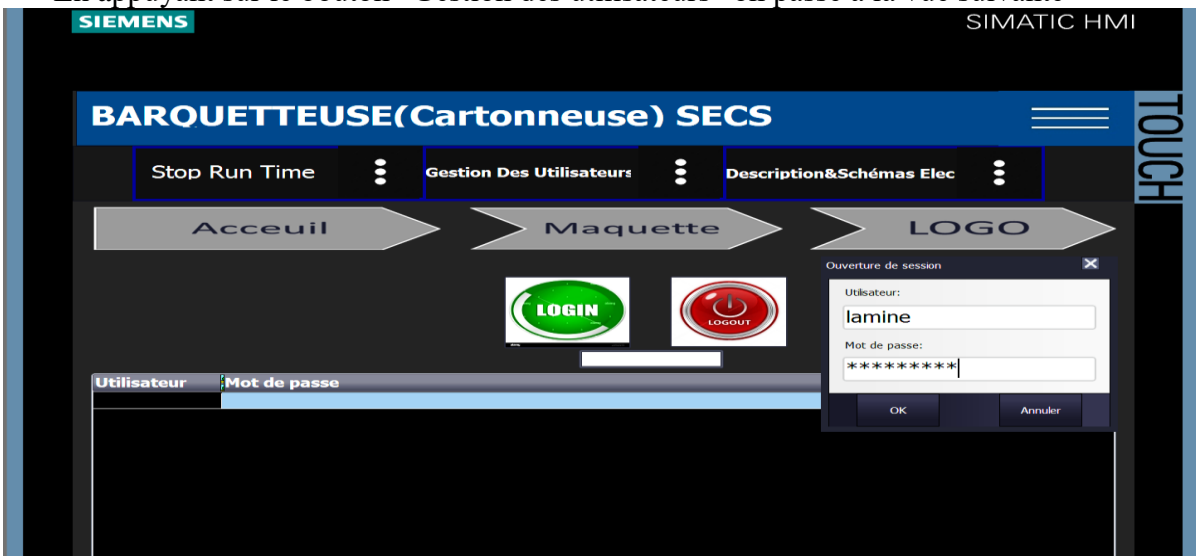


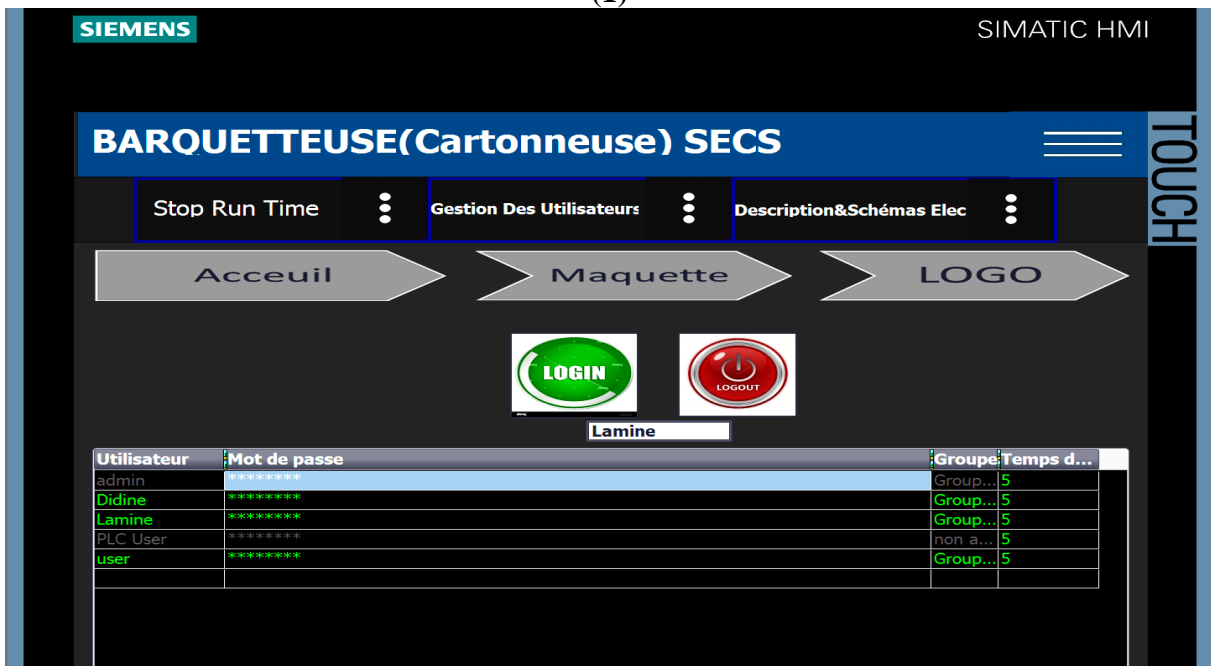
Figure IV. 9 : Configuration d'un bouton "First Page"

- Le bouton Précédent (<) : en suivant les mêmes étapes précédentes et change la fonction, cette fois on choisit la fonction « PDFVerslaPagePrécédente »
- Le bouton suivant (>) : on choisit la fonction « PDFVerslaPageSuivante »
- Le bouton vers le haut (^) : on choisit la fonction « PDFDéfilementversHaut »
- Le bouton vers le bas (v) : on choisit la fonction « PDFDéfilementversBas »
- Le bouton « LastPage » : on choisit la fonction « PDFVerslaDernièrePage »

- Le bouton « **Zoom 1 :1** » : on choisit la fonction « PDFZoomOriginal »
- Le bouton « **Zoom OUT** » : on choisit la fonction « PDFZoommoins »
- Le bouton « **Zoom IN** » : on choisit la fonction « PDFZoomplus »
- Le bouton « **Fit Height** » : on choisit la fonction « PDFAdapterHauteur »
- Le bouton « **Fit Width** » : on choisit la fonction « PDFAdapterLargeur »
- Le bouton « **Right** » : on choisit la fonction « PDFDéfilementDroite »
- Le bouton « **Left** » : on choisit la fonction « PDFDéfilementGauche »
- En appuyant sur le bouton “Gestion des utilisateurs” on passe à la vue suivante



(1)



(2)

Figure IV. 10 : Vue Gestion des utilisateurs
 (1) Affichage de la fenêtre « Ouverture de session »
 (2) La vue gestion des utilisateurs

La figure suivante montre les étapes (de 1 à 5) de configuration de cette vue.

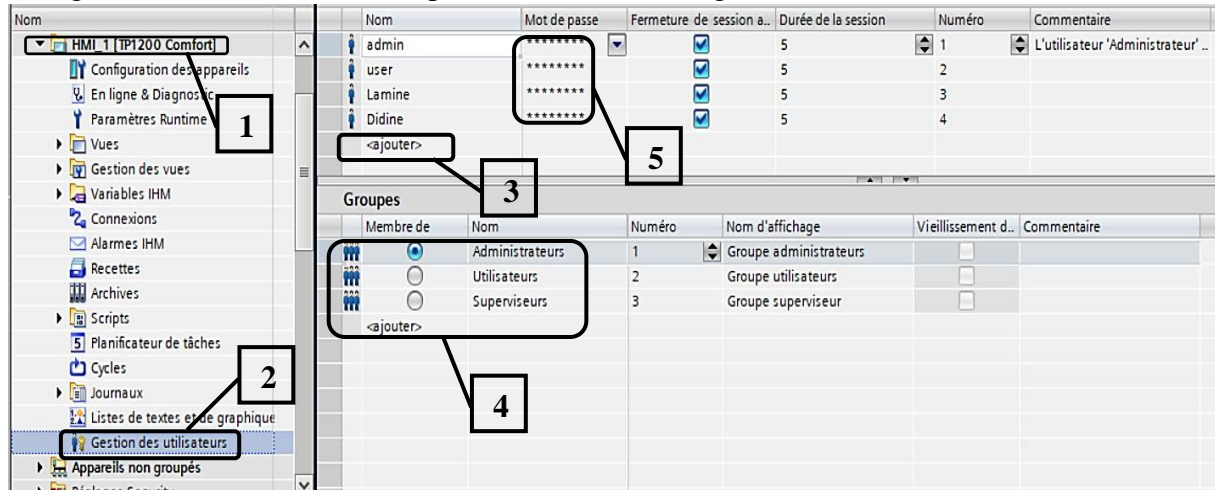


Figure IV. 11 : Configuration du vue « Gestion des utilisateurs »

- En appuyant sur le bouton en haut à droite, le menu s’affiche et sur ce dernier on peut sélectionner la vue qu’on veut activer.

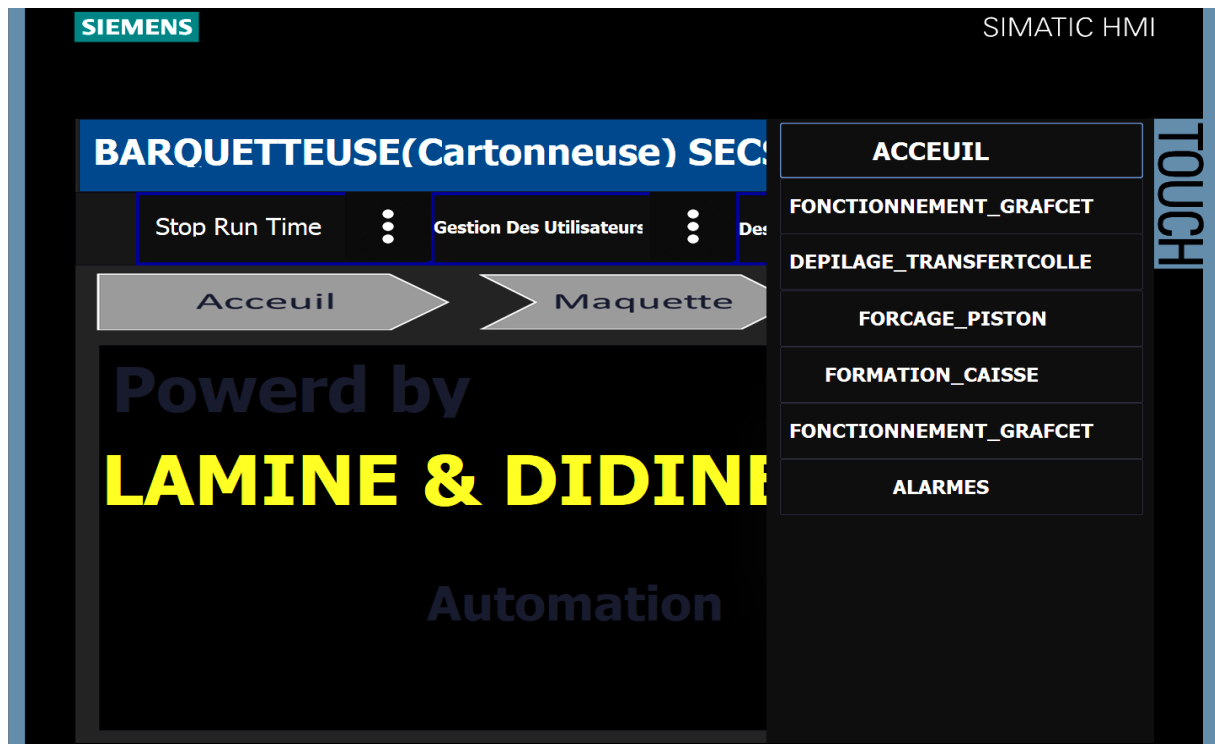


Figure IV. 12 : Vue de Menu

IV.5 Vues des GRAFCETs

Pour configurer les GRAFCETs de notre machine il faut qu’on ajoute des variables IHM, tel qu’on distingue deux types de variables, les variables externes et les variables internes :

Les variables externes : Ils permettent d’échanger des données entre les différents éléments d'un processus automatisé et un ordinateur de gestion. L'image d'une cellule mémoire

spécifiée de l'automate est appelée une variable externe. Cette cellule mémoire est accessible en lecture et en écriture à partir du pupitre de l'opérateur et de l'automate. La figure suivante montre un tableau des variables externes pour les sorties, même chose pour les étapes.

Nom ▲	Type de données	Connexion	Nom API	Variable API	Adresse	Mode d'accès	Cycle d'acquisit..
EV_COLLE	Bool	HMI_Liaison_1	PLC_1	EV_Colle		<accès symbolique>	1 s
EV_EQUERRAGE	Bool	HMI_Liaison_1	PLC_1	EV_Equerrage		<accès symbolique>	1 s
EV_PISTON_VIDE	Bool	HMI_Liaison_1	PLC_1	EV_Piston		<accès symbolique>	1 s
EV_PRESSAGE	Bool	HMI_Liaison_1	PLC_1	EV_Pressage		<accès symbolique>	1 s
EV_PRISE	Bool	HMI_Liaison_1	PLC_1	EV_Prise		<accès symbolique>	1 s
EV_RAMPE	Bool	HMI_Liaison_1	PLC_1	EV_Rampe		<accès symbolique>	1 s
FORCAGE_PISTON_BAS	Bool	HMI_Liaison_1	PLC_1	FORCAGE_PISTON_BAS		<accès symbolique>	1 s
FORCAGE_PISTON_HAUT	Bool	HMI_Liaison_1	PLC_1	Forçage_Piston_Haut		<accès symbolique>	1 s
KM4_CONVOYEUR_SORTIE	Bool	HMI_Liaison_1	PLC_1	Forçage_Piston_Bas		<accès symbolique>	1 s
KM4_MOTEUR_PISTON	Bool	HMI_Liaison_1	PLC_1	Moteur_piston		<accès symbolique>	1 s
TRANSFERT_TAQUET	Bool	HMI_Liaison_1	PLC_1	Transfert_Taquet		<accès symbolique>	1 s

Figure IV. 13 : Table des variables externes (les sorties)

Les variables internes : Ils ne sont pas liés à l'automate. Ces informations sont stockées dans la mémoire du pupitre. Ainsi, le pupitre operateur en question a la possibilité de consulter et d'écrire les variables internes. Par exemple, elles sont conçues afin de réaliser des calculs locaux à l'aide de WinCC. Les variables sont définies dans l'éditeur "Variables". Selon leur déclaration, ils ont reçu une configuration de base. Grâce à l'éditeur "Variables", nous avons la possibilité d'ajuster cette configuration en fonction de nos besoins spécifiques. L'éditeur s'ouvre automatiquement en ajoutant une variable. La figure suivante montre un tableau des variables internes (locaux).

Nom ▲	Type de données	Connexion
_acc	Bool	<Variable int... >
log	Bool	<Variable intern... >
maq	Bool	<Variable intern... >
vid	Bool	<Variable intern... >

Figure IV. 14 : Table des variables internes

IV.5.1 La configuration des étapes

Après avoir ajouté les tables des variables internes et externes on peut maintenant configurer les étapes des GRAFCETs tout d'abord on ajoute un rectangle, après on suit les étapes suivantes (de 1 à 4)

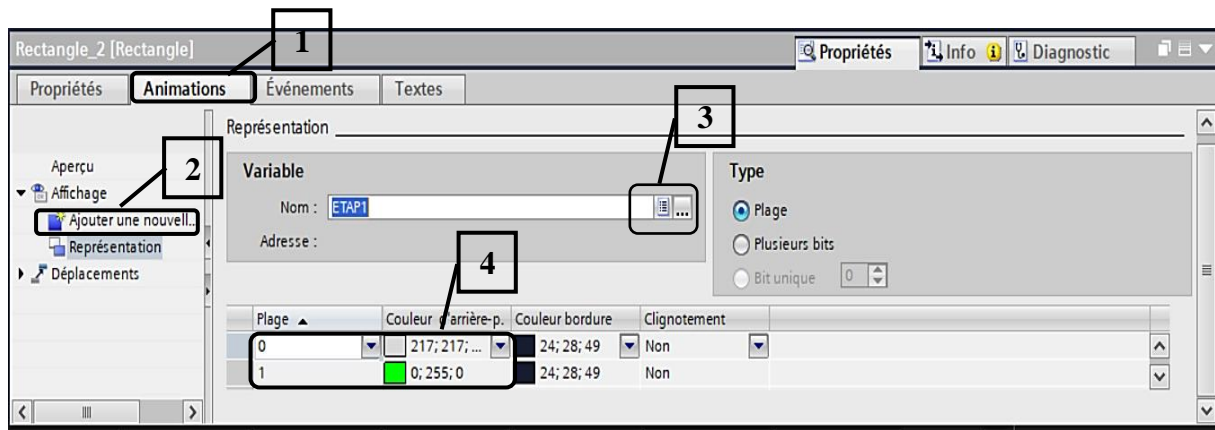


Figure IV. 15 : Configuration d’une étape

On clique sur le rectangle créé et on clique sur « Propriétés », après (1) on clique sur « Animation » ensuite (2) on clique sur « Ajouter une nouvelle animation » et (3) en ajoutant la variable qui correspond, enfin on détermine l’état d’étape verte si elle est active, sinon grise.

IV.5.2 Configuration des actions

On suit les mêmes étapes pour configurer une étape de GRAFCET, on change seulement la variable correspondante, pour visualiser l’action de chaque étape on a utilisé des vues contextuelles et on a ajouté des boutons pour les affichées.

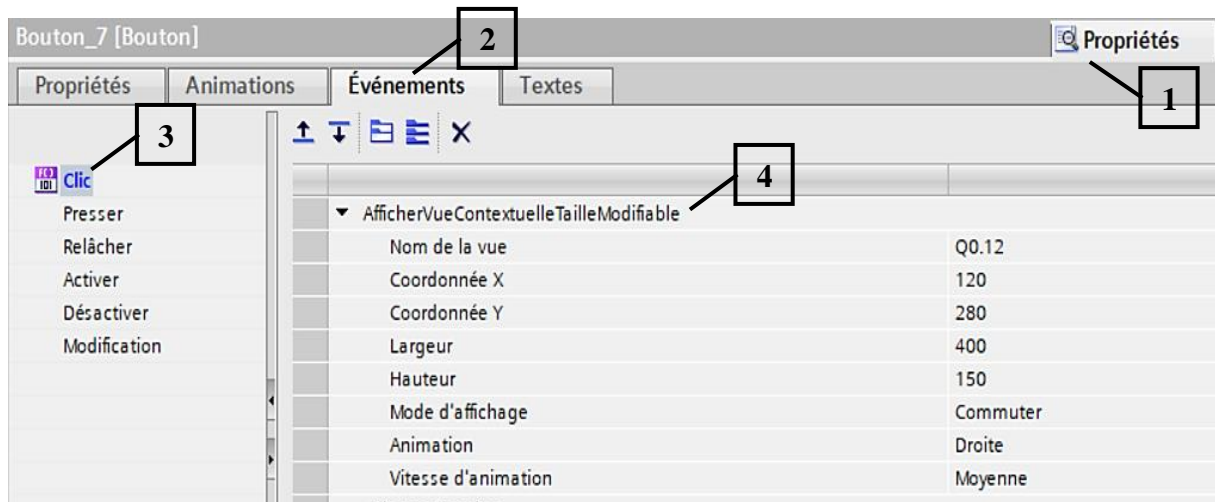


Figure IV. 16 : Etapes de configuration du bouton qui affiche la vue textuelle

IV.5.3 Configuration des transitions

Après qu’on ajoute un bouton on garde les mêmes étapes mentionnées dans la partie de configuration des boutons des actions, voici les étapes d’ajouter une vue contextuelle et la configurer :

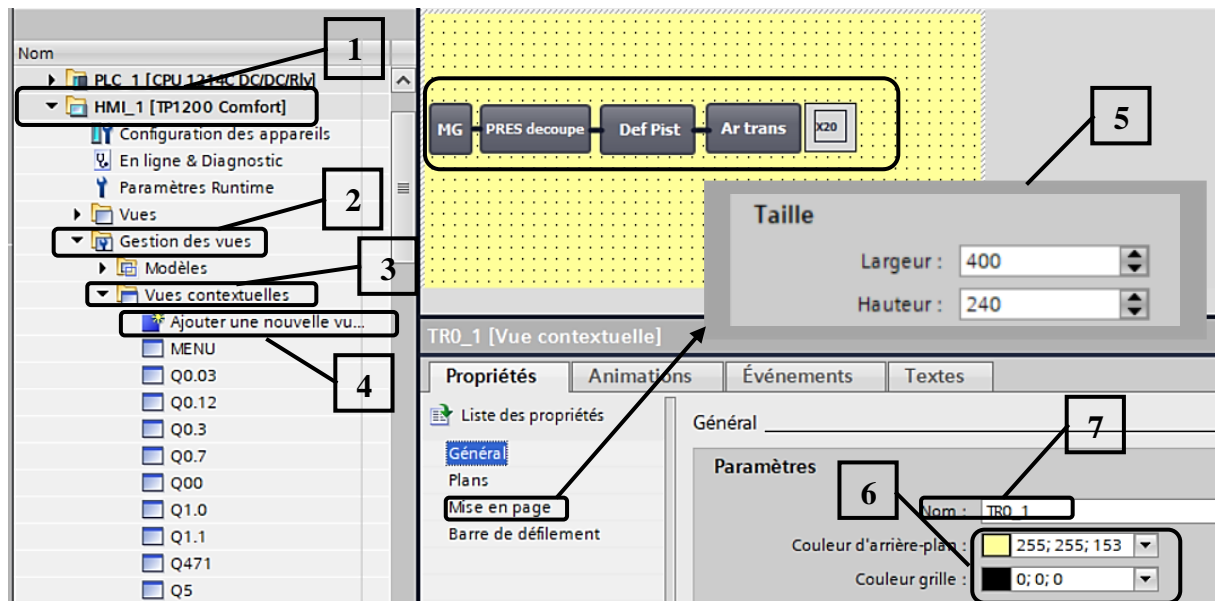


Figure IV. 17 : Etape de configuration d'une transition

IV.6 Vues des GRAFCETs

Pour chaque GRAFCET on a ajouté des boutons pour lire des vidéos qui montre le fonctionnement de la machine.

IV.6.1 GRAFCET « Dépilage_Transfert_Collage »

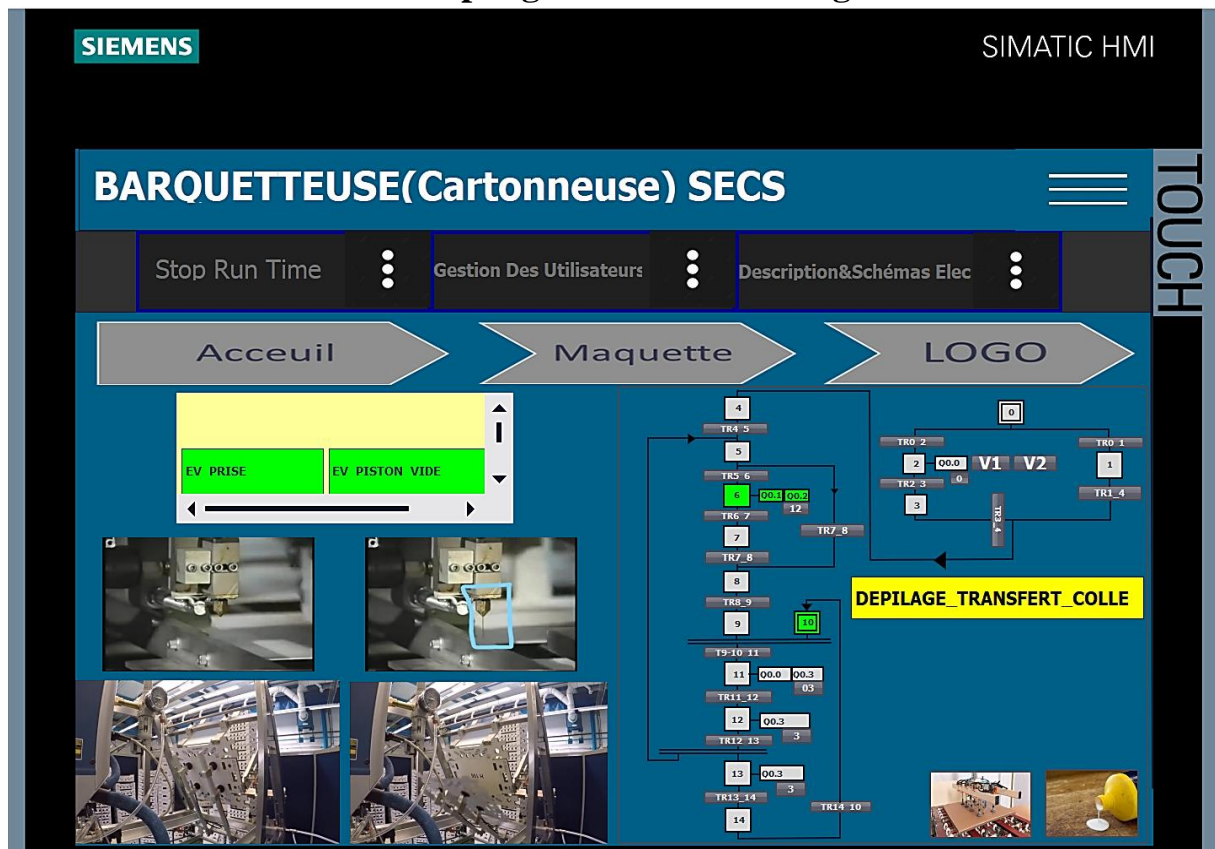


Figure IV. 18 : Vue GRAFCET de « Dépilage_Transfert_Collage »

Pour configurer les deux boutons qui sont à droite en bas qui lire des vidéos, on suit les étapes suivantes :

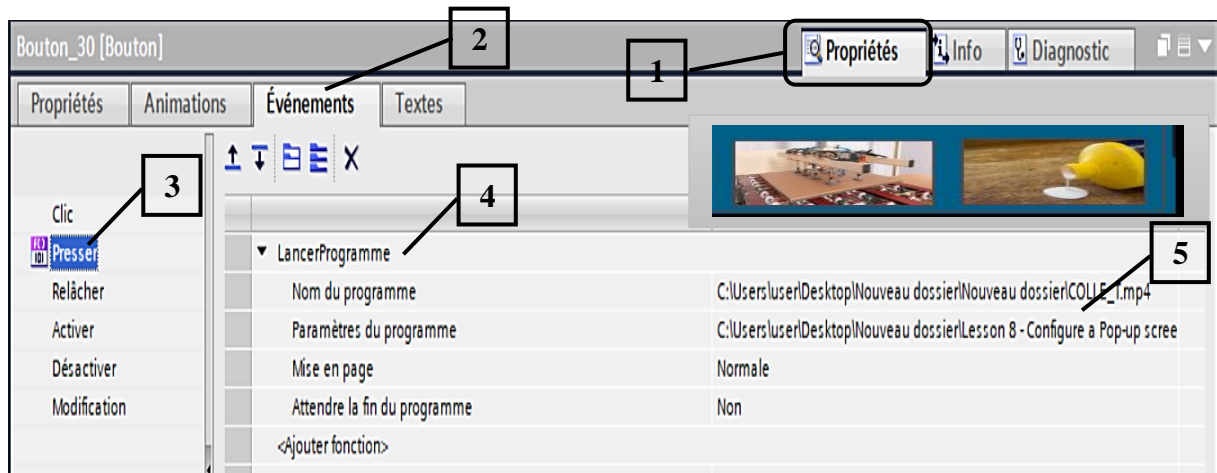


Figure IV. 19 : Configuration des boutons « Vidéo »

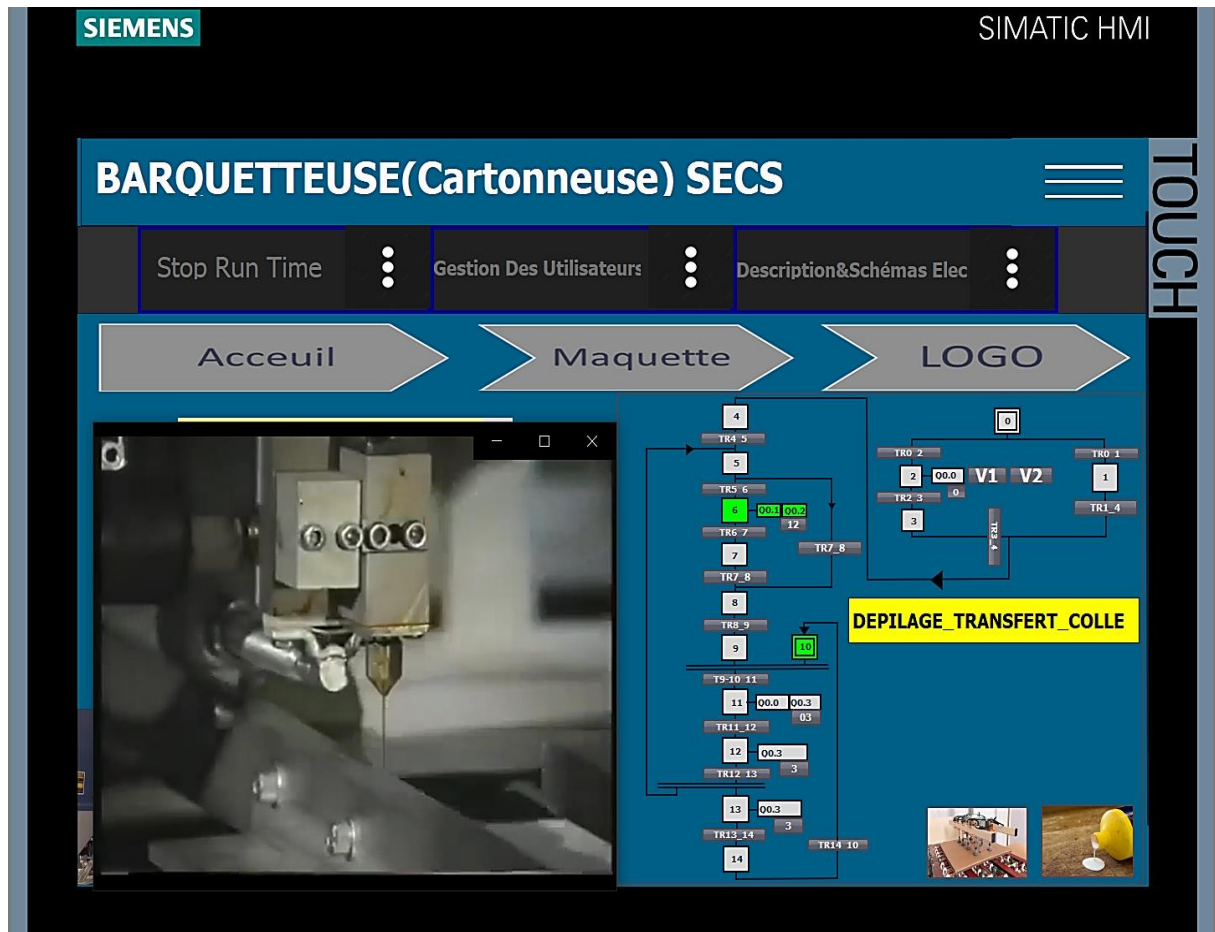


Figure IV. 20 : Exemple de lecture d'une vidéo

IV.6.2 GRAFCET « Formation_Caisse »

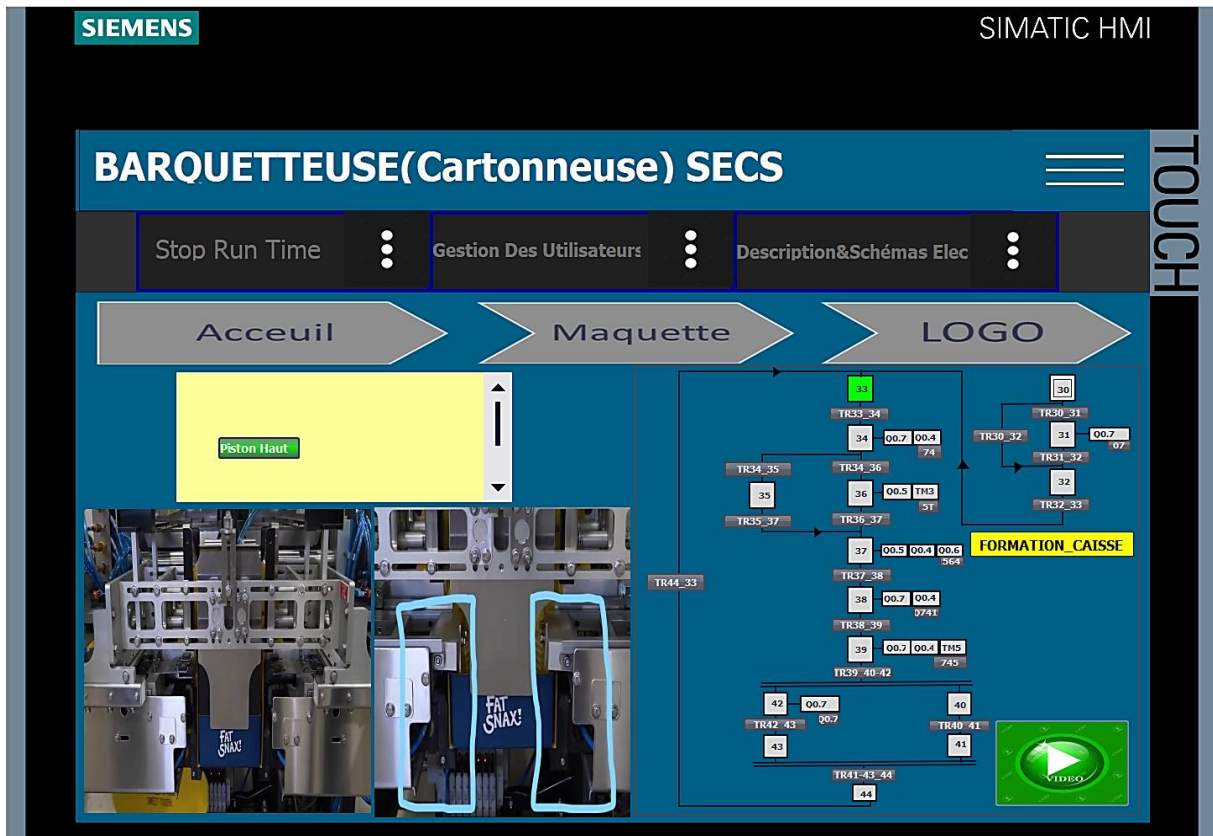


Figure IV. 21 : Vue GRAFCET de « Formation_Caisse »

IV.6.3 GRAFCET « Forçage_Piston »

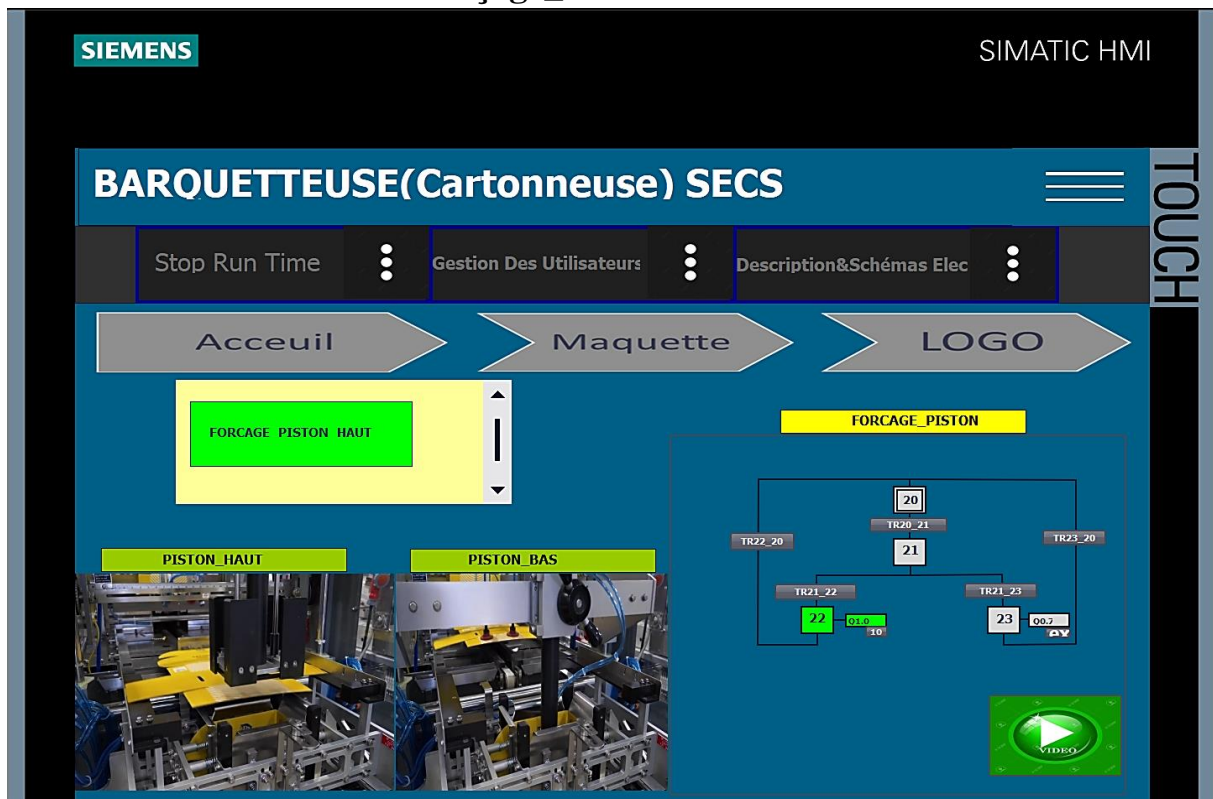


Figure IV. 22 : Vue GRAFCET de « Forçage_Piston »

IV.7 Vue générale

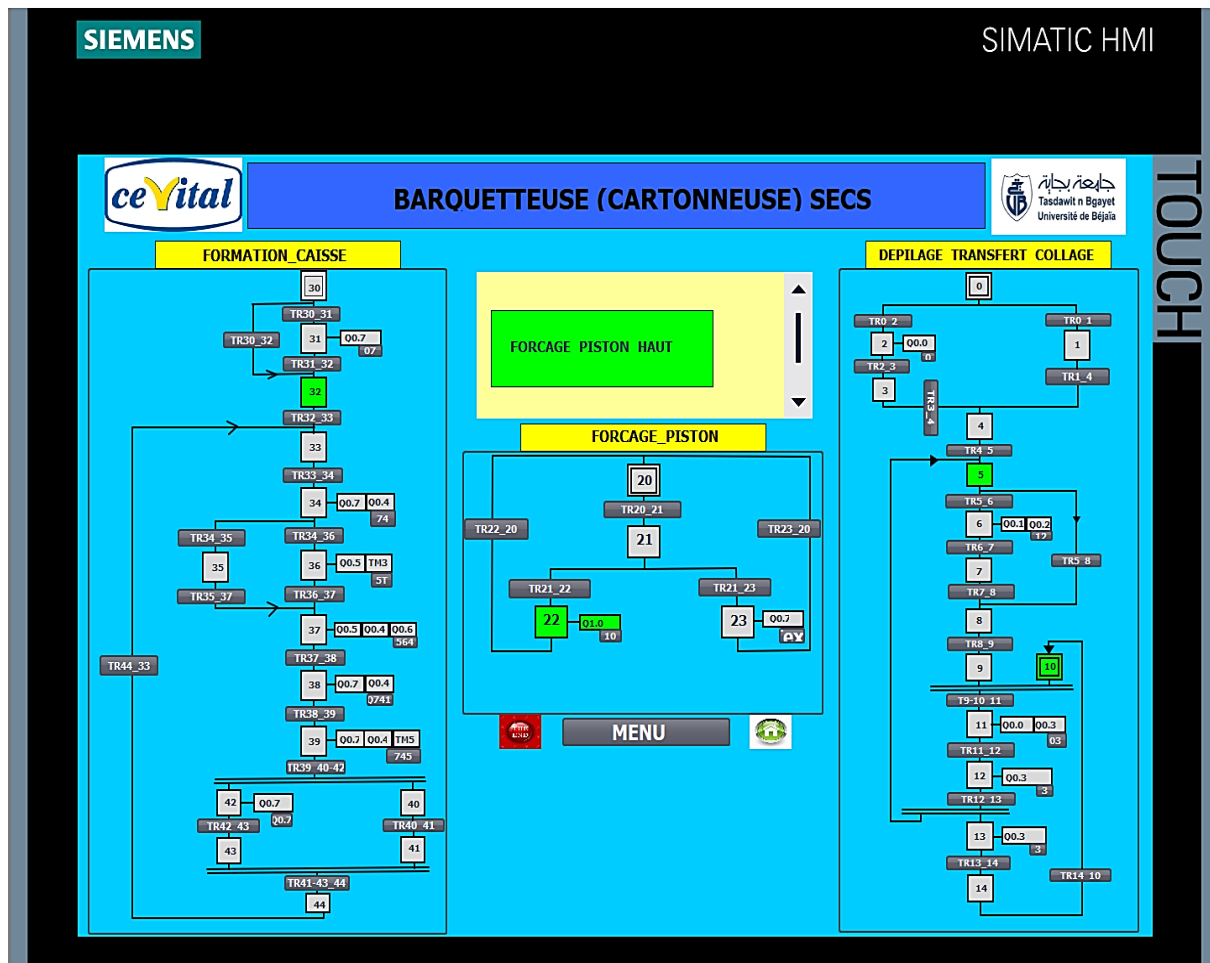


Figure IV. 23 : Vue générale

Lors de la conception du programme nous avons optés en premier lieu pour un automate compact sans ajouter un module d’entrées/sorties, ce choix nous est venu afin de réduire le cout, prochainement nous avons pensé au facteur humain, et pour éviter tout confusion et de permettre à l’opérateur un usage facile, nous avons ajouté un module d’entrées/sorties en vue d’installer un bouton d’initiation et un autre pour l’arrêt d’urgence.

On a choisi d’ajouter un module DI/DQ au lieu un module DI, car la déférence de prix entre ces deux modules est petite et c’est possible d’utiliser des sorties additives dans le future. Pour ajouter ce module on va suivre les étapes mentionnés dans la figure suivante :

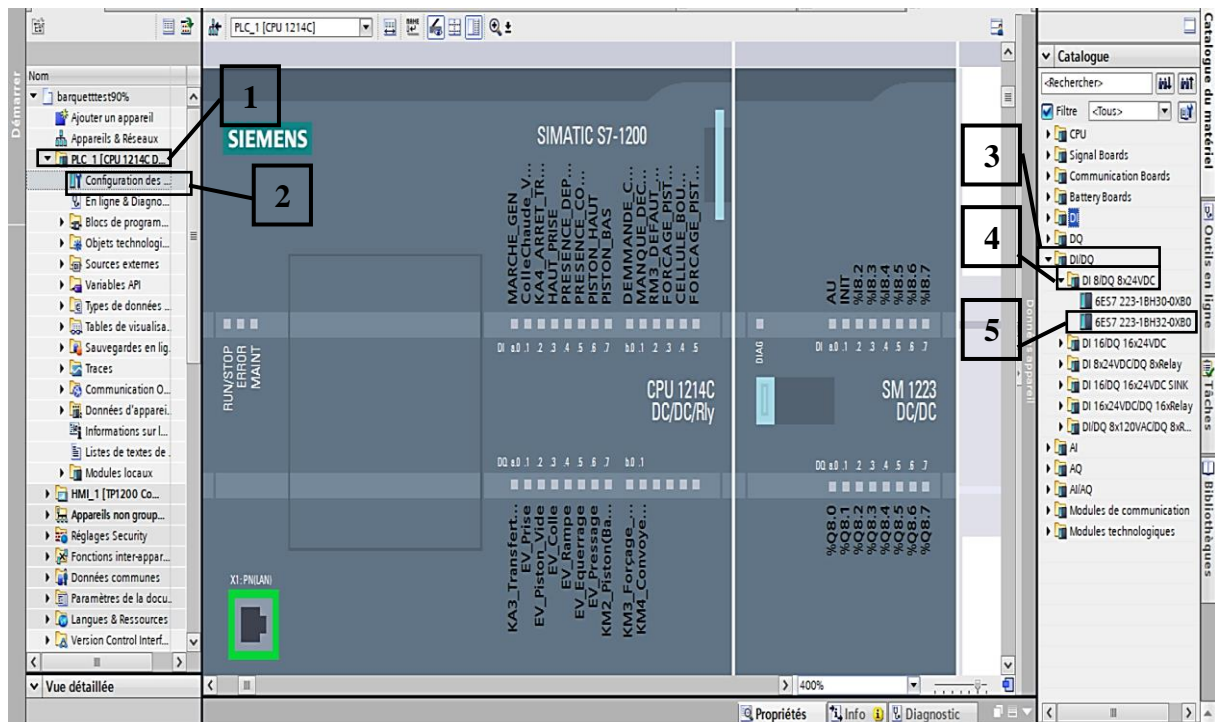


Figure IV. 24 : Etapes pour ajouter un module DI/DQ

- D’abord (1) on clique sur le dossier « PLC_1 [CPU1214C DC/DC/Rly] »
- (2) on clique sur « Configuration des appareils »
- Après avoir cliqué sur la CPU, (3) on clique sur le dossier « DI/DQ »
- (4) on clique sur le dossier « DI 8/DQ 8x24vDC », (5) on glisse le module sur le rack

Maintenant il nous reste de configurer les deux boutons ajoutés, bien sûr il faut les affectés au adresses et les ajoutés dans le bloc des données DB.

INPUTS							Bloc de données_1							
Nom	Type de données	Adresse	Réma...	Acces...	Écritu...	Visibi...	Nom	Type de données	Valeur de départ	Rémanence	Accessible	Écritu...	Visible da...	Valeur de ...
1	MARCHE_GEN	Bool	%I0.0				1	Static						
2	ColleChaude_VideArretMarche	Bool	%I0.1				2	I0-0	Bool	false				
3	KA4_ARRET_TRANSFERT	Bool	%I0.2				3	I0-1	Bool	false				
4	HAUT_PRISE	Bool	%I0.3				4	I0-2	Bool	false				
5	PRESENCE_DEPILAGE	Bool	%I0.4				5	I0-3	Bool	false				
6	PRESENCE_COLLE	Bool	%I0.5				6	I0-4	Bool	false				
7	PISTON_HAUT	Bool	%I0.6				7	I0-5	Bool	false				
8	PISTON_BAS	Bool	%I0.7				8	I0-6	Bool	false				
9	DEMANDE_CAISSSE	Bool	%I1.0				9	I0-7	Bool	false				
10	MANQUE_DECOUPE	Bool	%I1.1				10	I1-0	Bool	false				
11	RM3_DEFAULT_PISTON	Bool	%I1.2				11	I1-1	Bool	false				
12	FORCAGE_PISTON_BAS	Bool	%I1.3				12	I1-2	Bool	false				
13	CELLULE_BOURRAGE_CONVOY...	Bool	%I1.4				13	I1-3	Bool	false				
14	FORCAGE_PISTON_HAUT	Bool	%I1.5				14	I1-4	Bool	false				
15	AU	Bool	%I8.0				15	I8-0	Bool	false				
16	INIT	Bool	%I8.1				16	I8-1	Bool	false				

(1) (2)
 Figure IV. 25 : Tableaux qui montre les 2 variables ajoutées
 (1) Les deux variables ajoutées dans la table INPUTS
 (2) Les deux variables ajoutés dans le Bloc de données_1 (DB)

On ajoute maintenant un réseau dans le bloc OB1 pour initialiser les GRAFCETs, la figure suivante montre ce réseau :



Figure IV. 26 : Réseau d’initialisation des GRAFCETs

Parmi les méthodes de configurer un bouton d’urgence et de désactiver toutes les étapes en ajoutant un contact NC après chaque étape la figure IV.27 montre ce réseau, et on désactive toutes les actions pour cela on ajoute un bloc FC6.

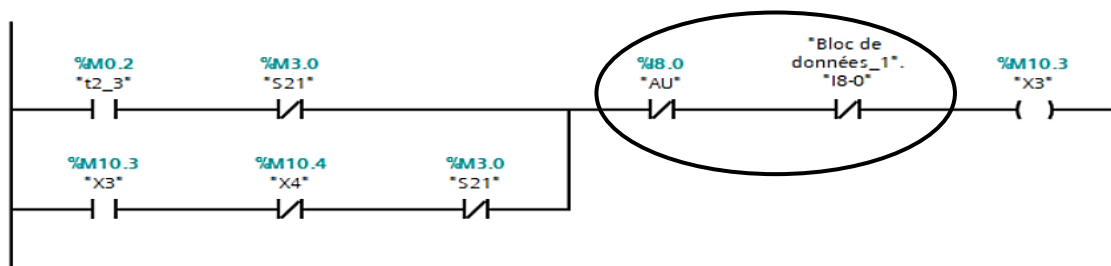


Figure IV. 27 : Le Bouton arrêt d’urgence

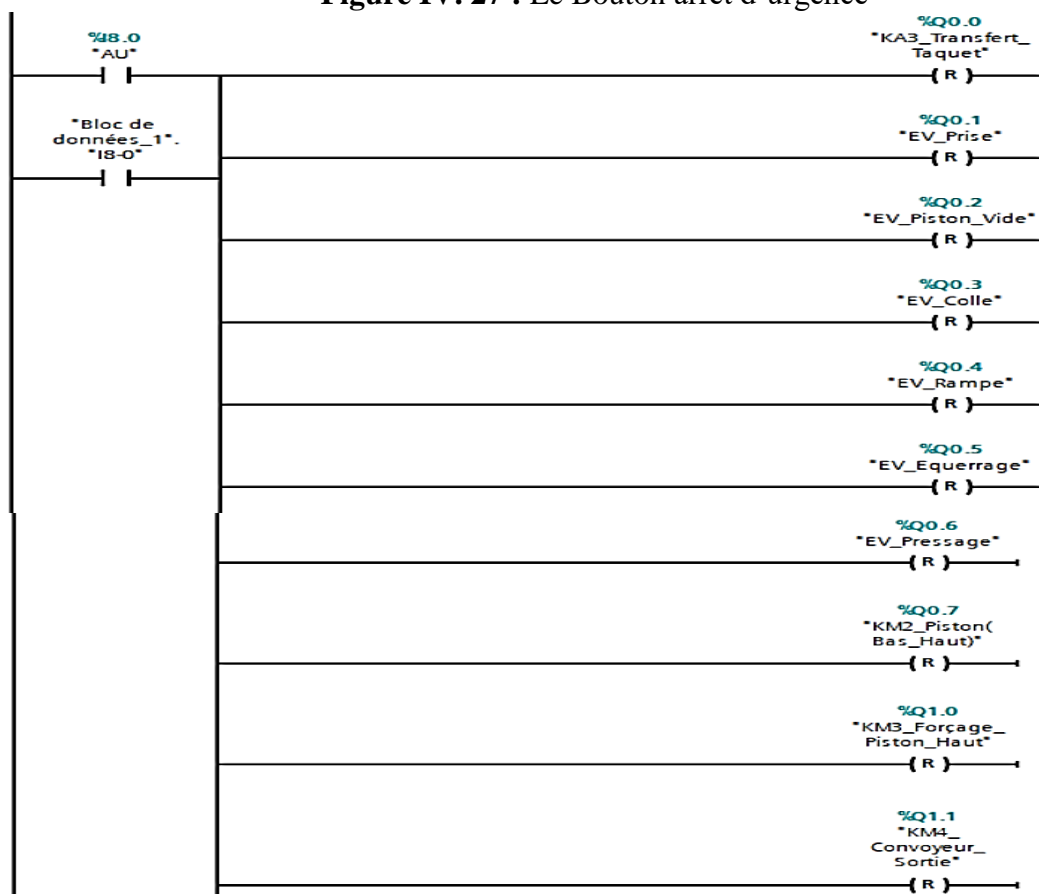


Figure IV. 28 : Désactivation des étapes

Le système avant de cliquer sur le bouton « INIT »

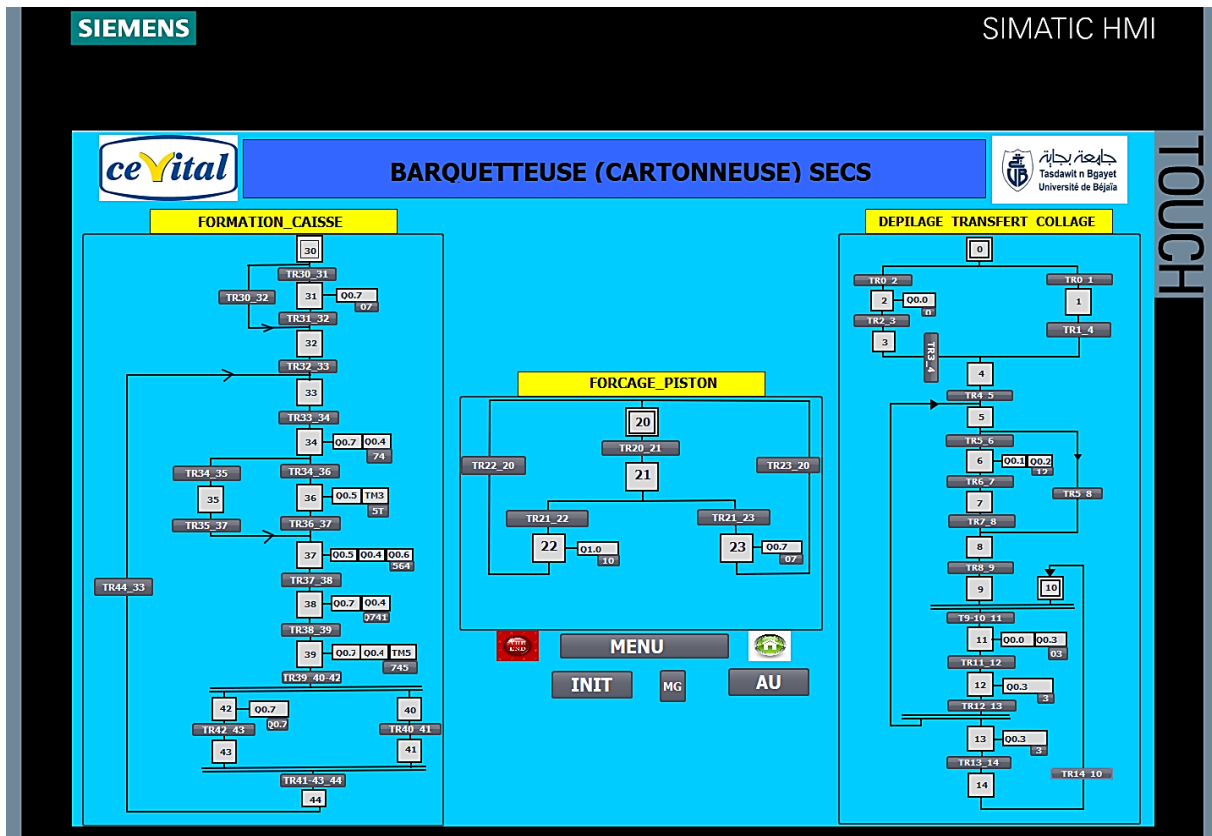


Figure IV. 29 : Vue générale avant l'initialisation

Après l'appui sur INIT

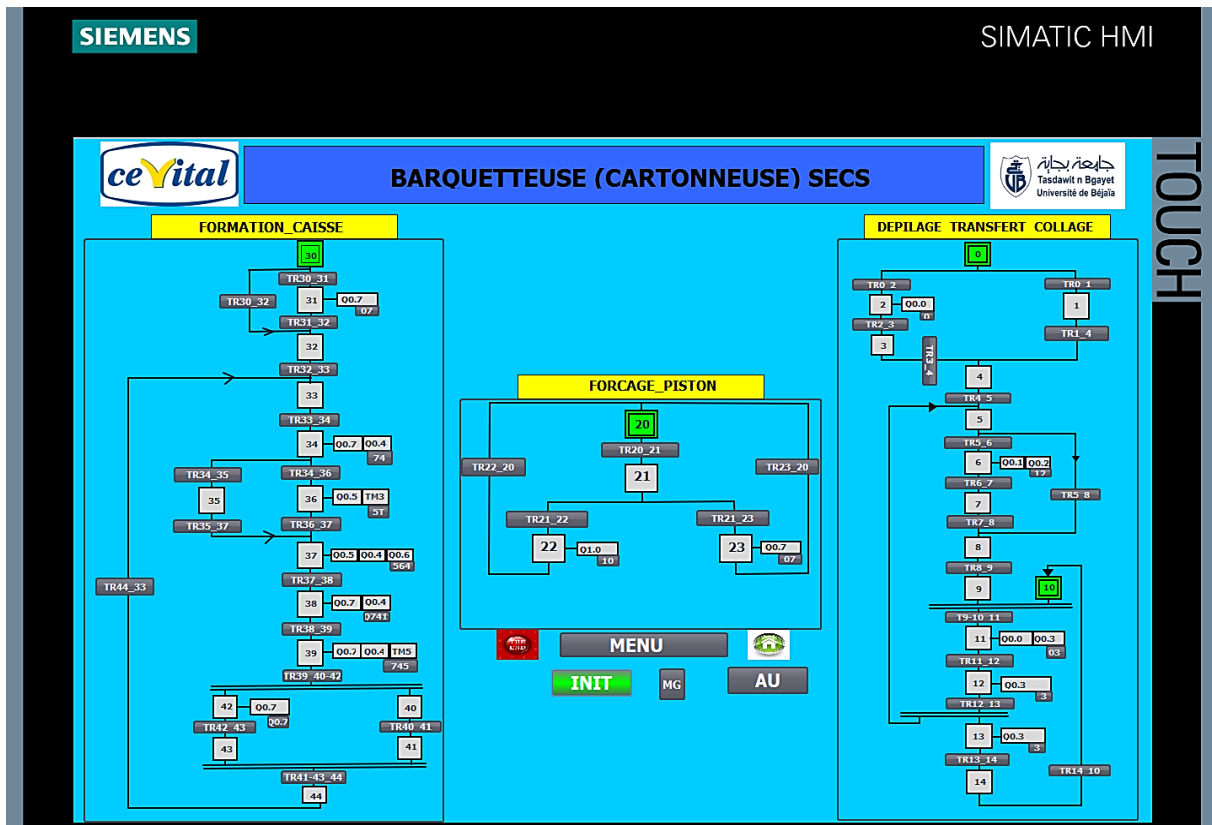


Figure IV. 30 : Vue générale après l'initialisation

Si on appui sur le bouton « AU » :

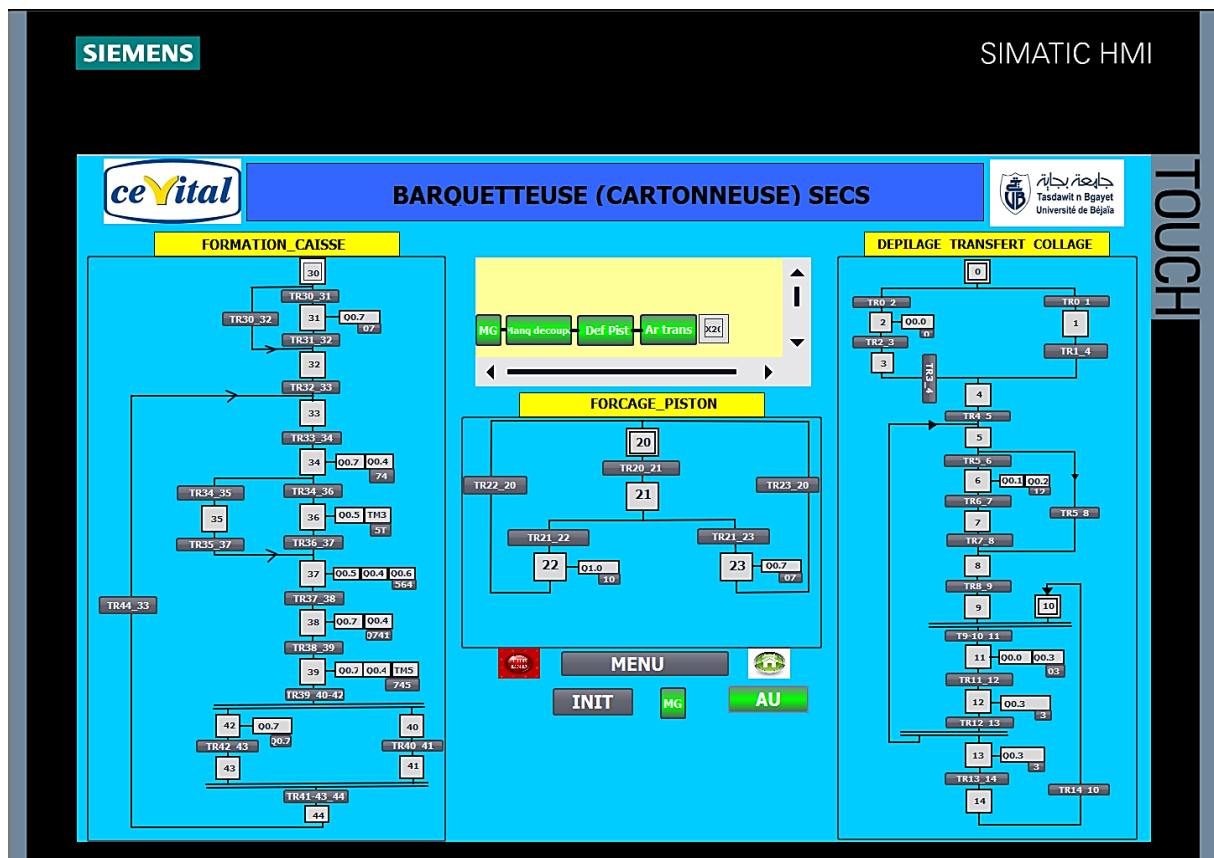


Figure IV. 31 : Vue générale après l'appui sur « AU »

IV.8 AlarmesIHM

Les alarmes indiquent les situations où des anomalies peuvent se produire pendant le processus, elles peuvent être utilisées pour diagnostiquer les erreurs, par exemple.

IV.8.1 Classes d'alarmes

- **"Erreurs"** Les alarmes TOR et analogiques sont utilisées pour signaler des états critiques ou dangereux du fonctionnement et du processus. Il est essentiel d'acquiescer toujours les alarmes de cette classe.
- **"Avertissements"** Les alarmes TOR et analogiques sont utilisées pour signaler les états normaux du fonctionnement et du processus, ainsi que les déroulements du processus. Les alarmes de cette catégorie ne nécessitent pas d'acquiescement.
- **"Système"** L'ensemble des alarmes de la classe "System" comprend les alarmes qui signalent les états du pupitre opérateur et des automates. Les messages de la catégorie "Système" font partie des messages système.

- " **Diagnostic**" Les alarmes de la classe "Diagnostic Events" comprennent les alarmes qui signalent les états et les événements des automates SIMATIC S7. L'utilisateur ne répond pas aux alarmes de cette catégorie [29].

Dans notre projet on a ajouté deux autres classes pour regrouper nos alarmes, la figure suivante montre ces deux classes

Nom d'affichage	Nom	Automate fini	Archive	Adresse e-mail	Couleur...	Couleur...	Couleur...	Couleur...
!	Errors	Alarme à acquittement si...	<aucune archiv...		255...	255...	255...	255...
	Warnings	Alarme sans acquittement	<aucune archiv...		255...	255...	255...	255...
\$	System	Alarme sans acquittement	<aucune archiv...		255...	255...	255...	255...
S7	Diagnosis events	Alarme sans acquittement	<aucune archiv...		255...	255...	255...	255...
A	Acknowledgement	Alarme à acquittement si...	<aucune archiv...		255...	255...	255...	255...
NA	No Acknowledgement	Alarme sans acquittement	<aucune archiv...		255...	255...	255...	255...
ALARMES_BARQUETTEUSE	ALARMES_BARQUETTEUSE	Alarme à acquittement si...	<aucune archiv...		255...	255...	255...	255...
alarme_analog	alarme_analog	Alarme à acquittement si...	<aucune archiv...		255...	255...	255...	255...

Figure IV. 32 : Vue classes d’alarmes

IV.8.2 Types des alarmes

On trouve dans le WinCC les types d’alarmes suivants :

1. Le système établit des alarmes pour surveiller le pupitre opérateur ou l'automate.
2. Des alarmes qui sont configurées par l'utilisateur pour surveiller l'installation.
 - **Les alarmes analogiques** permettent de détecter les dépassements de valeurs limites dans le processus en développement.
 - **Les alertes de bits** signalent un état dans le processus en cours.
 - **Les alarmes utilisateur** signalent des actions effectuées par les utilisateurs dans Runtime. Dans Runtime, il est possible de configurer des alarmes utilisateur dans des scripts pour des applications spécifiées par l'utilisateur.
 - **Les alarmes de l'API** : Le concepteur de l'automate programme une alarme de l'API définie par l'utilisateur dans STEP 7. On y intègre les valeurs d'état de l'automate, comme l'heure et les valeurs de processus. Lorsque l'on configure des alarmes de l'API dans STEP 7, elles sont reprises dans le fonctionnement intégré de WinCC dès qu'une connexion à l'automate est établie [29].

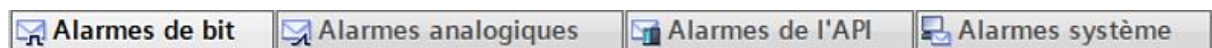


Figure IV. 33 : Types des alarmes

IV.8.3 Configuration d’alarme

On sélectionne "Alarmes IHM" sur le navigateur de projet, nous allons ouvrir le tableau "Alarmes de bit" et le remplir avec les bits de défaut qui déclenchent les alarmes en cas d'occurrence d'un défaut. Pour cela, nous allons choisir la variable "Défauts" de type "Int" à

partir de la table des variables standards, dans la colonne "Variable de déclenchement". Ensuite, nous sélectionnons le bit de déclenchement qui est identifié par son numéro dans le mot %MW20, par exemple : le bit numéro 8 est %M20.0.

Alarmes de bit									
ID	Nom	Texte d'alarme	Classe d'alar...	Variable de d...	Bit de ..	Adresse de dé..	Variable d'ac...	Bit d'a..	Adresse d'acq..
1	Alarme_de_bit_1	ATTENTION_MANQUE_DECOUPE	ALARMES_...	ALARMES_...	8	%M21.0	acquiteme...	0	acquitement...
2	Alarme_de_bit_2	ATTENTION_ABSENCE_COLLE	ALARMES_BA...	ALARMES_TOR	9	%M21.1	acquitement	1	acquitement...
3	Alarme_de_bit_3	ATTENTION_DEFAUT_PISTON	ALARMES_BA...	ALARMES_TOR	10	%M21.2	acquitement	2	acquitement...
4	Alarme_de_bit_4	ATTENTION_ABSENCE_DEPILAGE	ALARMES_BA...	ALARMES_TOR	11	%M21.3	acquitement	3	acquitement...
<ajouter>									

Figure IV.1 : Table des alarmes IHM

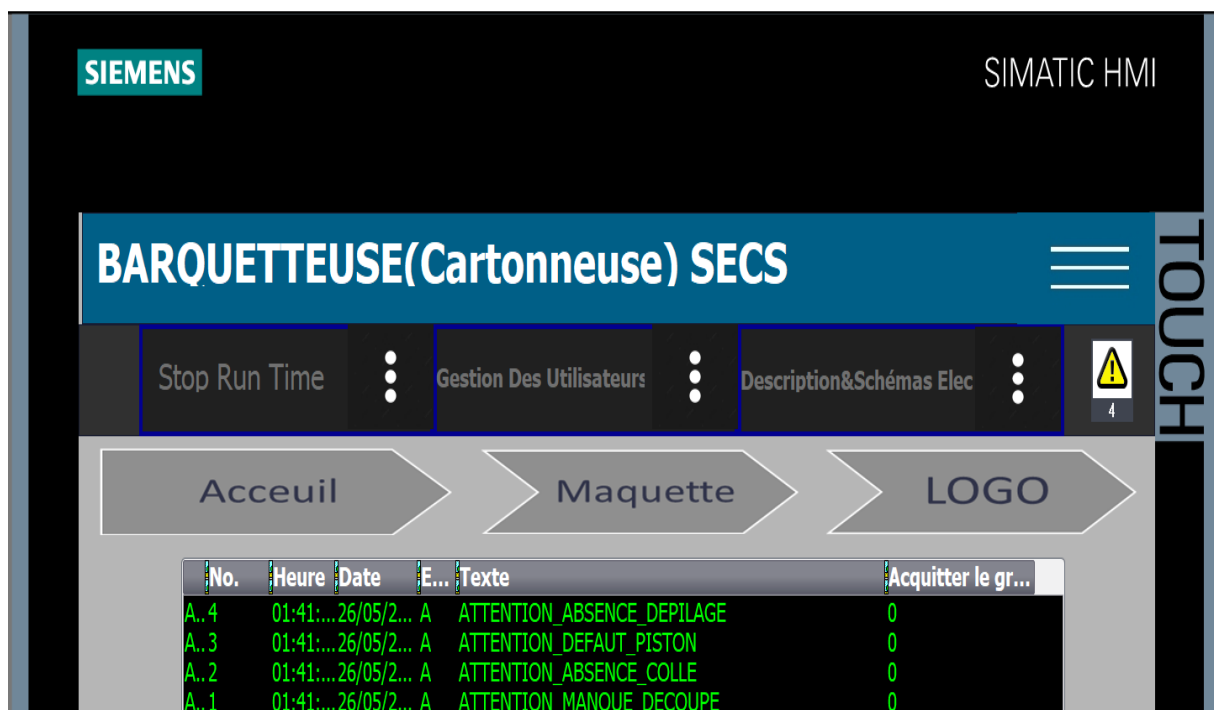


Figure IV. 34 : Vue des alarmes TOR

Archives d’alarmes

Les archives d'alarmes contiennent toutes les informations liées à une alarme qui se produit dans le processus visualisé, y compris les données de configuration. Les archives permettent de consulter toutes les caractéristiques d'une alarme, telles que la classe d'alarme, l'horodatage et les textes de l'alarme. Une éventuelle modification des données de configuration d'une alarme entraîne la création d'un nouveau segment d'archive contenant les nouvelles données de configuration. Cette fonctionnalité empêche la modification de modifier des alarmes qui ont déjà été archivées avant la modification. Afin de sauvegarder une alarme, suivez les instructions de configuration suivantes :

- 1- **Définir les "Paramètres d'archives"** : détermination des paramètres pour l'archivage d'alarmes dans les paramètres d'archives.
- 2- **Activation de l'archivage d'alarmes** : décision d'archiver les alarmes de certaines classes d'alarmes.
- 3- **Affectation de l'alarme à la classe d'alarmes archivable.**

Les alarmes de la barquetteuse sont archivées dans la mémoire de l'automate, il suffit d'insérer une clé dans le port USB de l'IHM pour les récupérer sous la forme d'un fichier .CSV (Excel) qui est créé automatiquement.

La figure montre une archive des alarmes avec le temps de déclenchement et la date.

	A	B	C	D	E	F	G	H
1	Time_ms	MsgProc	StateAfter	MsgClass	MsgNum	TimeString	MsgText	PLC
2	3,87E+10	5	1	33	6	02.04.2024 10:17:32	ATTENTION_ABSENCE_DEPILAGE	HMI_Connection_1
3	3,87E+10	5	0	33	6	02.04.2024 10:17:34	ATTENTION_DEFAULT_PISTON	HMI_Connection_1
4	3,87E+10	5	1	33	6	02.04.2024 10:17:35	ATTENTION_ABSENCE_COLLE	HMI_Connection_1
5	3,87E+10	5	1	33	6	02.04.2024 10:17:36	ATTENTION_MANQUE_DECOUPE	HMI_Connection_1

Figure IV. 35 : Archivage des alarmes

IV.9 Conclusion

Dans ce chapitre, nous avons donné une description du WinCC et nous avons élaboré la configuration de l'écran tactile avec ce logiciel, nous avons réalisé la supervision des GRAFCETs de notre barquetteuse, nous avons aussi créé et configuré des alarmes, cette opération va servir à remplacer le pupitre de manœuvre à entrées physiques.

Cette amélioration offre à l'opérateur de la barquetteuse une utilisation meilleure, et un meilleur suivi de l'évolution du processus.

Le travail présenté dans ce projet de fin d'étude est une réponse à un besoin exprimé par l'entreprise CEVITAL au niveau du pôle de sucre. Le but est de réaliser une migration d'automate obsolète Schneider TSX Micro vers l'automate SIEMENS S7 1200 qui est nouveau doté de plus de puissance et de performance, puis élaborer une interface homme machine (IHM) qui permet de commander et visualiser le fonctionnement de la barquetteuse.

Pour cela, nous avons présenté d'une manière générale l'entreprise CEVITAL, ainsi que les différentes parties de la cartonneuse, une étape très importante ce pour la compréhension du fonctionnement et de l'élaboration de son cahier de charges.

Dans le deuxième chapitre nous avons abordé notre travail par la présentation matérielles et logiciels des deux types d'automate Schneider TSX Micro et SIEMENS S7 1200 (CPU 1214C DC/DC/Rly) ce qui nous a facilité de conclure l'importance de la migration effectuée et la partie programmation.

Ensuite, nous avons élaboré le cahier de charges de la barquetteuse et son GRAFCET, par la suite on a traduit ce GRAFCET en langage LADDER qui est compatible avec l'automate S7 1200.

Finalement, pour tester notre programme et visualiser le fonctionnement de la machine nous avons configuré une interface Comfort TP-1200.

Ce travail nous a permis d'atteindre plusieurs conclusions :

- Pour que l'entreprise puisse être compétitive elle doit suivre les révolutions technologiques, l'automatisation des machines dans le domaine industriel est indispensable.
- Les APIs sont des éléments principaux dans le milieu industriel.
- Pour les besoins des industries de points de vue : fiabilité, performance, et disponibilité des pièces de rechange, il est nécessaire de changer l'automate TSX Micro par un autre automate S7-1200, plus performant.
- Le logiciel TIA PORTAL offre une fiabilité et flexibilité pour le fonctionnement de l'automate S7-1200 et son interface Comfort TP-1200.

Par ailleurs, au cours de ce travail nous avons préalablement tracé quelques objectifs à atteindre ce qui fut notre cas, en effet pendant le stage pratique au sein de l'entreprise CEVITAL

nous avons su configurer un écran tactile qui facilite la tâche de l'intervention et la gestion de l'opérateur.

De plus, une amélioration des performances de la machine a été portée en ajoutant des alarmes permettant de détecter les manques ou les problèmes survenus lors du fonctionnement de la barquetteuse.

Ce projet nous a permis de nous familiariser avec les automates programmables Schneider TSX Micro et SIEMENS S7-1200 ainsi que leurs langages de programmation qui obéissent à des règles spécifiques.

En vue de perspectives nous proposons quelques recommandations :

- Remplacer le vérin simple effet au niveau de margeur par un vérin à double effet ou par un système électrique à double action afin d'optimiser la vitesse de margeur et d'augmenter sa flexibilité.
- Equiper la barquetteuse par un vérin à double effet au niveau de la station de formation de caisse au lieu de moteur électrique à un seul sens avec bielle-manivelle ce qui permet l'amélioration de rendement et la productivité de la cartonneuse.
- Changer le capteur à contact chargé de la détection de la caisse formée par un capteur photoélectrique pour réduire les coûts d'échange des capteurs à contact sachant que ces derniers s'usent.

Cette expérience nous a offert une occasion d'améliorer nos connaissances, bénéficier d'un avant-goût sur le monde industriel de l'automatisation des processus, et cela influe sur le choix de nos carrières tant qu'ingénieurs du futur dans l'automatique. On a certainement appris des tas de choses à travers ce projet.

Références bibliographiques

- [1] *CeVital Group Presentation French 2013*. (2013, 6 octobre). [Diapositives]. Récupéré le 04 mars 2024 de SlideShare. <https://fr.slideshare.net/GroupeCevital/cevital-group-french-2013v2-26909207>
- [2] *Situation géographique - Briefing sur l'entreprise Cevital*. (s. d.). Récupéré le 04 mars 2024 de <https://123dok.net/article/situation-g%C3%A9ographique-briefing-sur-l-entreprise-cevital.y4wdr5x5>.
- [3] Béranger, D. (2018). *Machines électriques : Principes de base et applications industrielles* (p. 78). Dunod.
- [4] Total, 2007. *Les équipements générateurs et moteurs électriques support de formation cours EXP-PR-EQ150-FR Révision 0.2 Exploration* (p. 117)
- [5] Charpentier, A. (2016). *Électrotechnique : Principes et applications* (p. 215). Ellipses.
- [6] Boudouda, A. (2019). *Cours schémas et appareillages électriques* (Vol. 32). Université de Boumerdes.
- [7] Courtois, M. (2010). *Électrotechnique et automatismes industriels* (p. 123). Dunod.
- [8] Fouquet, J.-L. (2014). *Appareillage électrique et automatismes* (p. 102). Dunod.
- [9] Robert, F. (2015). *Techniques de l'ingénieur : Pneumatique industrielle* (p. 87). Techniques de l'ingénieur.
- [10] Bourdon, C. (2020). *Pompes à vide: Fonctionnement, choix et applications*. Dunod
- [11] Sitelec. (s.d.). Détecteurs. Récupéré le 06 mars 2024 de <https://sitelec.org/cours/hu/detecteurs.htm#position>
- [12] Ménard, L. (2018). *L'histoire des interfaces de programmation (API)*. Paris, France : Presses Universitaires de France.
- [13] Formation Automatismes. (s.d.). *Matériel niveau 1*. Solutions Industrielles. Récupéré le 15 mars 2024 de https://tiaportal_formation_automatismes.solutions-industrielles.com/livret/niveau-1/s001-materiel.html
- [14] Bertrand, M. (2010). *Automates programmables industriels*. Techniques de l'Ingénieur
- [15] IEC. (s.d.). *La norme IEC 61131-3 : Concepts et notions de base, langages de programmation*. Récupéré le 15 mars 2024 de <https://site-web-du-document>
- [16] Dupont, A. (2019). *Les avantages des interfaces de programmation (API)*. Paris, France : Presses Universitaires de France.
- [17] Granville, V., & Roques, A. (2021). *Concevoir des API RESTful: Principes, technologies et bonnes pratiques*. Editions Eyrolles.

- [18] Schneider Electric. (2024). *PLC - Contrôleurs logiques programmables*. Consulté le [06/03/2024], sur <https://www.se.com/fr/fr/work/products/product-launch/guides/plc.jsp>
- [19] TSX Micro main manual. (s.d.). Récupéré le 10 mars 2024, sur https://pneumatykanet.pl/pub/przekierowanie/TSX_Micro_main_manual.pdf
- [20] Bertrand, M. (Décembre 2010). *Automates programmables industriels : Tome 5, Logiciel PL7 PRO*. Techniques de l'ingénieur.
- [21] SIEMENS. (2013). *Contrôleurs SIMATIC*. SIMATIC.
- [22] Automation24. (s.d.). *Siemens CPU 1214C - 6ES7214-1AG40-0XB0*. Récupéré le 10 mars 2024, de <https://www.automation24.fr/siemens-cpu-1214c-6es7214-1ag40-0xb0>
- [23] SIEMENS, « Fiche technique 6ES7214-1HG40-0XB0 », SIMATIC 2017.
- [24] Document de formation pour une solution complète d'automatisation TIA : MODULE A4 : Programmation de la CPU 315-2DP. Siemens A/D coopérâtes with éducation : 5/2004
- [25] Siemens, « SIMATIC S7 Automate programmable, Manuel système », livre, 2011
- [26] Witzmann, C., & Dufour, F. (2023). *Maîtriser TIA Portal V17 : Programmation, communication et supervision* (p.453). Éditions Dunod.
- [27] Mme.BELLAHSENE,2022. Support de cours Programmation avancée des API
- [28] *Obsolescence technologique de TSX Micro*. (s. d.). Récupéré le 07 mars 2024 de <https://www.se.com/fr/fr/product-range/540-modicon-tsx-micro/>
- [29] hehp tia portal
- [30] « Plate-forme d'automatisme Modicon TSX Micro » par télémécanique
- [31] Société Augustin Thierry. (2018). *Utilisation de PL7 PRO*. Récupéré le 10 mars 2024 de https://augustin-thierry.fr/wp-content/uploads/2018/06/utilisation_de_pl7pro.pdf

Annexes

**Annexe A : Généralités sur l'automate
TSX Micro et logiciel de programmation
PL7 PRO**

A.1 Les composants d'automate TSX Micro

La figure suivante représente les différents composants de l'automate TSX Micro :

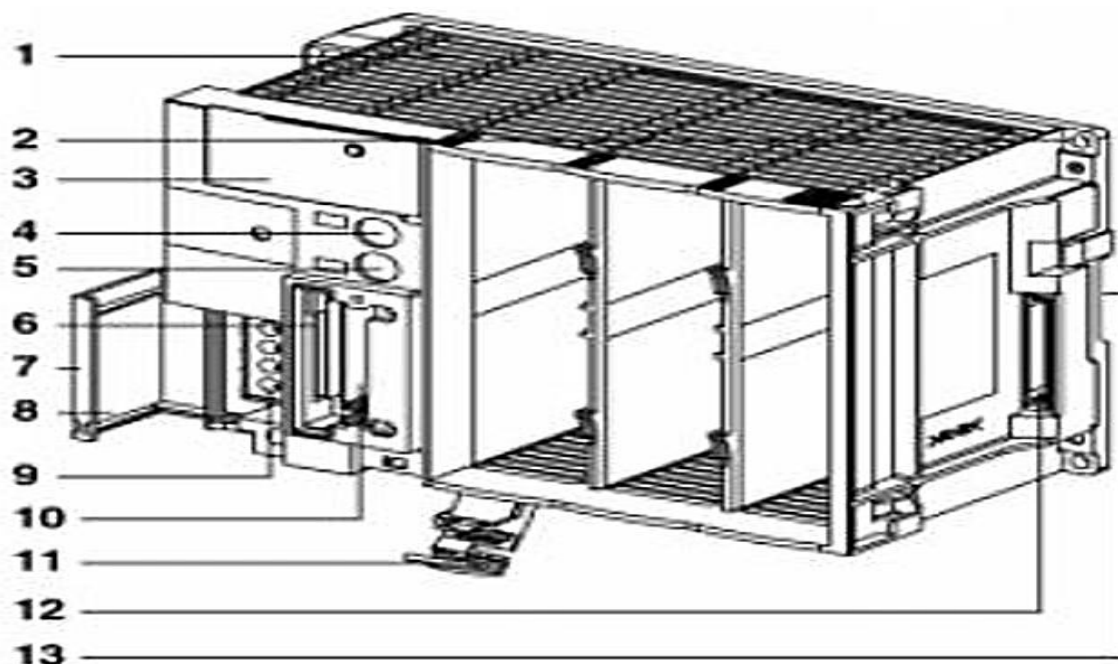


Figure A. 1 : Les composants d'automate TSX Micro

Tableau A. 1 : Les composants d'automate TSX Micro

Repère	Description
1	Bac à 3 emplacements, intégrant l'alimentation, le processeur et sa mémoire de base.
2	Trou de fixation de l'automate.
3	Bloc de visualisation centralisée.
4	Prise terminal TER
5	Prise de dialogue opérateur AUX.
6	Emplacement pour une carte d'extension mémoire. En l'absence de carte, cet emplacement est équipé d'un cache qu'il est obligatoire de maintenir en place; son extraction provoquant l'arrêt de l'automate.
7	Trappe d'accès aux bornes d'alimentation.
8	Étiquette à renseigner pour le changement de la pile.
9	Bornes d'alimentation
10	Emplacement pour un coupleur de communication.
11	Trappe d'accès à la pile optionnelle et au commutateur de protection en écriture du système d'exploitation.
12	Connecteur de raccordement du mini-bac d'extension, protégé de base par un cache amovible.
13	Dispositif pour montage sur profilé DIN.

A.2 Les langages de programmation sur PL7 PRO

Le PL7 pro est le logiciel qui permet la conception et la mise en œuvre des applications pour automates Micro tel que le Micro TSX3721 actuellement équipé sur la barquetteuse. [30]
Le logiciel PL7 propose 4 langages de programmation :

- **Langage à contacts**

Le langage à contacts (LD) est un langage graphique. Il permet la transcription de schémas à relais, il est adapté au traitement combinatoire. Il offre les symboles graphiques de base : contacts, bobines, blocs. L'écriture de calculs numériques est possible à l'intérieur de blocs opération [30].

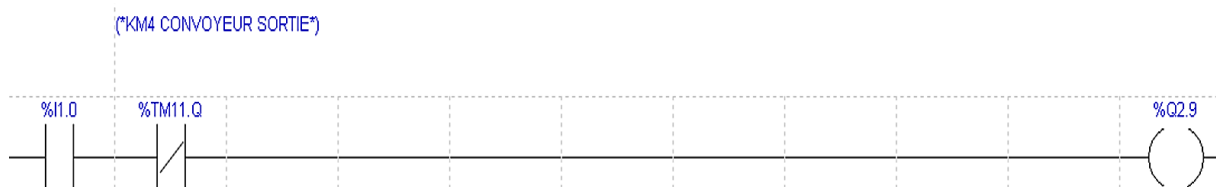


Figure A. 2 : Exemple d'un réseau en langage à contact

- **Liste d'instructions**

Le langage liste d'instructions (IL) est un langage "machine" booléen qui permet l'écriture de traitements logiques et numériques.

```
IL : MAST - SR1
! %L0 : LD      %I1.0
        ANDN   %M12
        OR (   %TM4.Q
        AND    %M17
        )
        AND    %I1.7
        ST     %Q2.5
! %L5 : LD      %I1.10
        ANDN   %Q2.3
        ANDN   %M27
        IN     %TM0
        LD     %TM0.Q
        AND    %M25
        AND    %MW0:X5
        [%MW15 := %MW18+500]
! %L10: LD      %I1.2
        AND    %I1.4
        SR2
```

Figure A. 3 : Exemple de programme en langage liste d'instructions

- **Littéral structuré**

Le langage littéral structuré (ST) est un langage de type "informatique" permettant l'écriture structurée de traitements logiques et numériques.

- **Grafcet**

Le langage Grafcet permet de représenter graphiquement et de façon structurée le fonctionnement d'un automatisme séquentiel.

A.3 Configuration du logiciel

- Pour concevoir une nouvelle application, accéder au logiciel « PL7-PRO » puis ouvrir un nouveau fichier :

A ce stade, il va falloir faire le choix du type de processeur associé à l'automate, ainsi que du type de programmation envisagé : - GRAFCET « Oui » : Programmation d'un Grafcet avec accès possible aux trois traitements : Prl, Chart et Post ; - GRAFCET « Non » : Programmation d'équations logiques type « Ladder » avec accès à un seul traitement : Main (traitement équivalent au Post).

Remarque :

Pour ouvrir un projet existant cliquez sur « Fichier », puis sur « Ouvrir ». Sélectionnez votre fichier (*.STX) dans son répertoire et cliquez sur « OK »

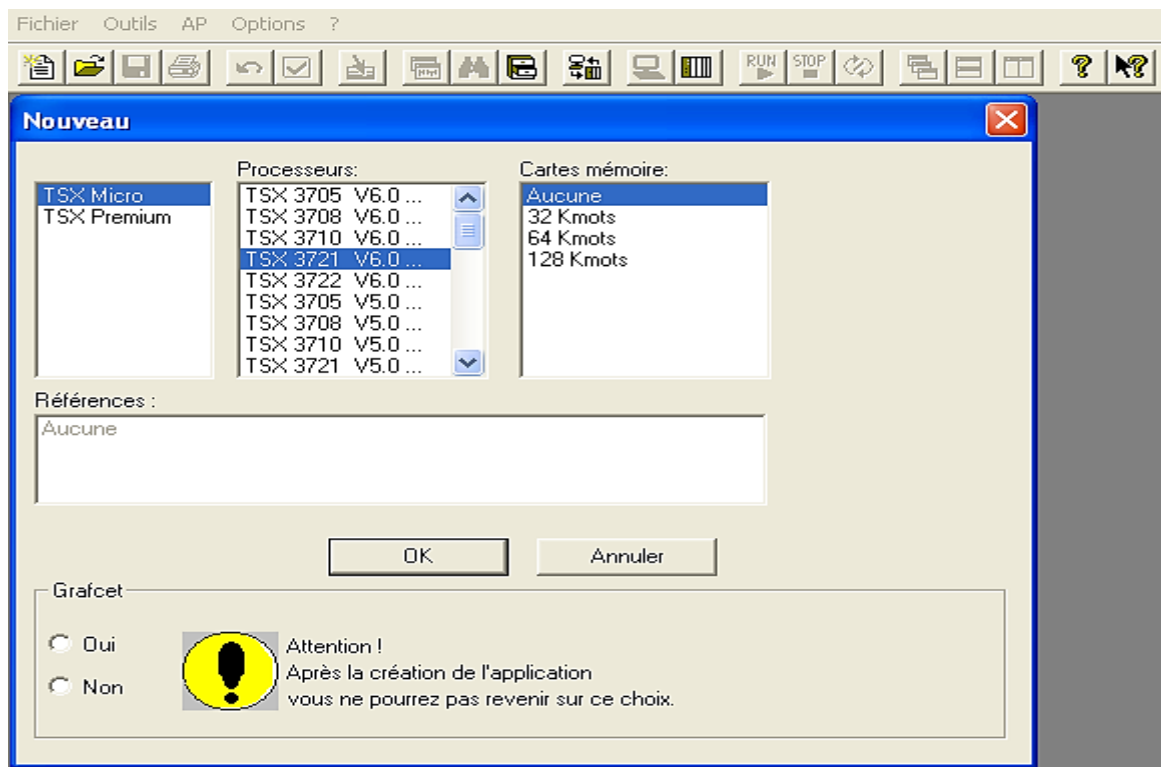


Figure A. 4 : Configuration logiciel

- Après avoir validé, on arrive au menu suivant :

Ce menu permet d'accéder entre autre : à une configuration matérielle nécessaire, aux différents traitements de programmation et à des fonctions complémentaires.

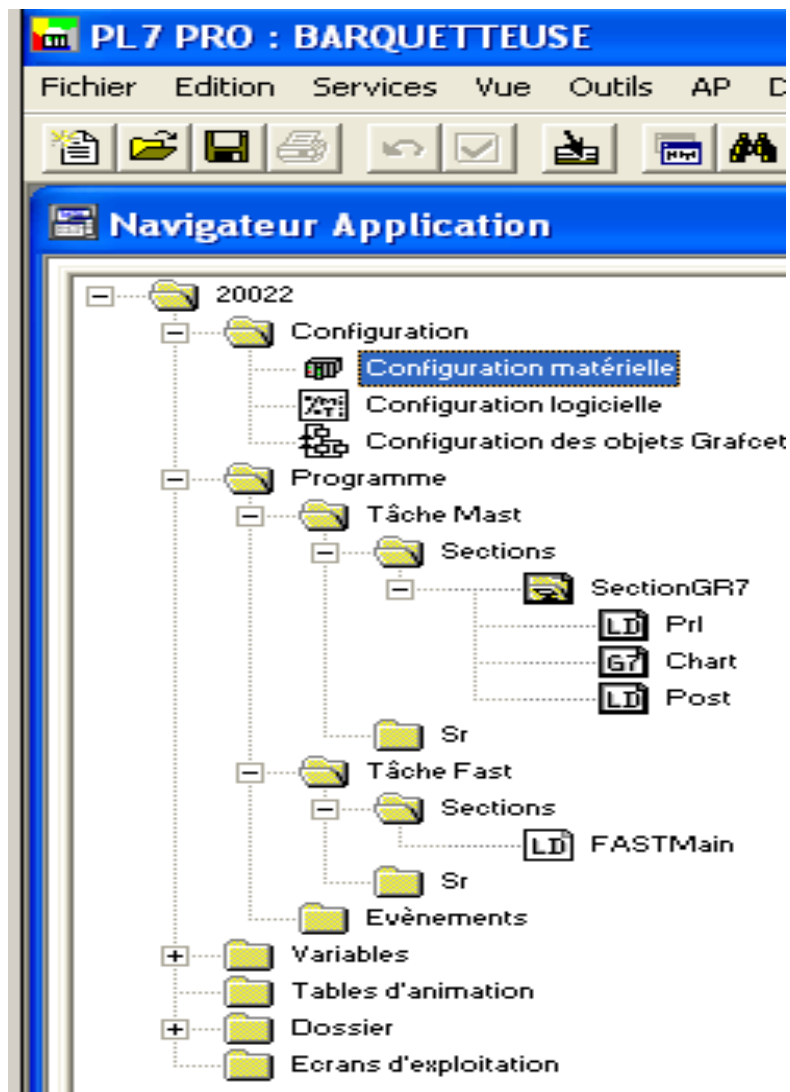


Figure A. 5 : Vue de menu

- **Configuration matérielle nécessaire**

Après avoir « cliqué » sur Configuration Matérielle, il va falloir déclarer toutes les cartes E/S associées au processeur de l'automate précédemment défini. Ces différentes cartes permettent à l'automate de gérer physiquement l'état des différents capteurs et préactionneurs du système.

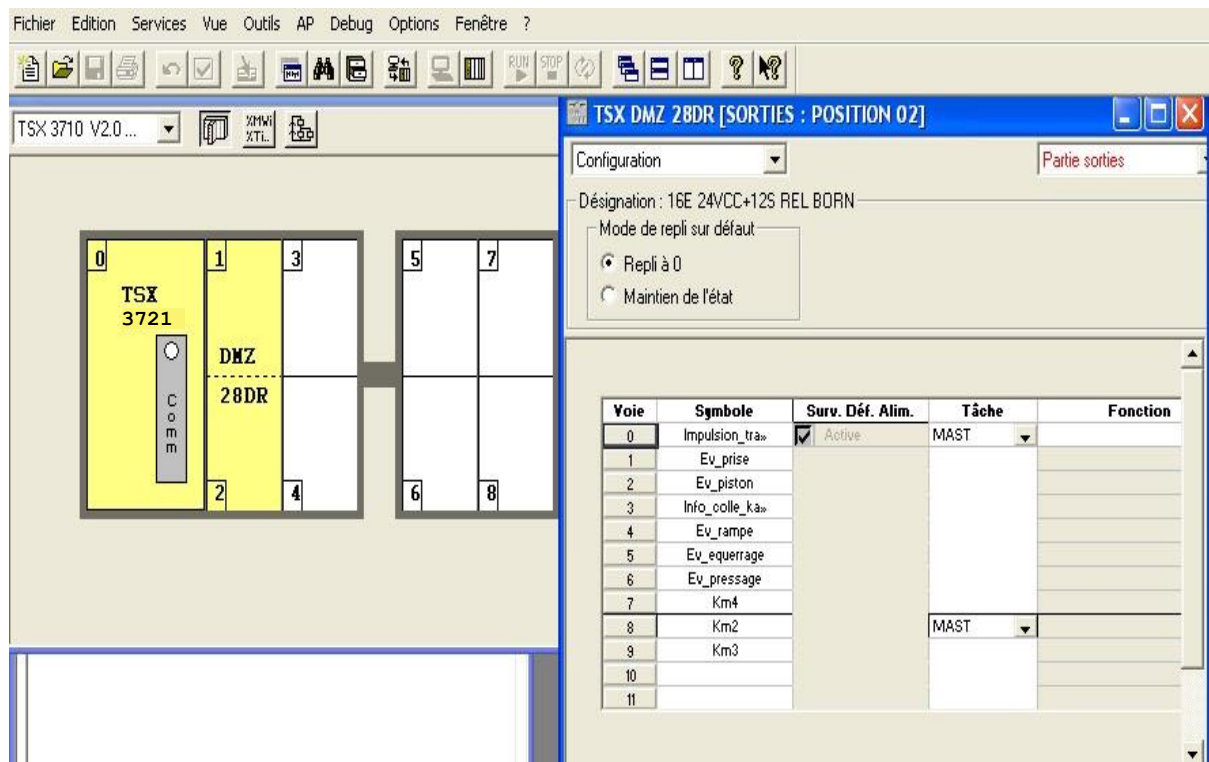


Figure A. 6 : Configuration matériel

- **Traitements de programmation :** Ouvrez l'arborescence jusqu'à « CHART ». Double cliquez sur "CHART" et dessinez votre grafcet en utilisant les icônes en bas de l'écran ou les touches F2 à F12 [31].

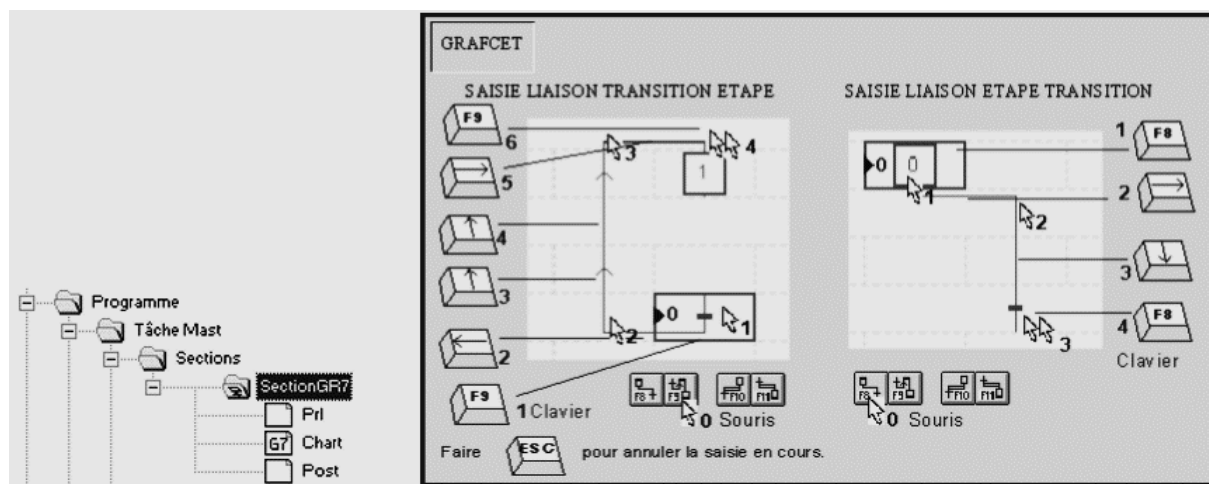


Figure A. 7 : Configuration de GRAFCET

Une fois votre grafcet fini (en rouge), cliquez sur l'icône "Valider". Le grafcet devient noir. Pour écrire une transition, il faut double cliquer sur la transition correspondante. Choisissez le langage LD et cliquez sur « OK ».

- **Traitement Postérieur « Post »** (en cas de Grafcet - Oui) ou « Main » (en cas de Grafcet - Non) : Dans cette page, on va gérer l'écriture des sorties.

Remarque : Dans un système séquentiel, différentes étapes peuvent être associées à une même action, il faut alors regrouper ces étapes, et les associer à l'action considérée !

Exemple :

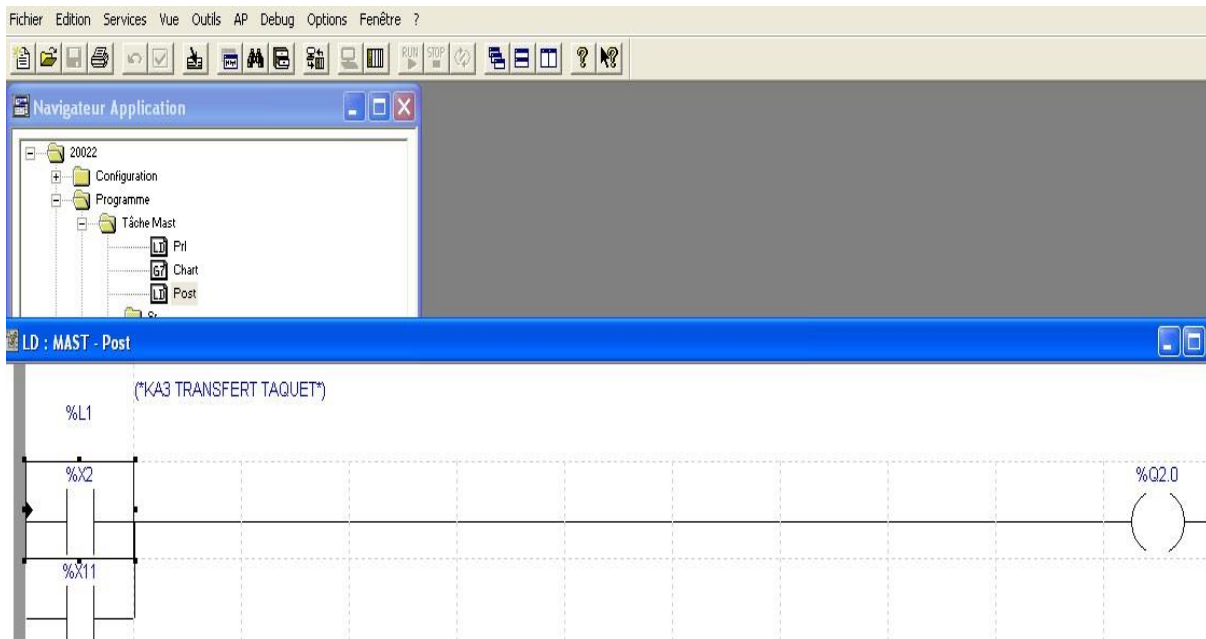
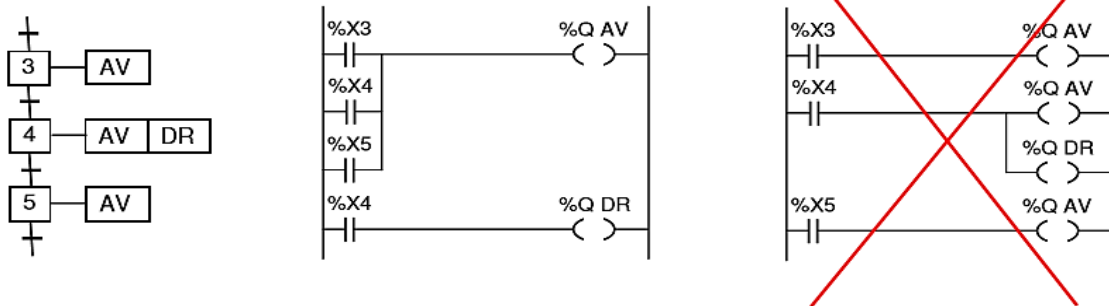


Figure A. 8 : Exemple d'un traitement Postérieur

- **Gestion d'une temporisation par bloc temporisateur**

Avantage : Traitement de toute durée de temporisation supérieure à 10ms.

Inconvénient : Il faut définir un bloc temporisateur en traitement postérieur et définir sa durée dans les paramètres de configuration (Fonctions complémentaires - FB prédéfinis).

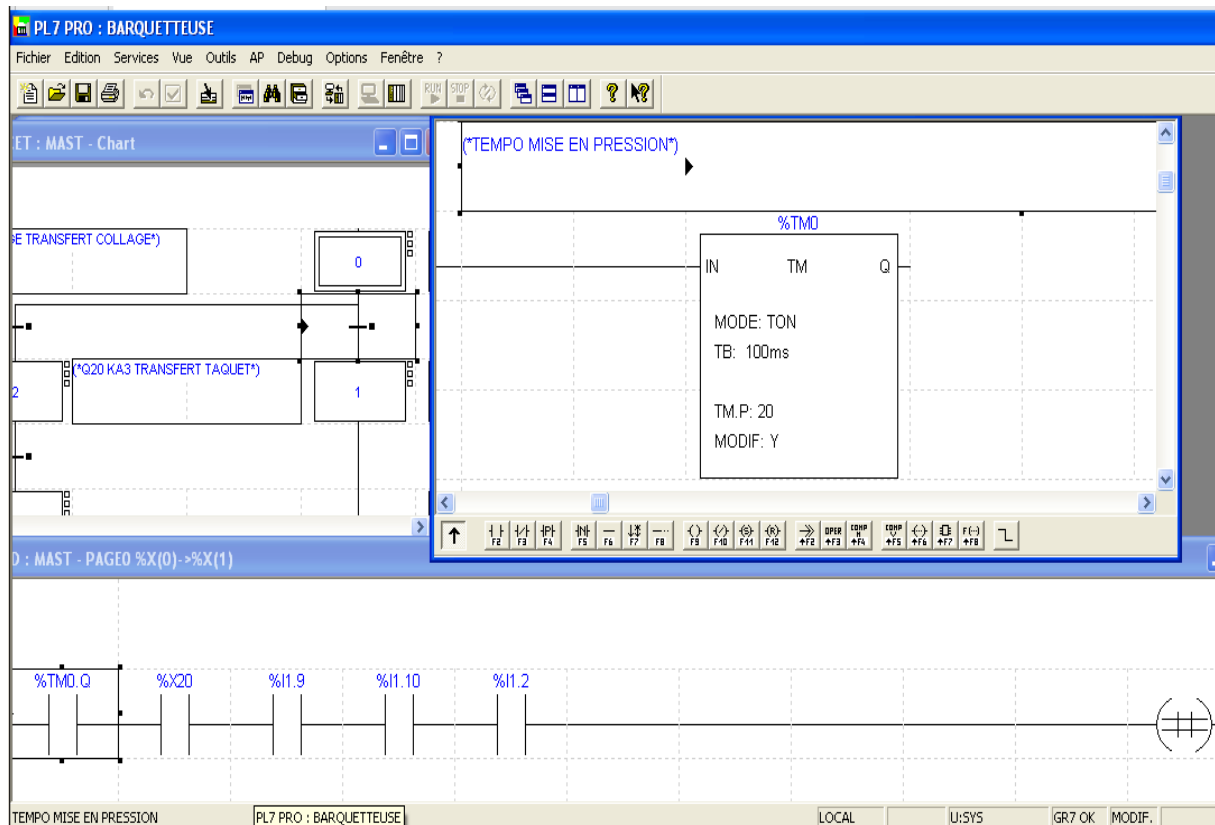


Figure A. 9 : Exemple de configuration d'une temporisation

➤ **Gestion des variables internes (FB prédéfinis) :**

Paramètres		FB PREDEFINIS		TM		Zone de saisie	
Repère	Type	Symbole	Preset	Mode	TB	Req	
%TM0	TM	Tempo_marche	20	TON	100 ms	✓	
%TM1	TM		20	TON	10 ms	✓	
%TM2	TM		200	TON	10 ms	✓	
%TM3	TM		40	TON	10 ms	✓	
%TM4	TM		70	TON	10 ms	✓	
%TM5	TM		5	TON	10 ms	✓	
%TM6	TM		10	TON	10 ms	✓	
%TM7	TM		20	TON	10 ms	✓	
%TM8	TM		300	TON	10 ms	✓	
%TM9	TM		60	TON	1 s	✓	
%TM10	TM		2	TON	1 s	✓	
%TM11	TM		9999	TON	1 mn	✓	

Figure A. 10 : Exemple de gestion de la durée des temporisateurs

➤ **Adressage des objets bits**

L'adressage des bits internes, système et étapes suit les règles suivantes :

%	M, S ou X	i
Symbole	Type d'objet	Numéro

➤ **Adressage des objets mots**

L'adressage des mots (hors mots de modules d'entrées/sorties, et blocs fonction) suivent une même syntaxe décrite ci-après :

%	M, k ou S	B, D, W ou F	i
Symbole	Type d'objet	Format	Numéro

Exemples :

%MW10 = mot interne simple longueur numéro 10 ; %MF20 = mot interne flottant numéro 20 ; %KD21 = double mot constant numéro 21 ; %SW5 = mot système numéro 5

➤ **Gestion des variables Entrées/Sorties (E/S) :**

Paramètres			
E/S		Adr. 1: TSX DMZ 28DR	
Repère	Type	Symbole	
xI1.0	EBOOL	Marche_générale	
xI1.1	EBOOL	Colle_chaude	
xI1.2	EBOOL	Arret_transfert	
xI1.3	EBOOL	Haut_prise	
xI1.4	EBOOL	Presence_depilage	
xI1.5	EBOOL	Presence_colle	
xI1.6	EBOOL	Piston_haut	
xI1.7	EBOOL	Piston_bas	
xI1.8	EBOOL	Demande_caisse	
xI1.9	EBOOL	Présence_decoupe	
xI1.10	EBOOL	Piston_pret	
xI1.11	EBOOL	Forçage_piston_bas	
xI1.12	EBOOL	Cellule_bourrage	
xI1.13	EBOOL	Forçage_piston_haut	

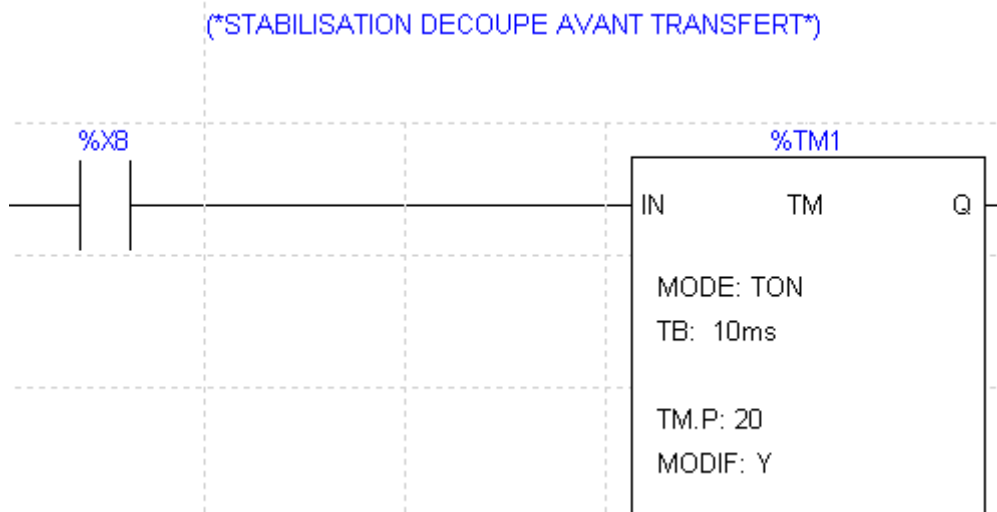
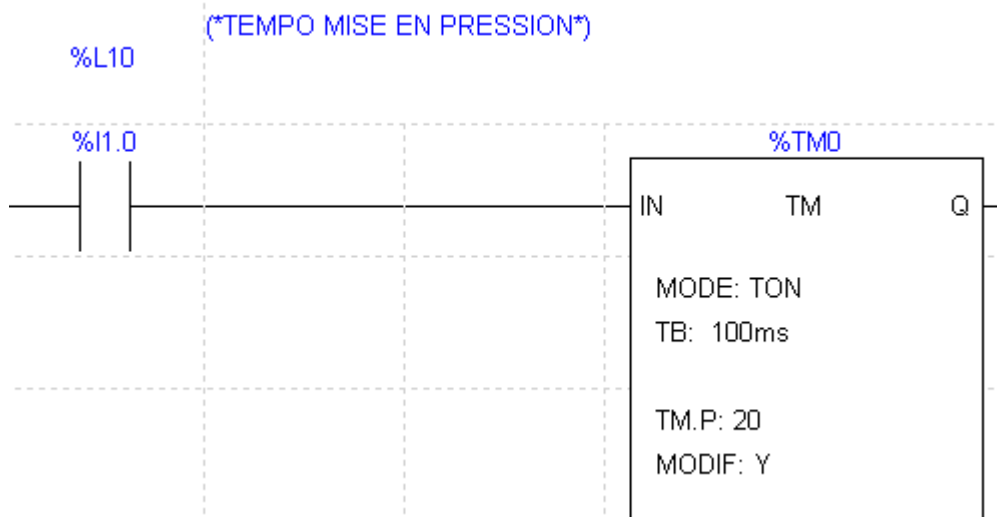
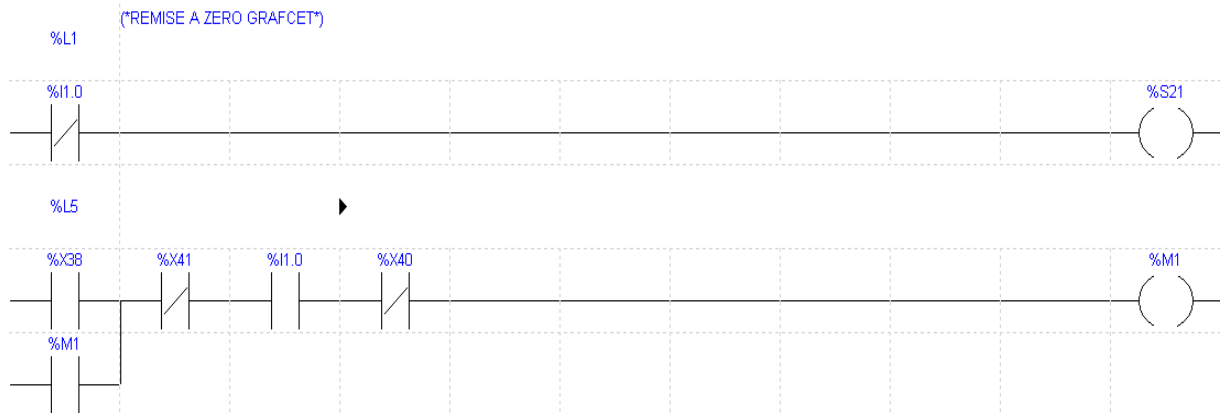
Figure A. 11 : Table des entrées

Paramètres			
E/S		Adr. 2: TSX DMZ 28DR	
Repère	Type	Symbole	
xQ2.0	EBOOL	Transfert_taquet	
xQ2.1	EBOOL	Ev_prise	
xQ2.2	EBOOL	Ev_piston	
xQ2.3	EBOOL	Ev_colle	
xQ2.4	EBOOL	Ev_rampe	
xQ2.5	EBOOL	Ev_equerrage	
xQ2.6	EBOOL	Ev_pressage	
xQ2.7	EBOOL	Moteur_piston	
xQ2.8	EBOOL	Forçage_piston_haut	
xQ2.9	EBOOL	Forçage_piston_bas	

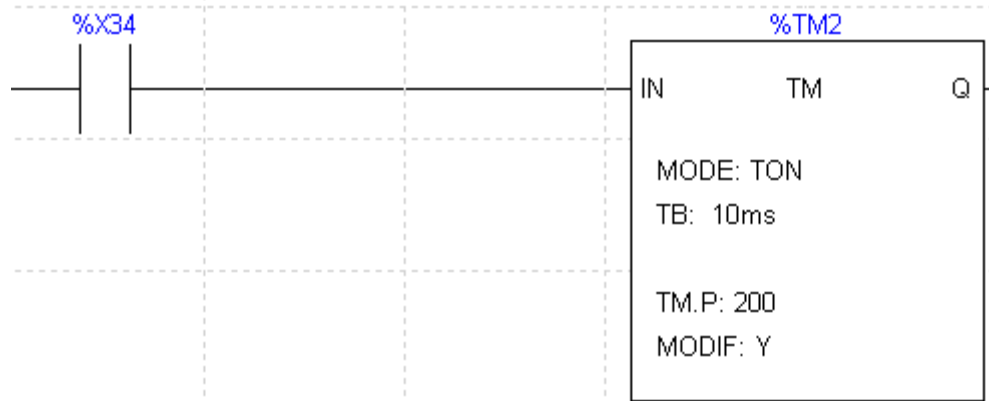
Figure A. 12 : Table des sorties

A.4 Réalisation des GRAFCETs de la barquetteuse

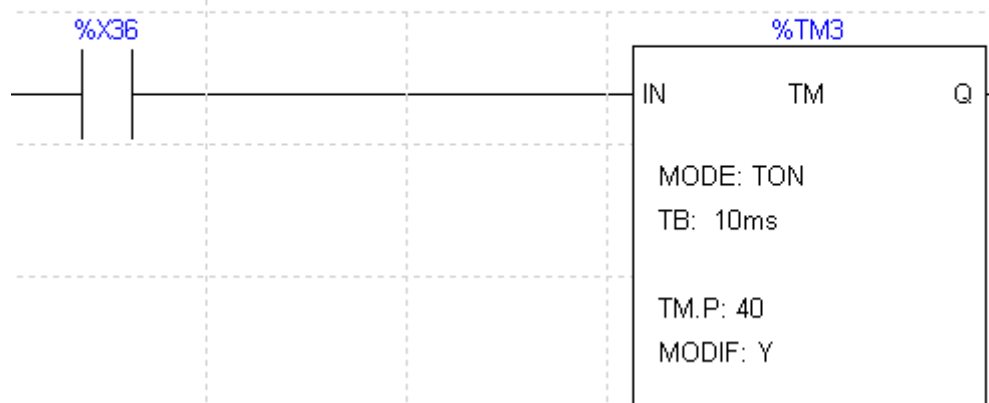
• Traitement préliminaire (PRL noté Pr1)



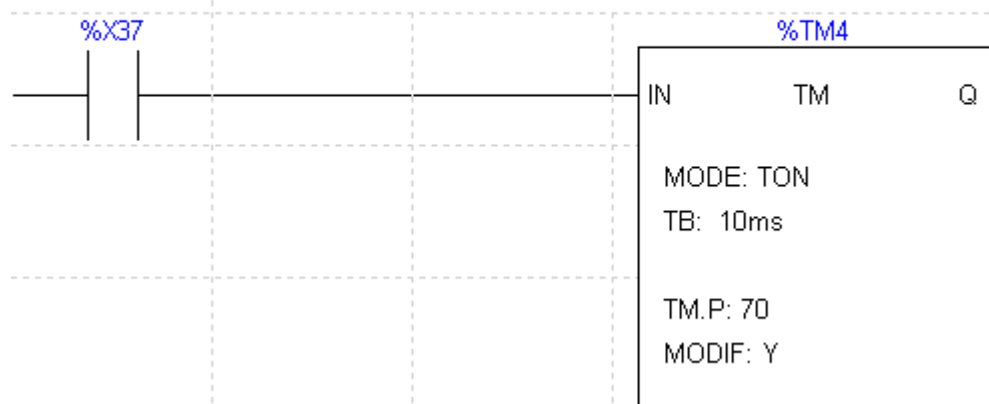
(*CONTROLE PISTON ARRIVE EN BAS*)



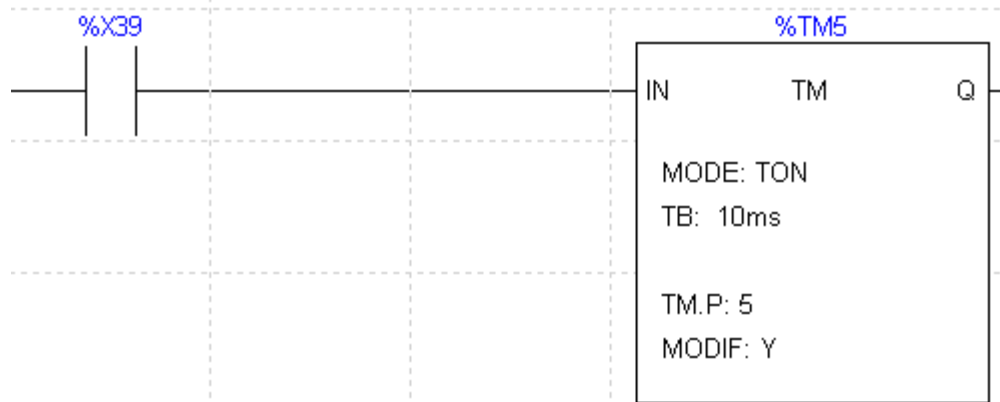
(*EQUERRAGE*)



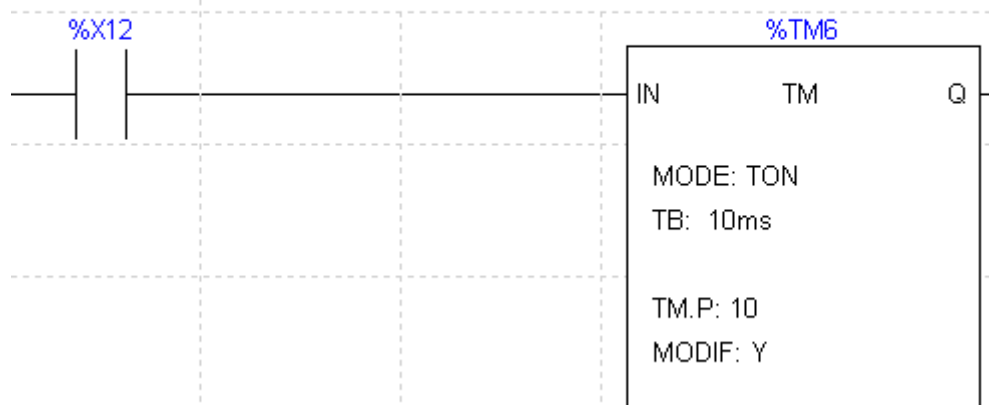
(*PRESSAGE*)



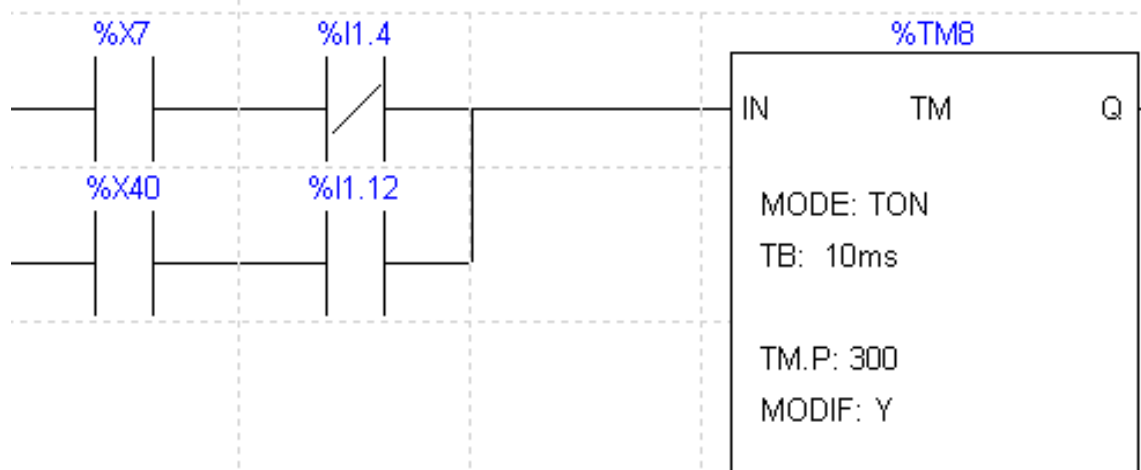
(*ANTICIPATION TRANSFERT*)



(*ANTICIPATION REMONTEE VERIN PRISE*)



(*DEFAULT PRISE DECOUPE DEFAULT DECOUPE PUIT*)



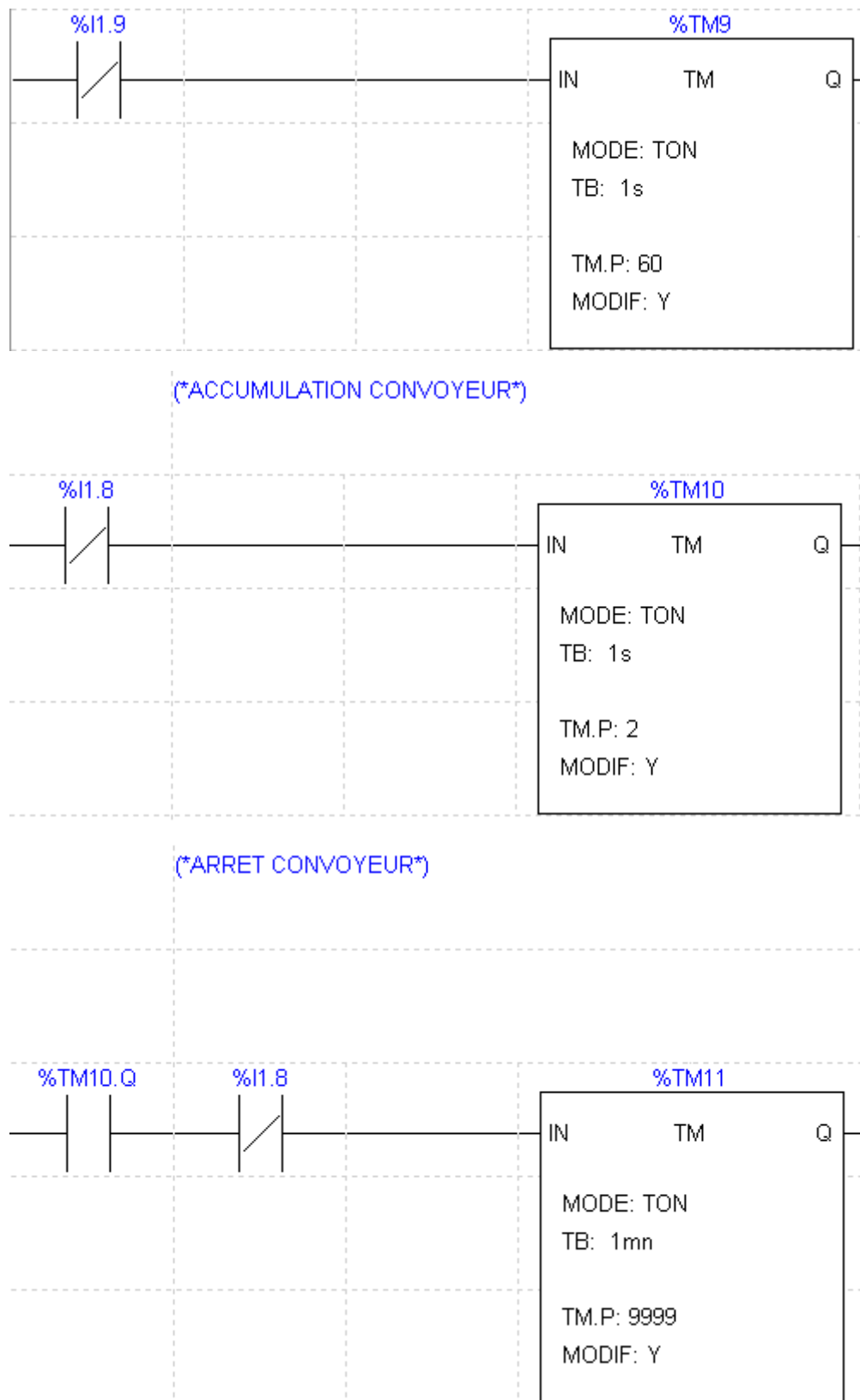
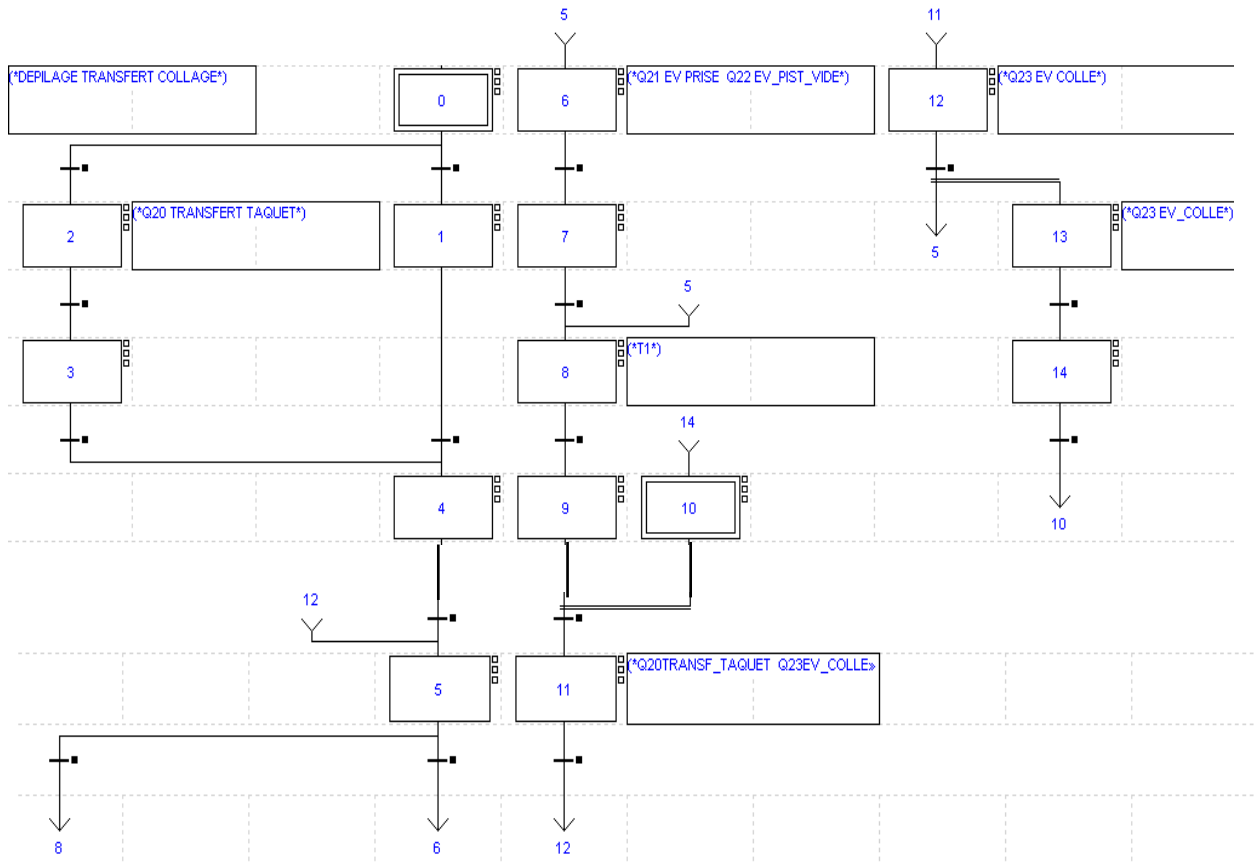


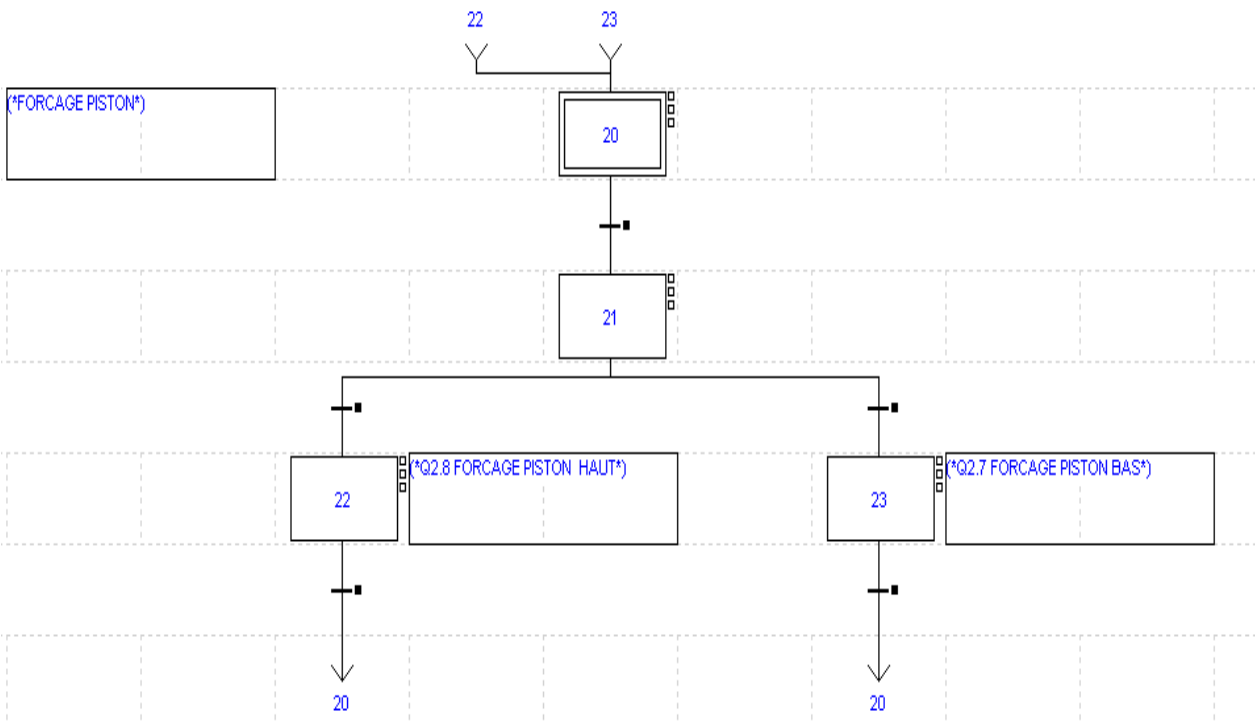
Figure A. 13 : La partie PRL

• **Traitement séquentiel (SEQ noté Chart)**

- **GRAFCET « DEPIPAGE_TRANSFERT_COLLAGE »**



- **GRAFCET « FORCAGE_PISTON »**



- GRAFCET « FORMATION_CAISSÉ »

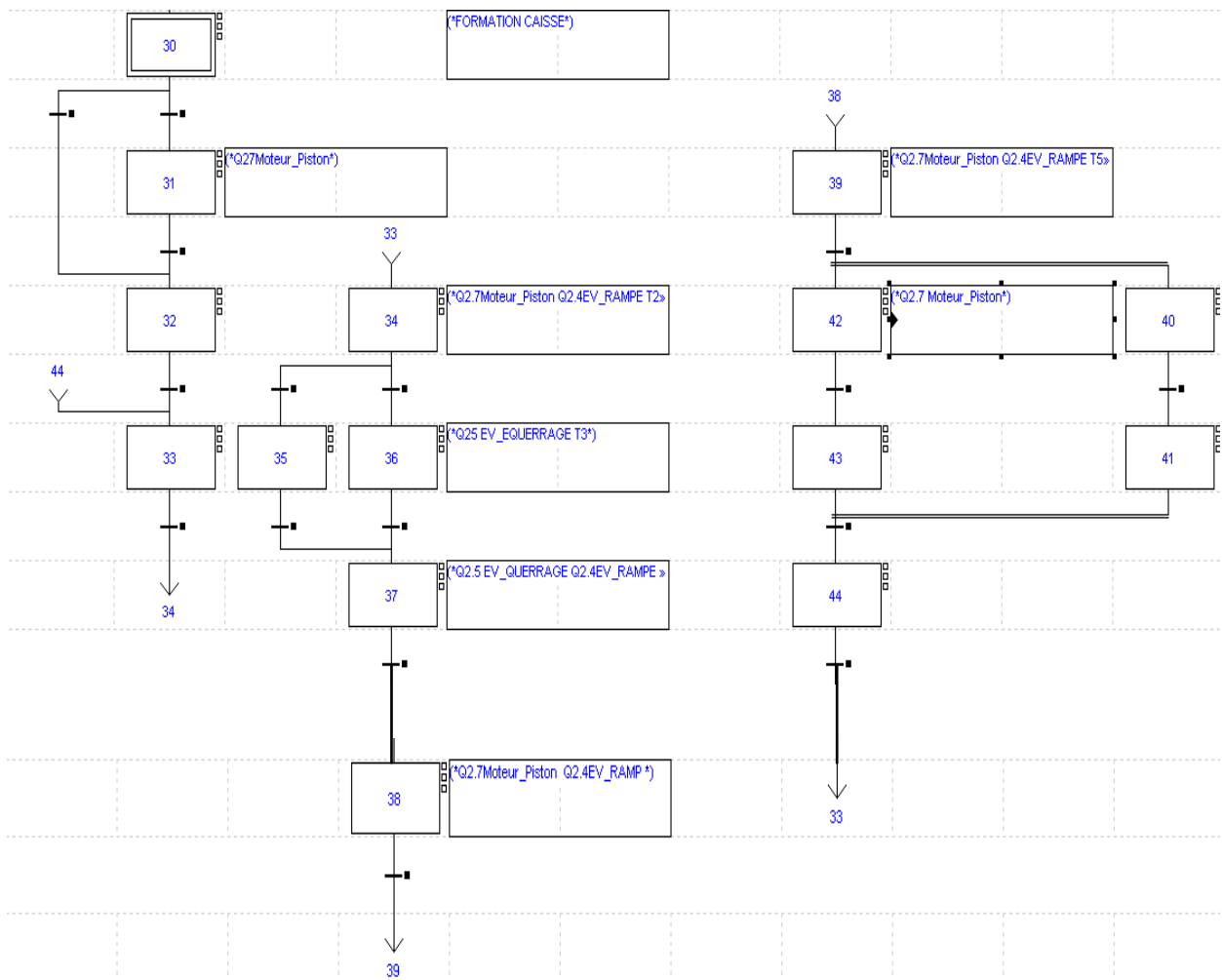
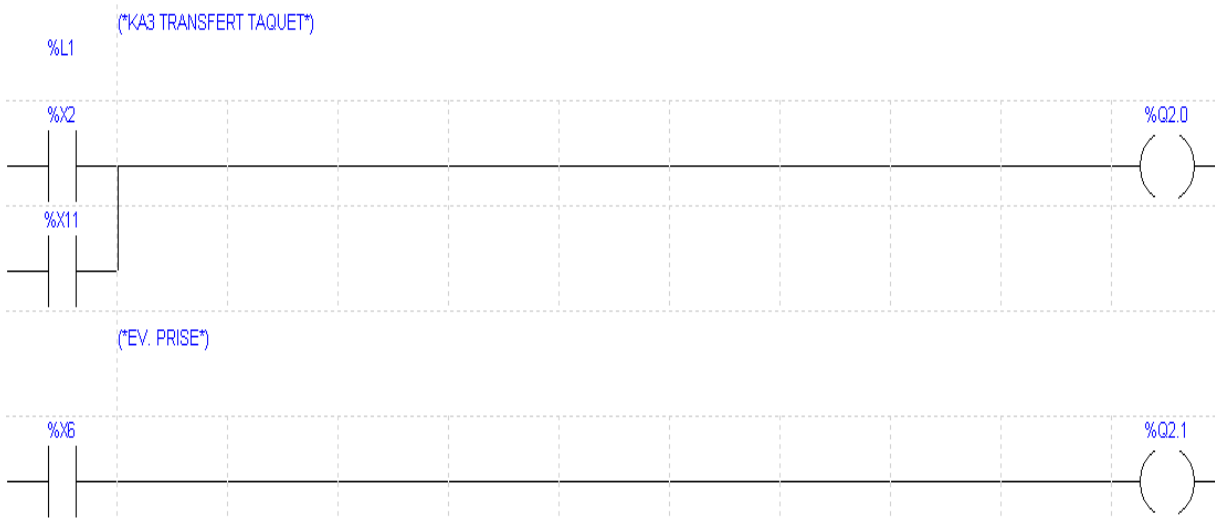
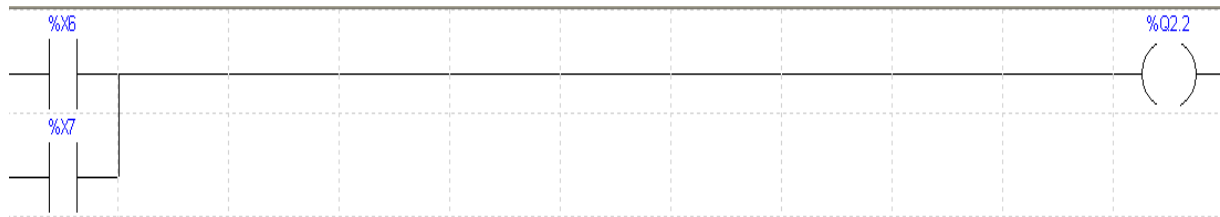


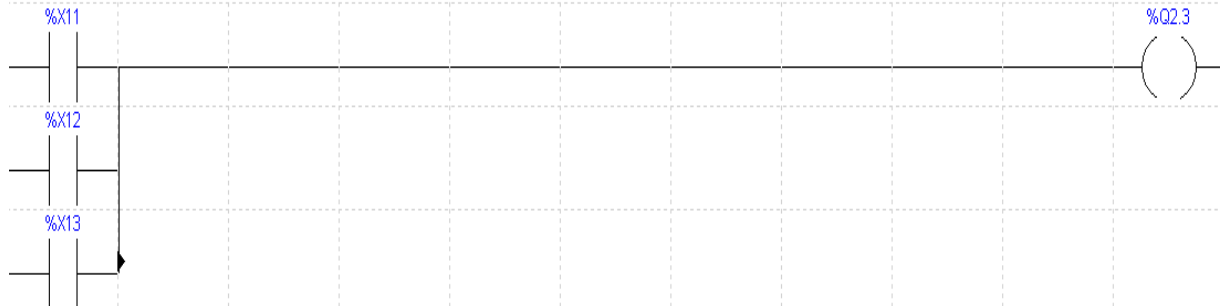
Figure A. 14 : La partie chart

- **Traitement postérieur (POS noté Post)**

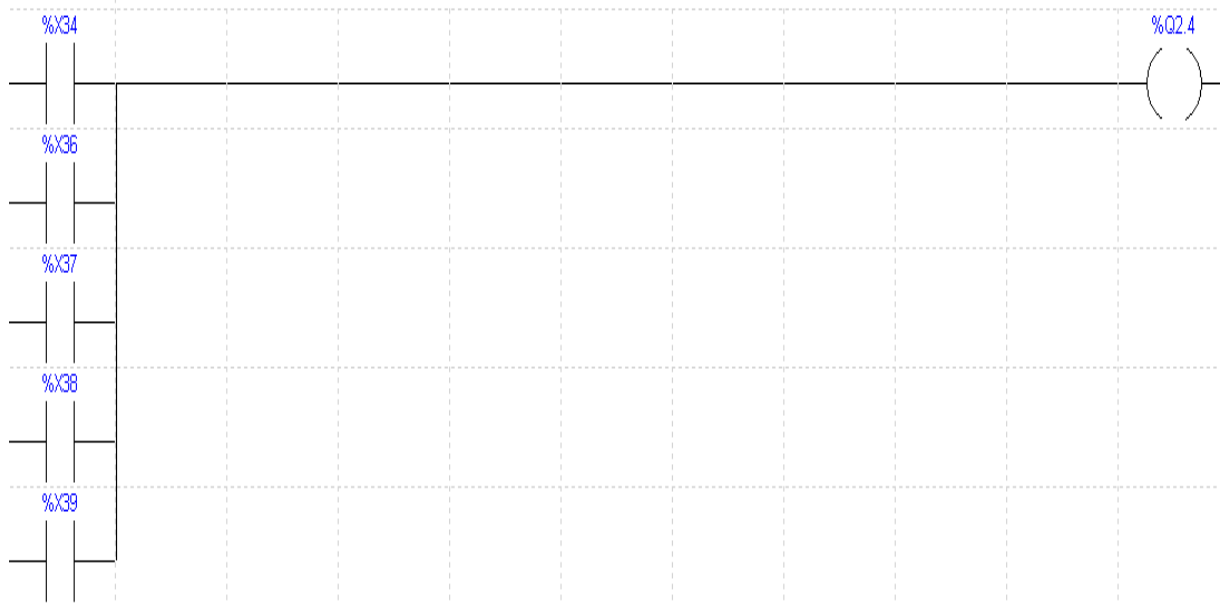




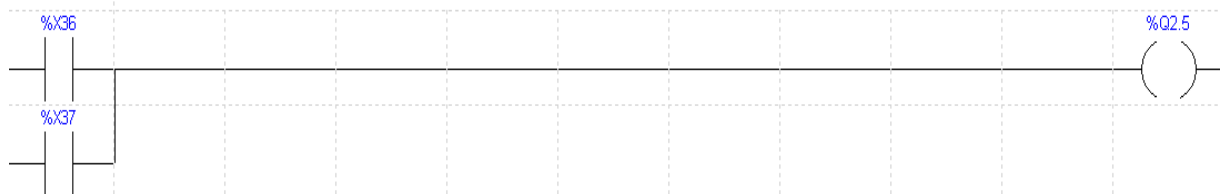
(*INFO COLLE KA10*)



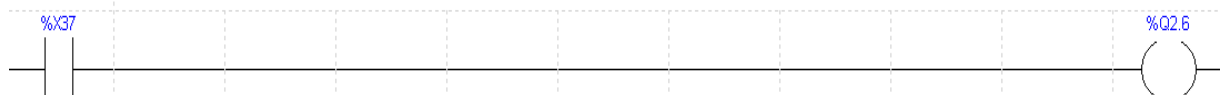
(*EV RAMPE*)

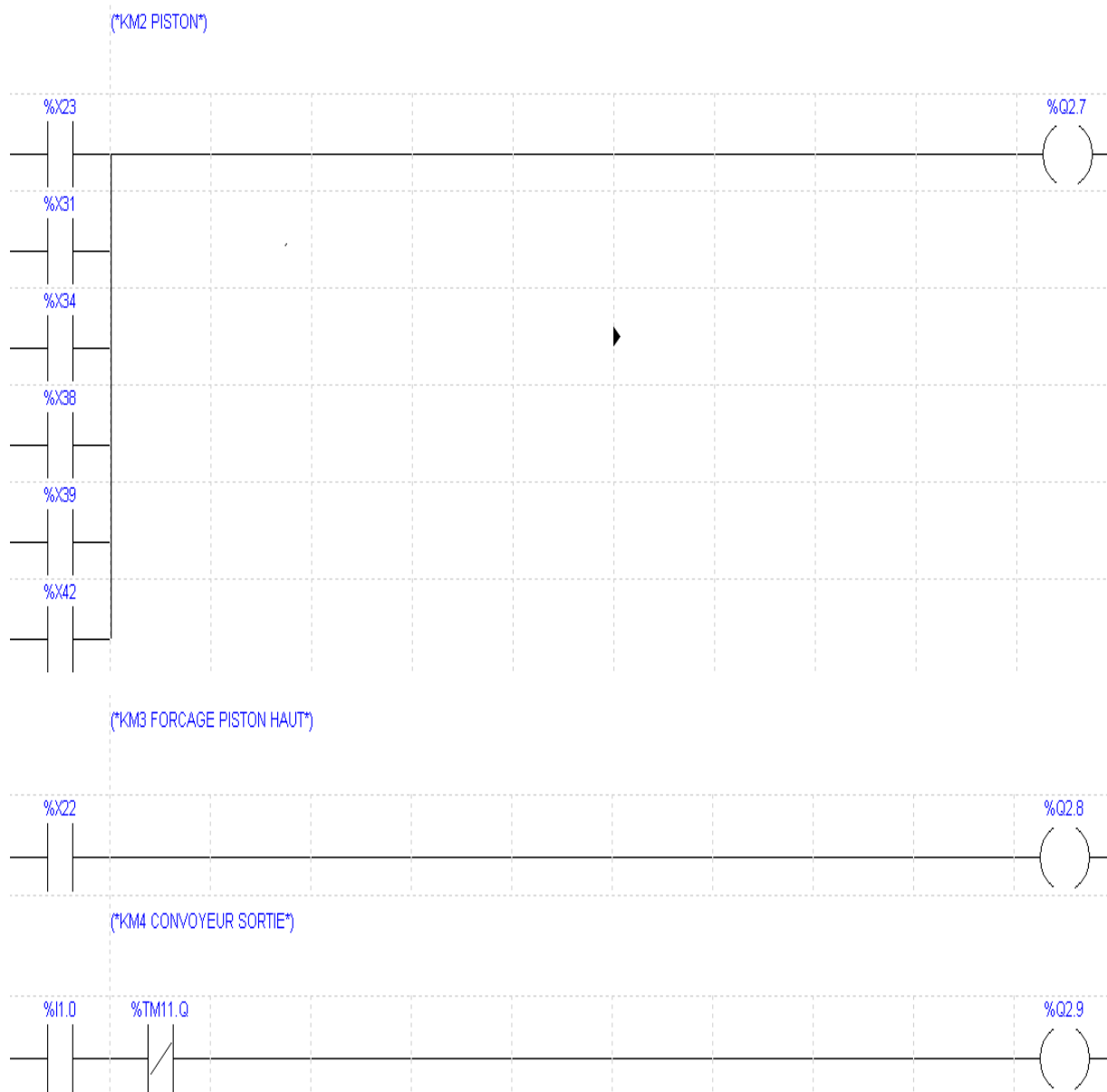


(*EQUERRAGE*)



(*PRESSAGE*)



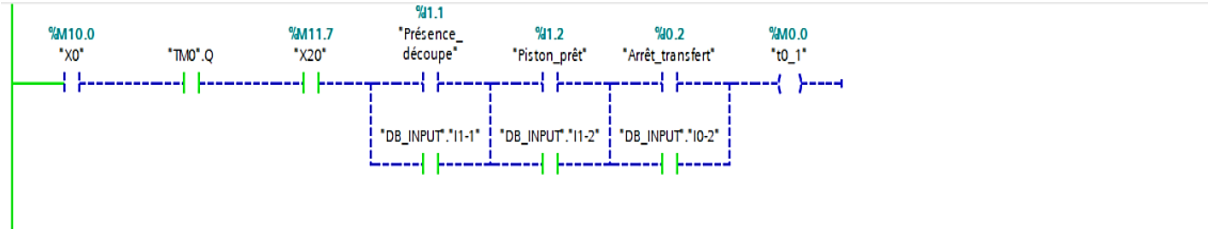
**Figure A. 15 : La partie POST**

**Annexe B : Le programme complet
en ladder sur TIA Portal**

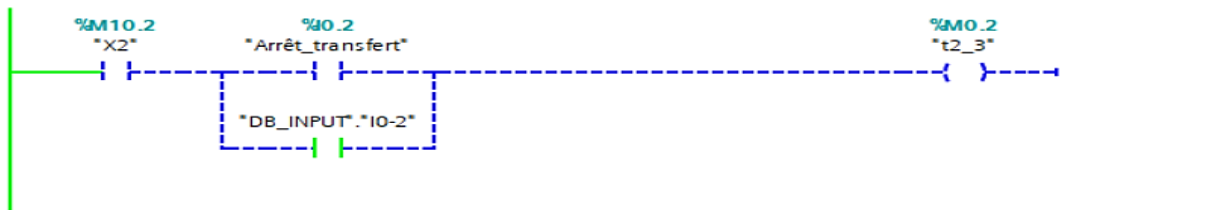
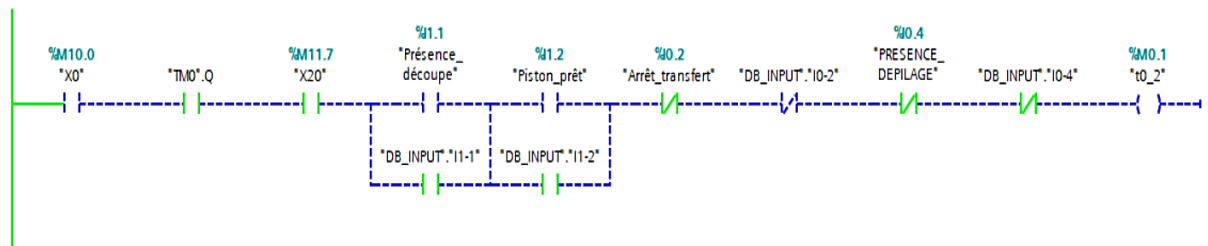
- **DEPILAGE_TRANSFERT_COLLAGE[FC1]**

Pour organiser notre programme et nos fonctions on a créé d'abord les réseaux des transitions, ensuite on créé les réseaux des étapes.

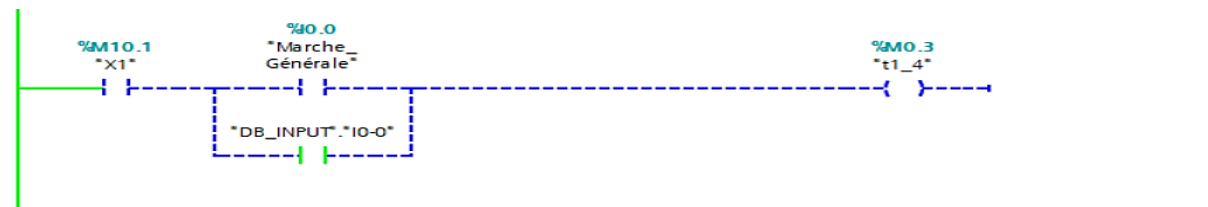
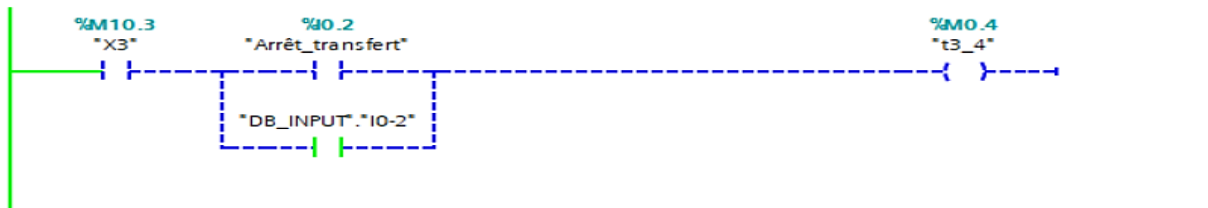
Les transitions :



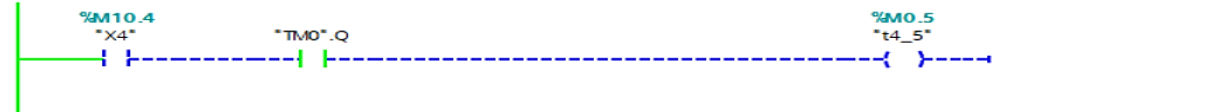
Réseau 2 : Les Transitions TR0-2
Commentaire

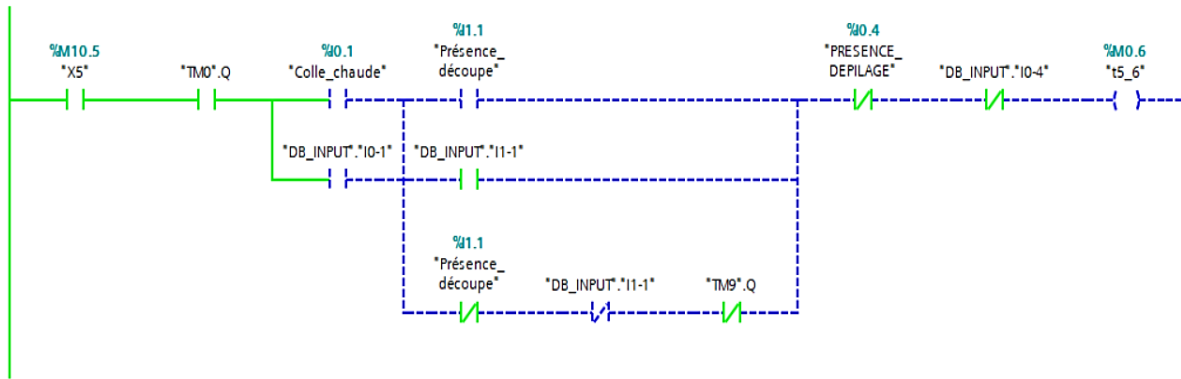


Réseau 4 : Les Transitions TR3-4
Commentaire



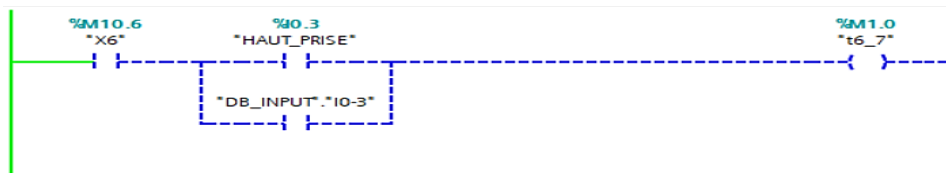
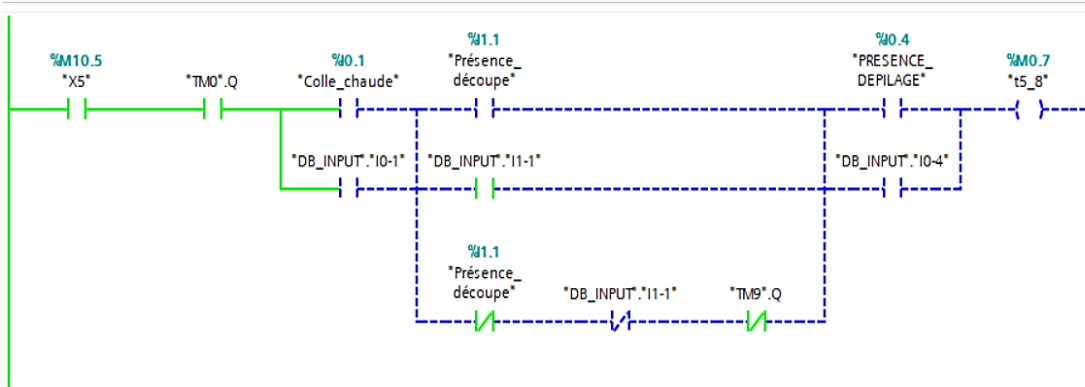
Réseau 6 : Les Transitions: TR4-5
Commentaire





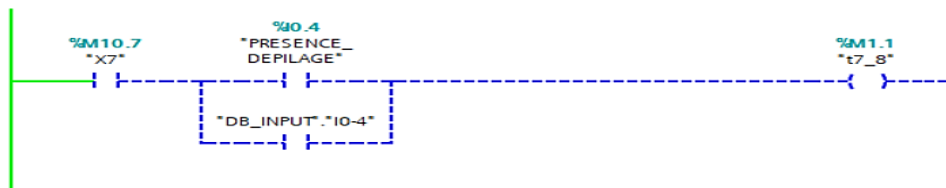
Réseau 8 : es Transitions: TR5-8

Q



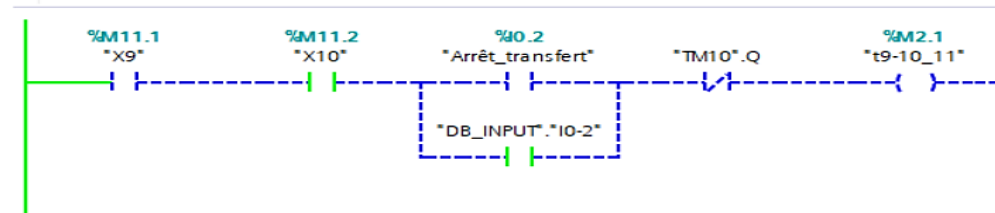
Réseau 10 :

Commentaire



Réseau 12 :

Commentaire



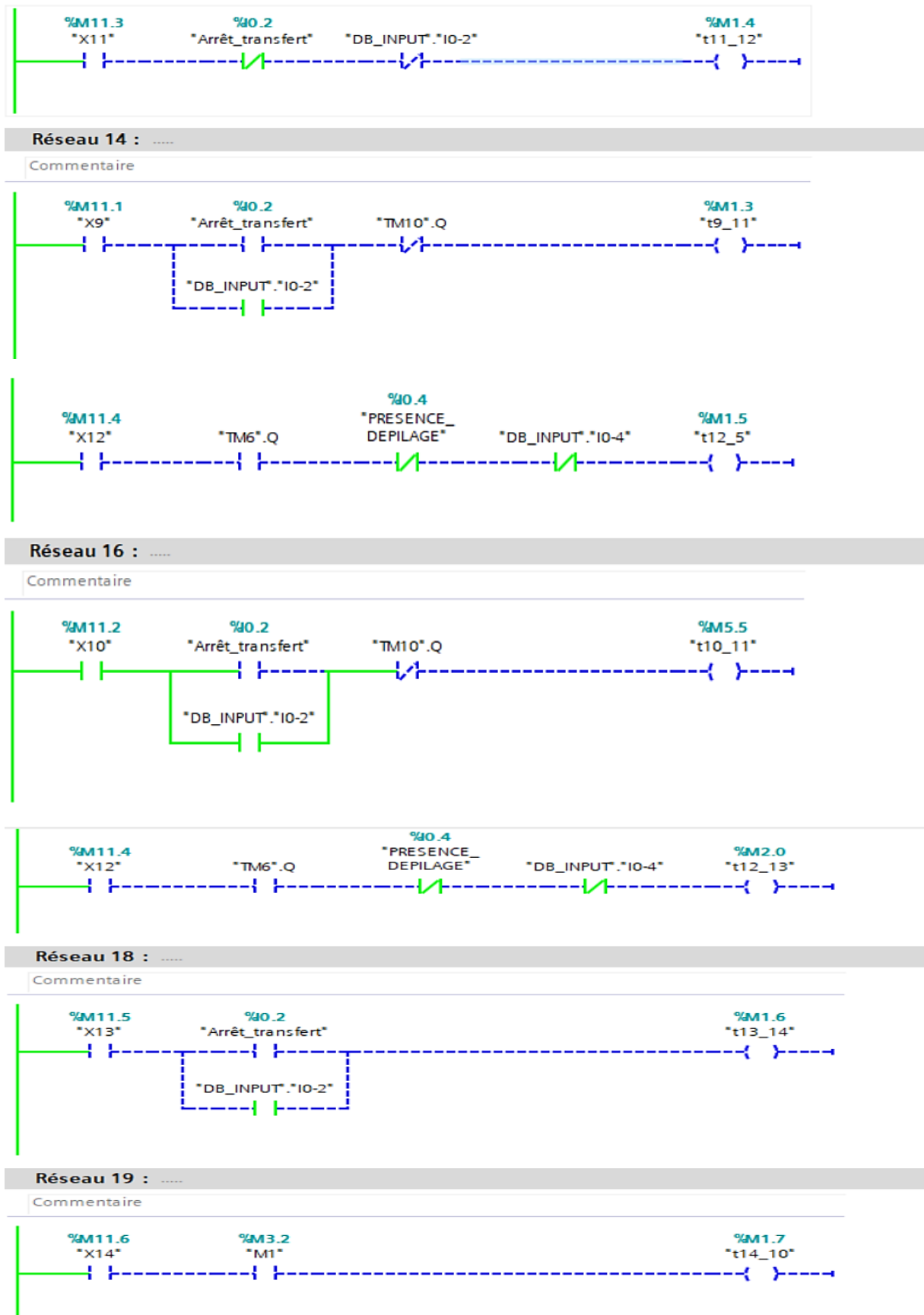
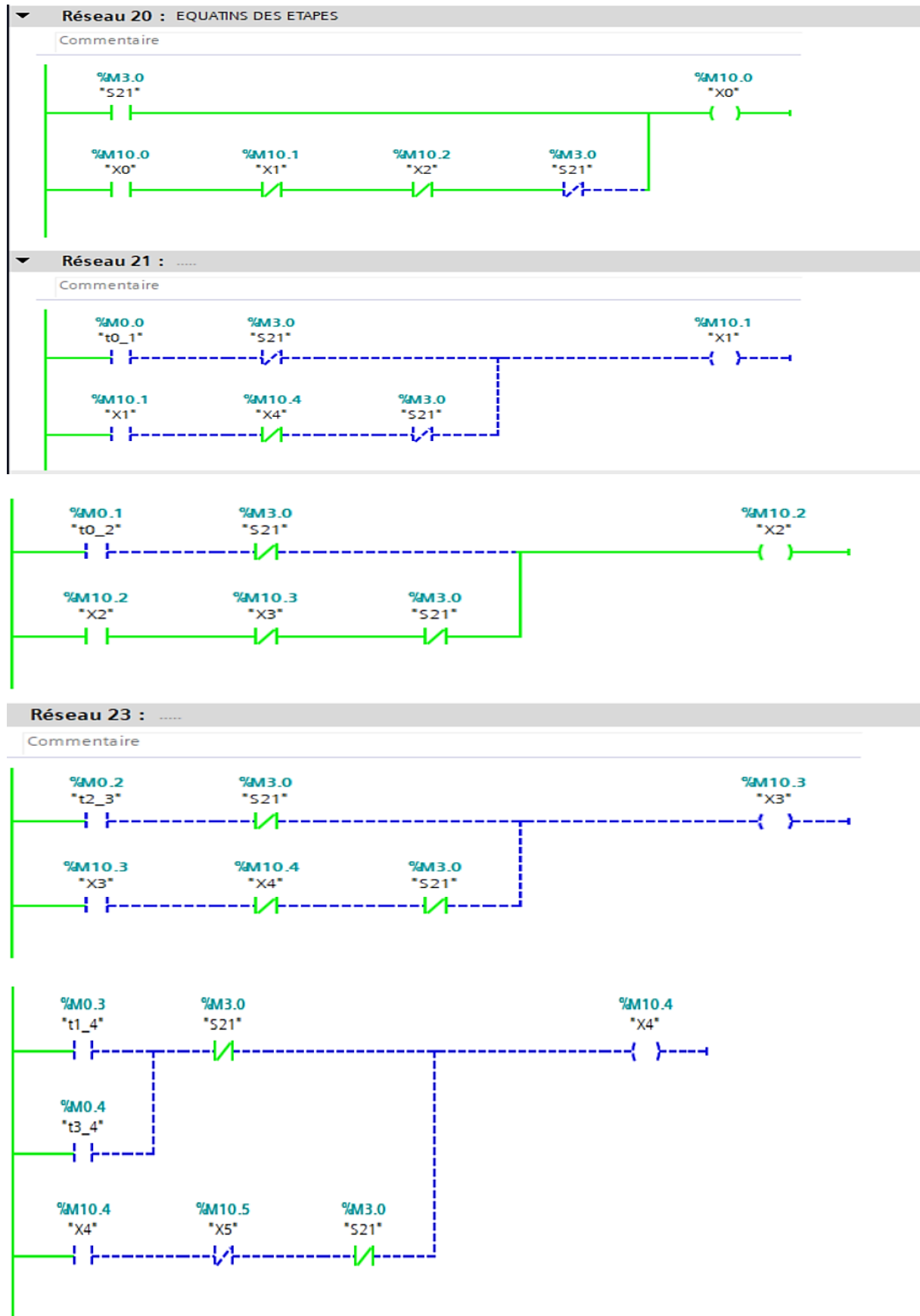
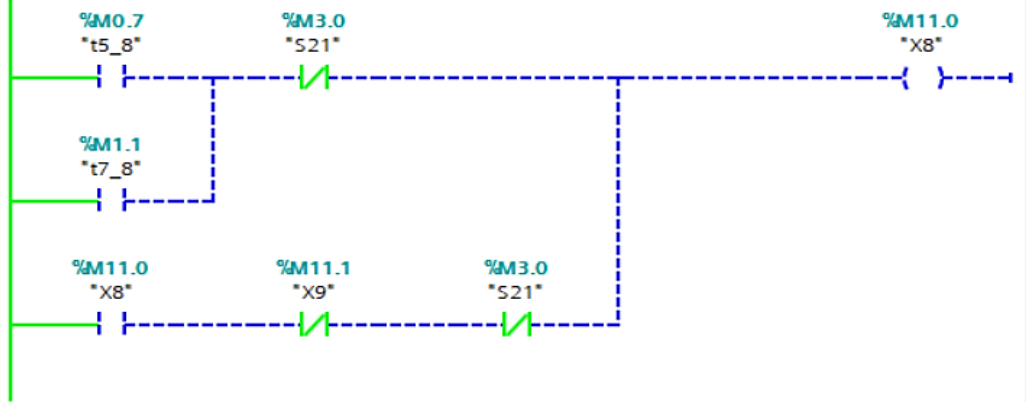
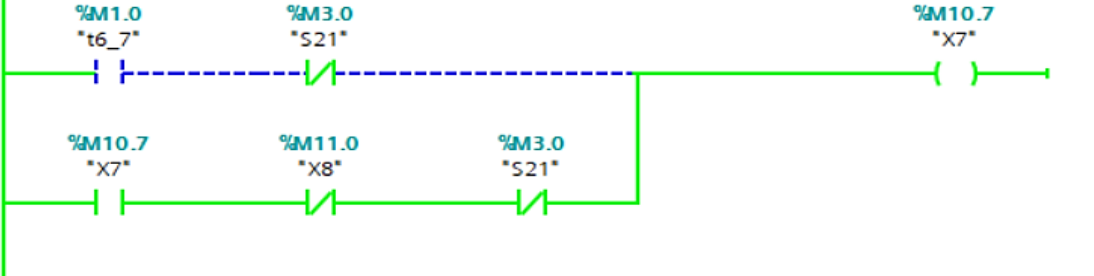
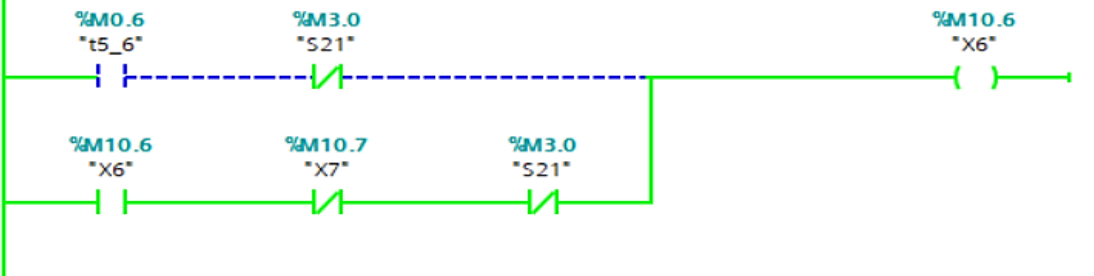
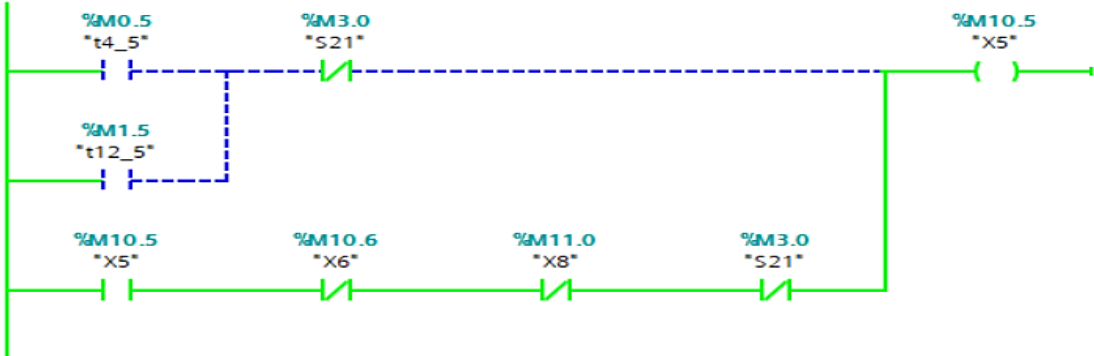


Figure B. 1 : Les transitions du GRAFCET « Dépilage_Transfert_Collage »

Les étapes :





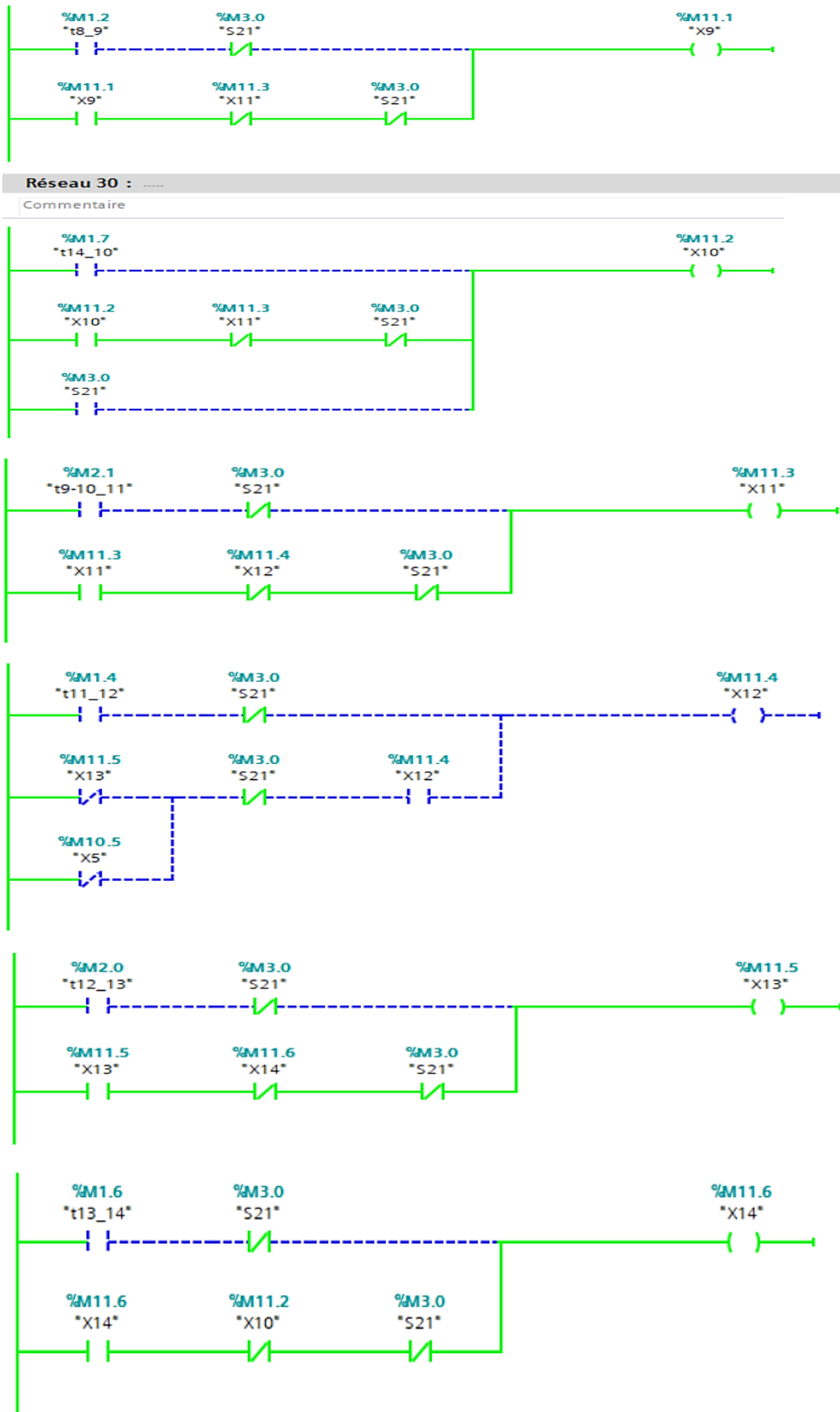
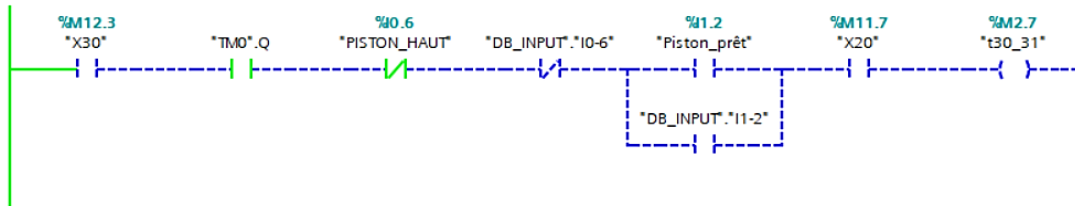


Figure B. 2 : Les étapes du GRAFCET « Dépilage_Transfert_Collage »

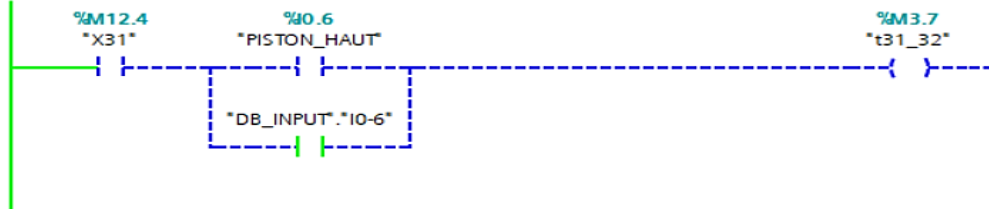
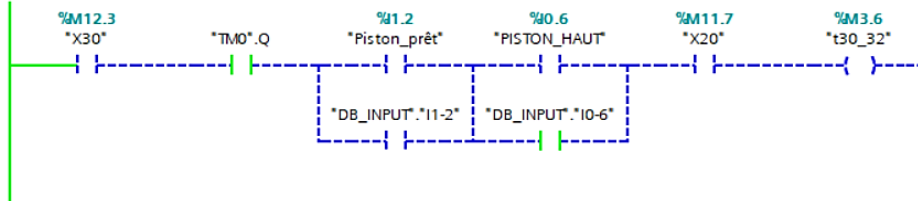
- FORMATION_CAISSE[FC3]

Les Transitions :



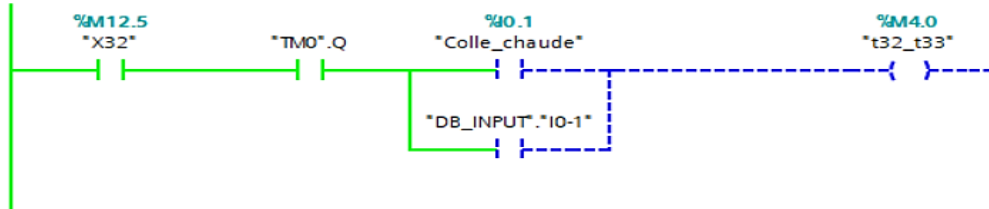
Réseau 3 :

Commentaire



Réseau 5 :

Commentaire



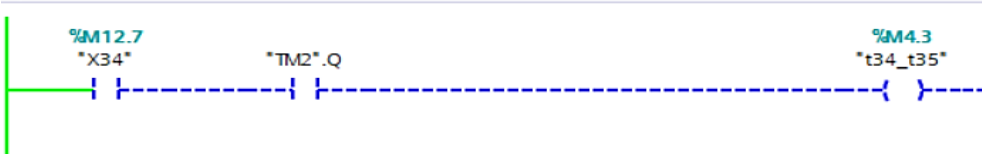
Réseau 7 :

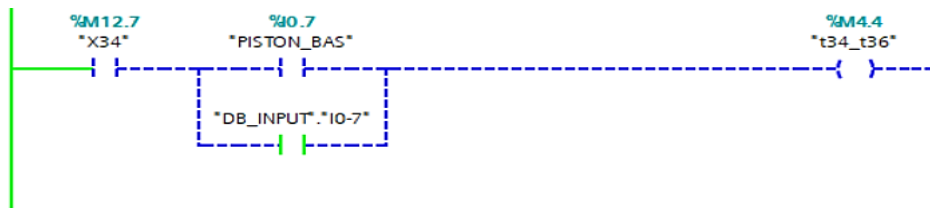
Commentaire



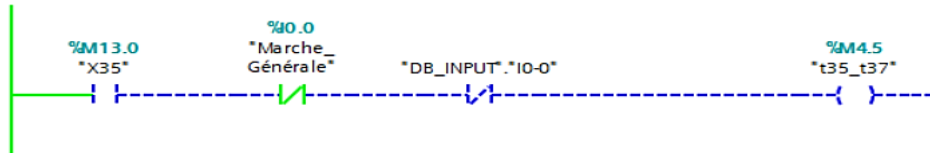
Réseau 8 :

Commentaire

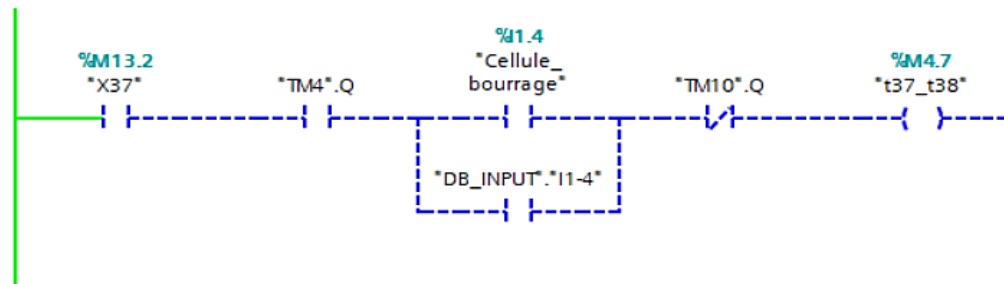


**Réseau 10 :**

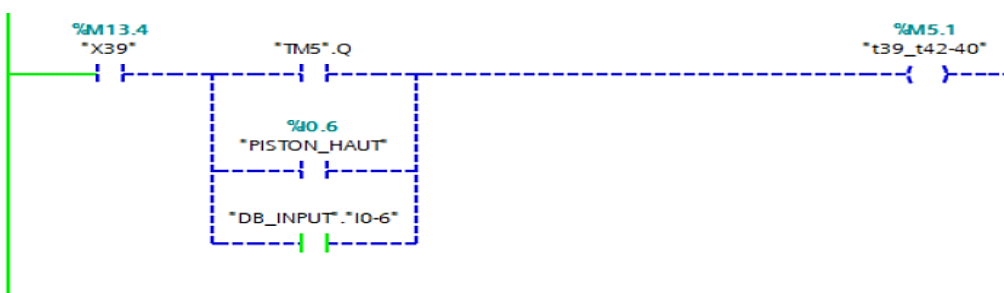
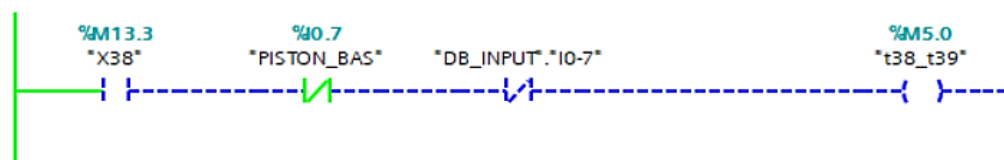
Commentaire

**Réseau 11 :**

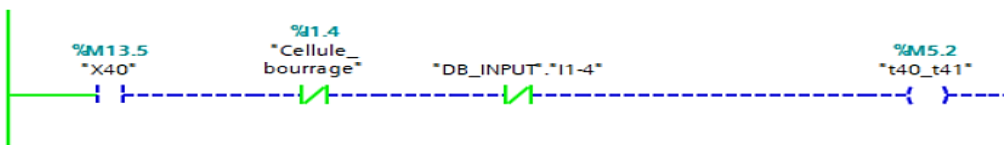
Commentaire

**Réseau 13 :**

Commentaire

**Réseau 15 :**

Commentaire



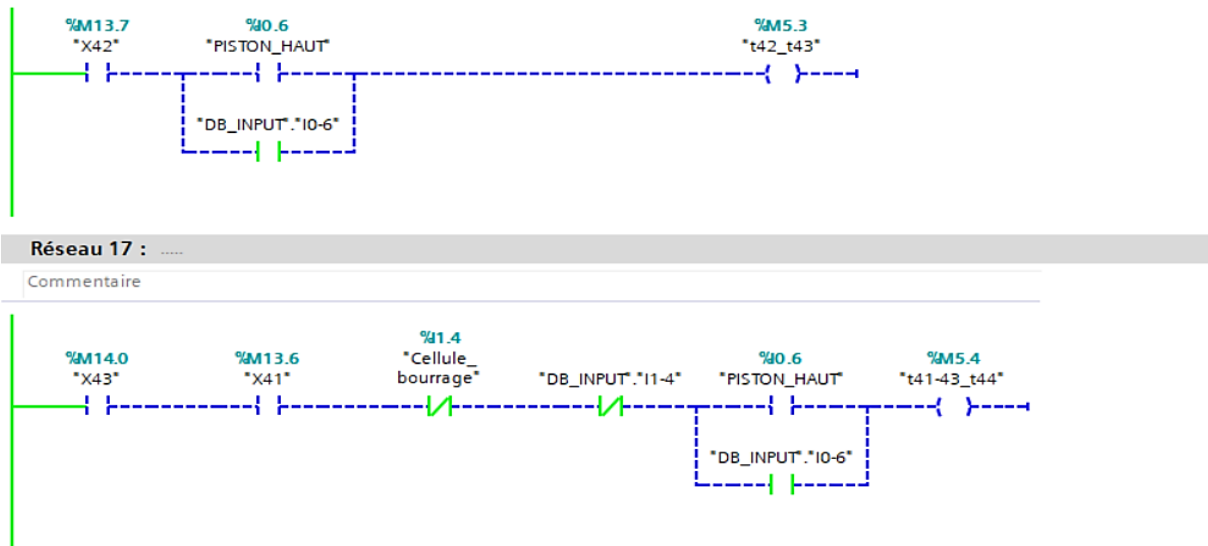
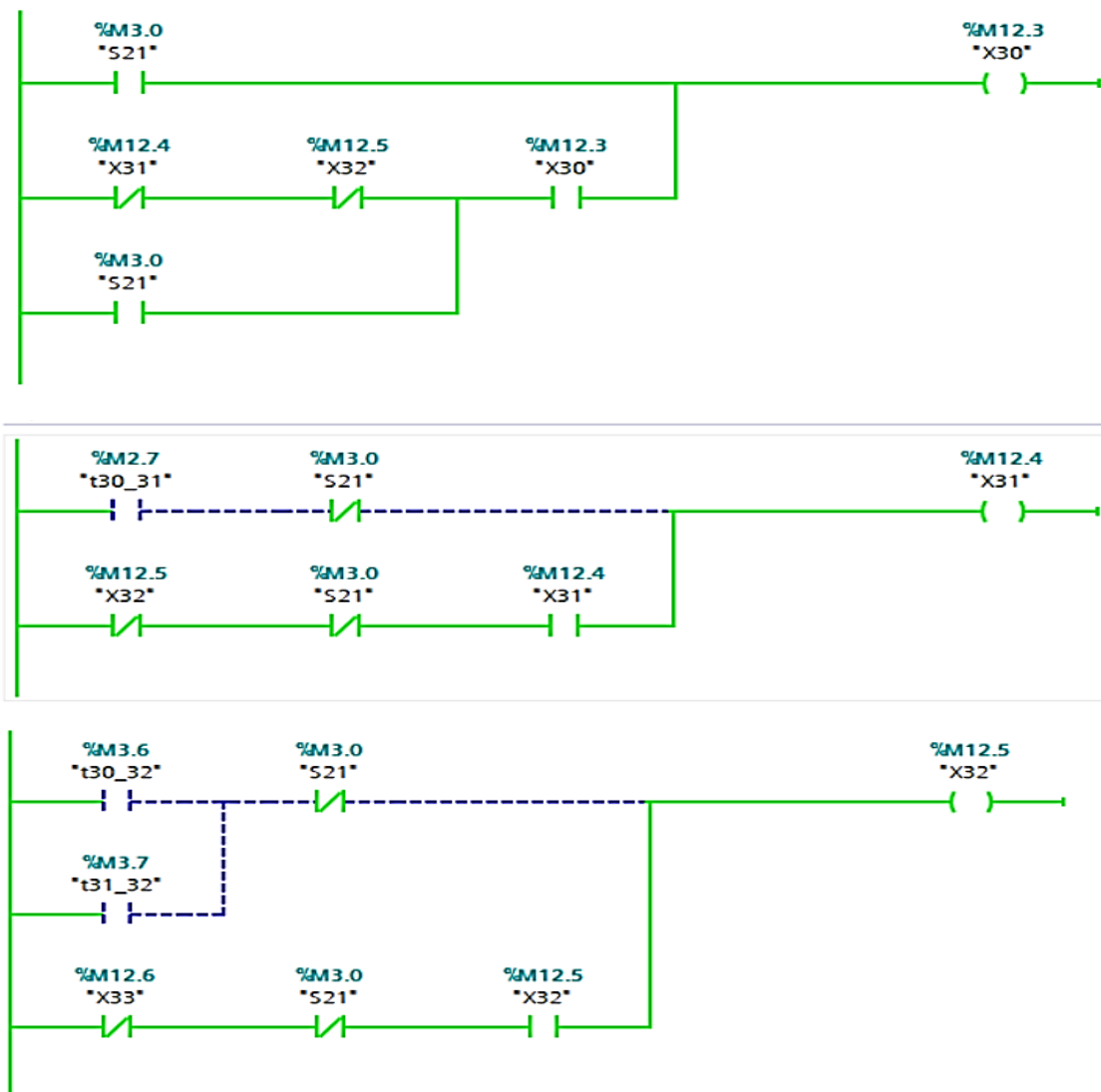
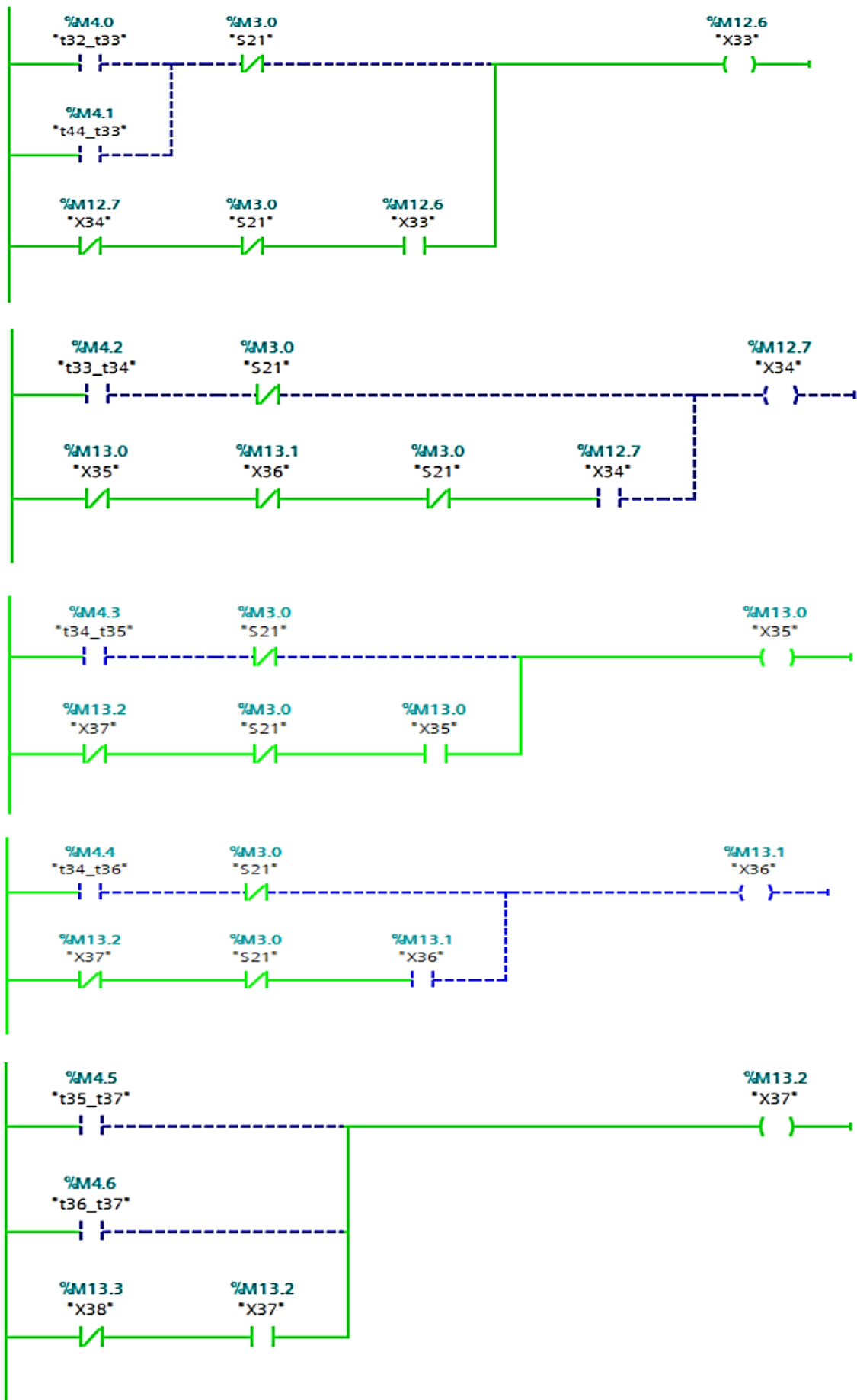
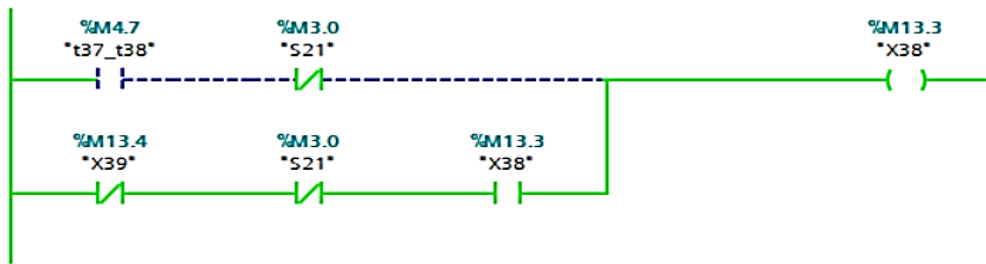


Figure B. 3 : Les transitions du GRAFCET « FORMATIO N_CAISSE »

Les étapes :

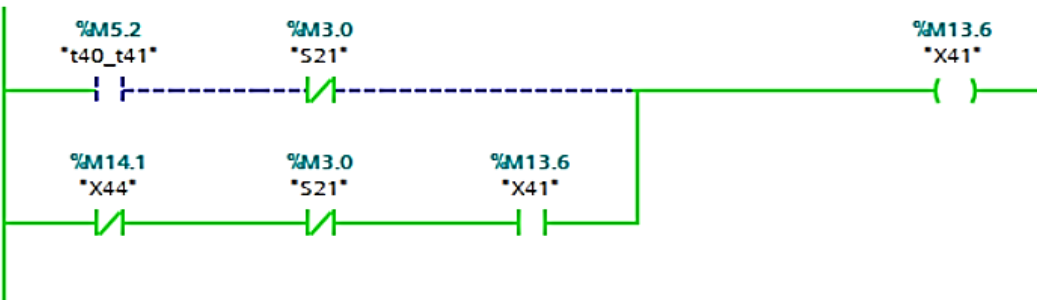
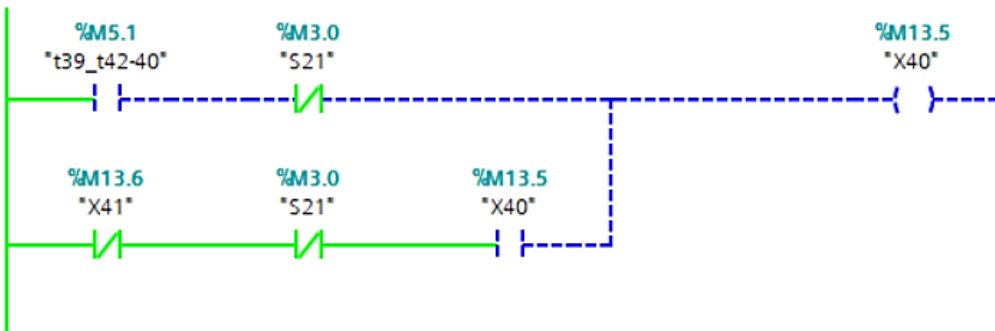
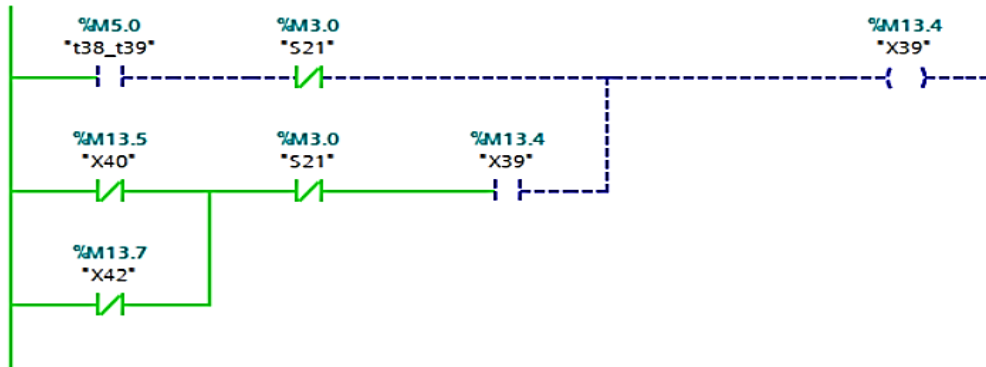






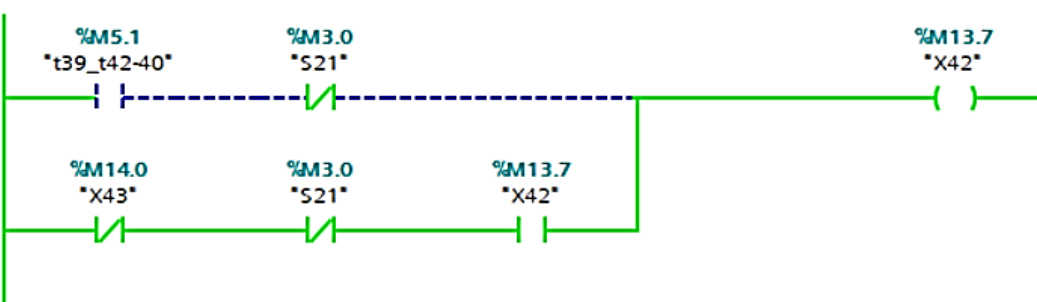
Réseau 27 :

Commentaire



Réseau 30 :

Commentaire



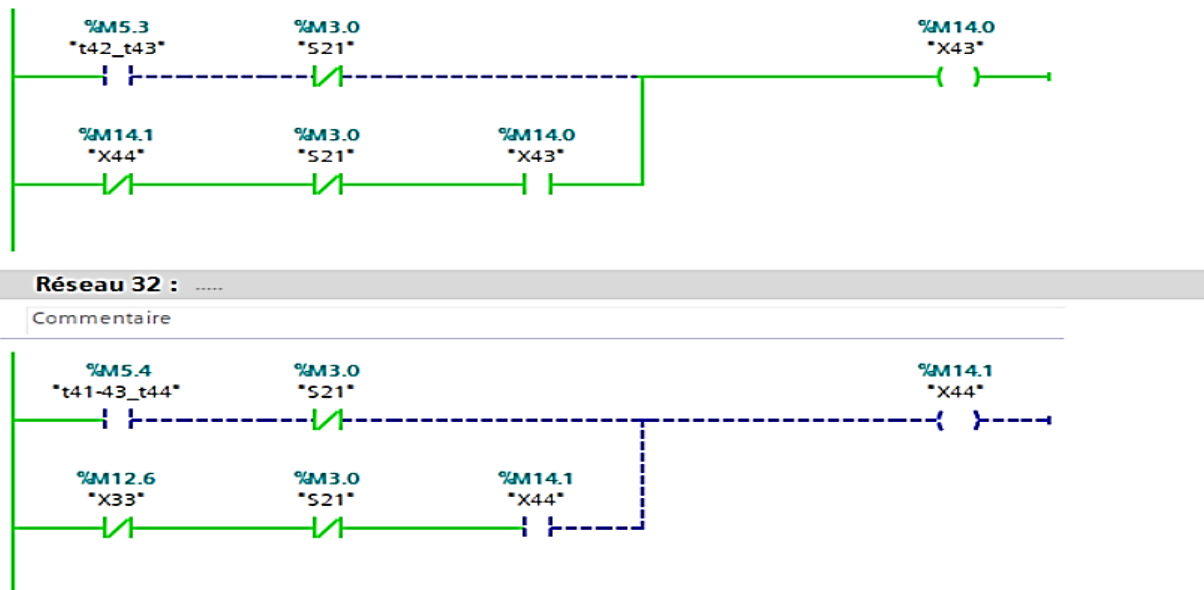
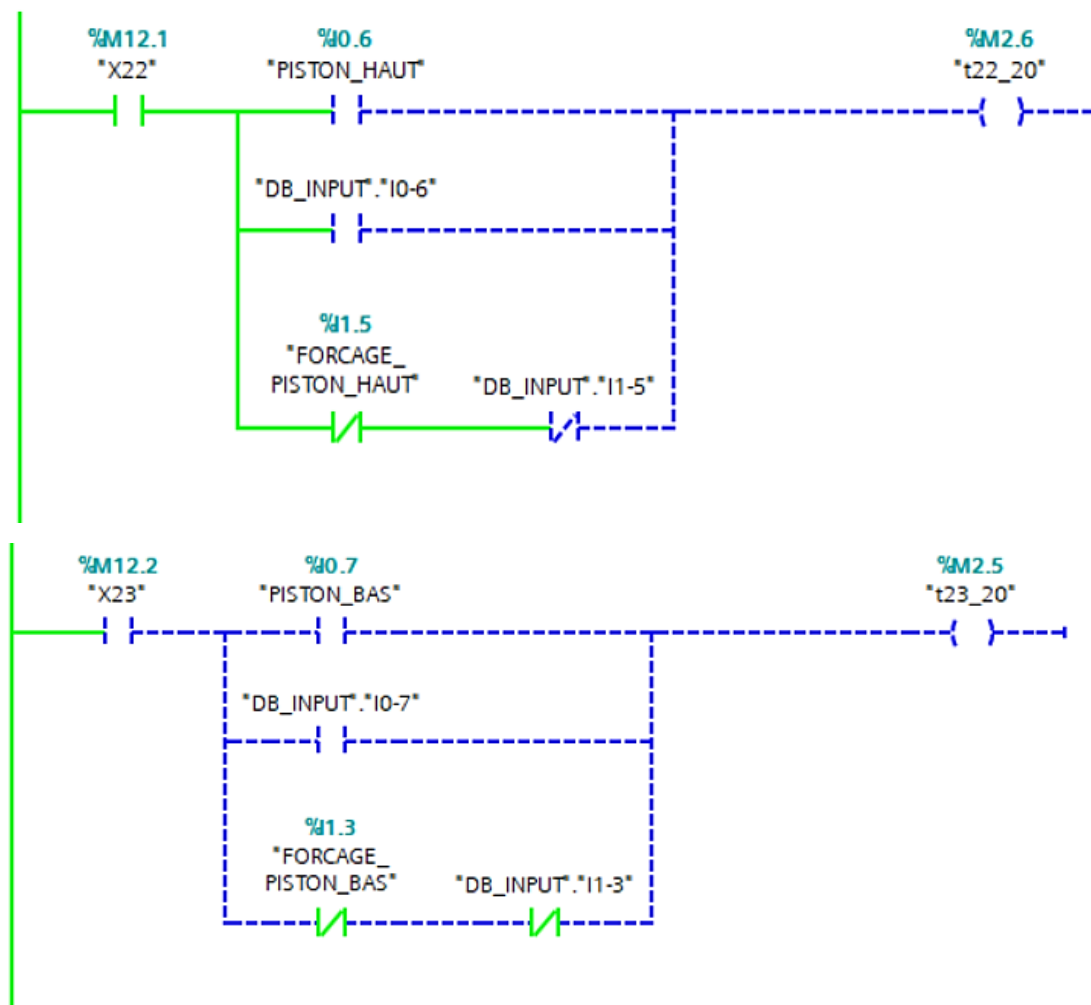


Figure B. 4 : Les étapes du GRAFCET « FORMATION_CAISSE »

- FORCAGE_PISTON[FC2]

Les transitions :



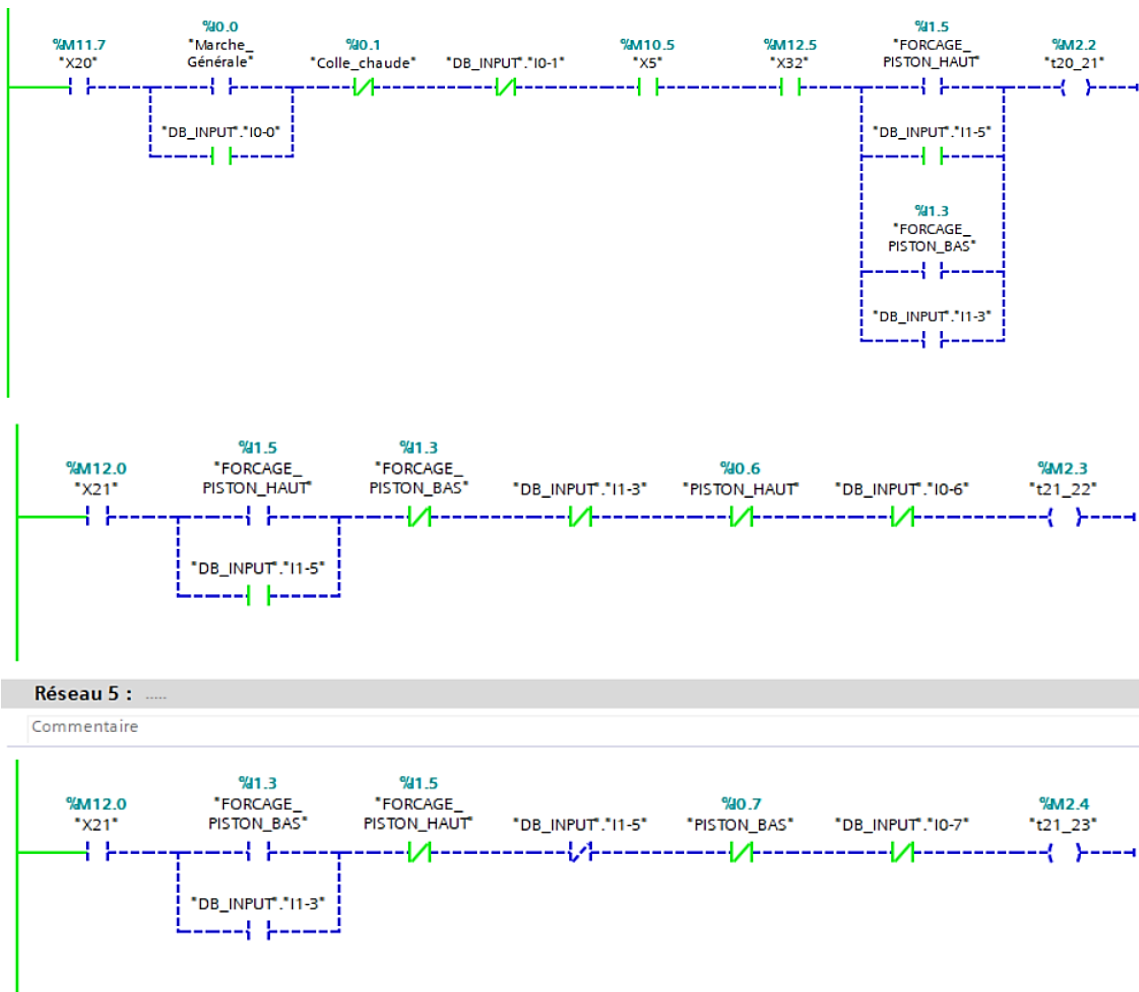
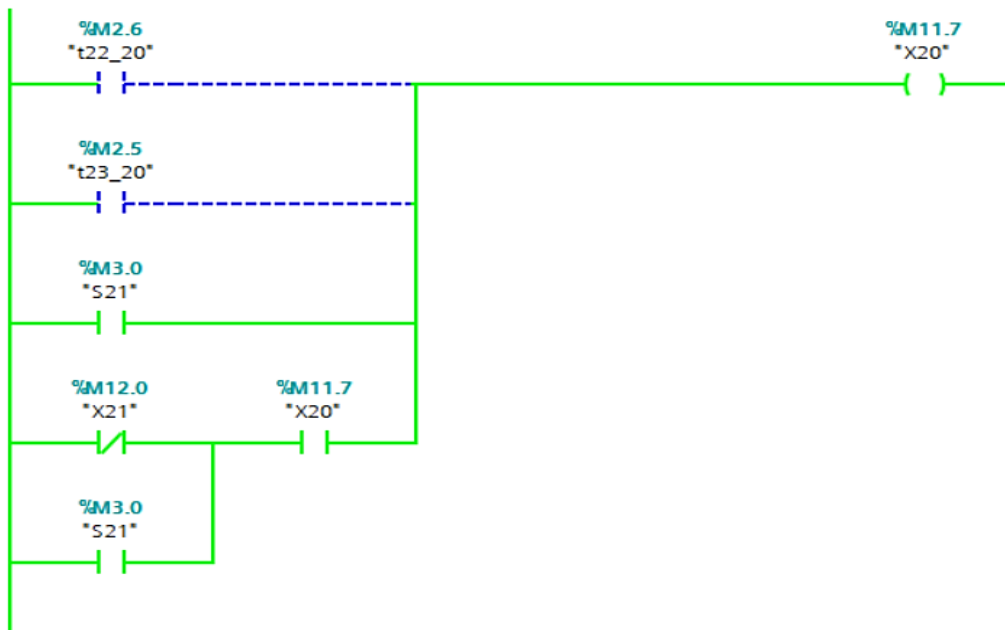


Figure B. 5 : Les transitions du GRAFCET « FORCAGE_PISTON »

Les étapes :



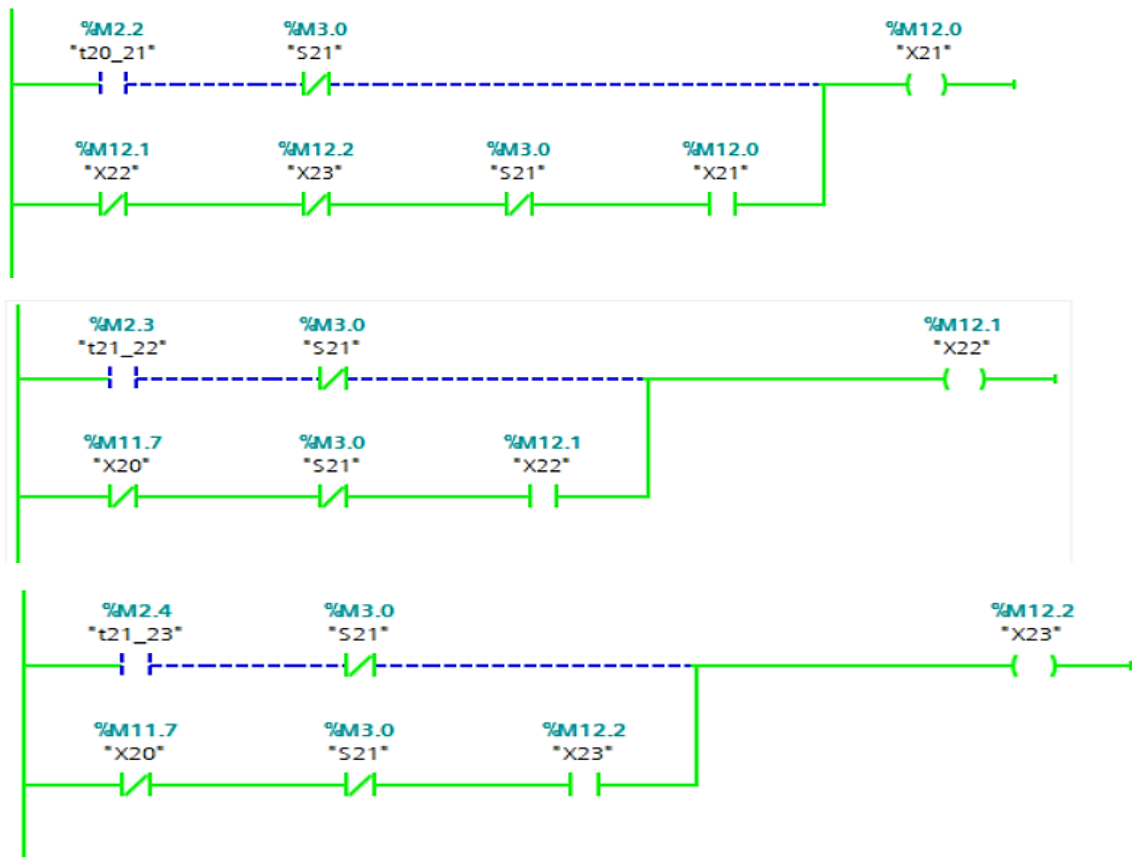
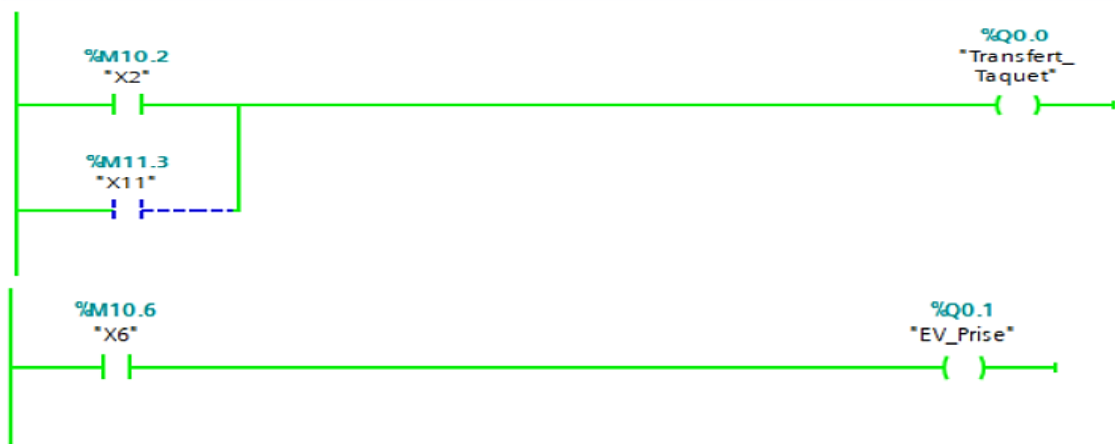


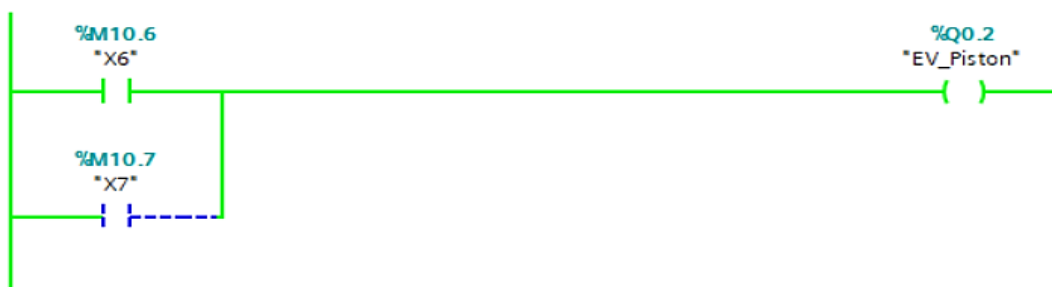
Figure B. 6 : Les étapes du GRAFCET « FORCAGE_PISTON »

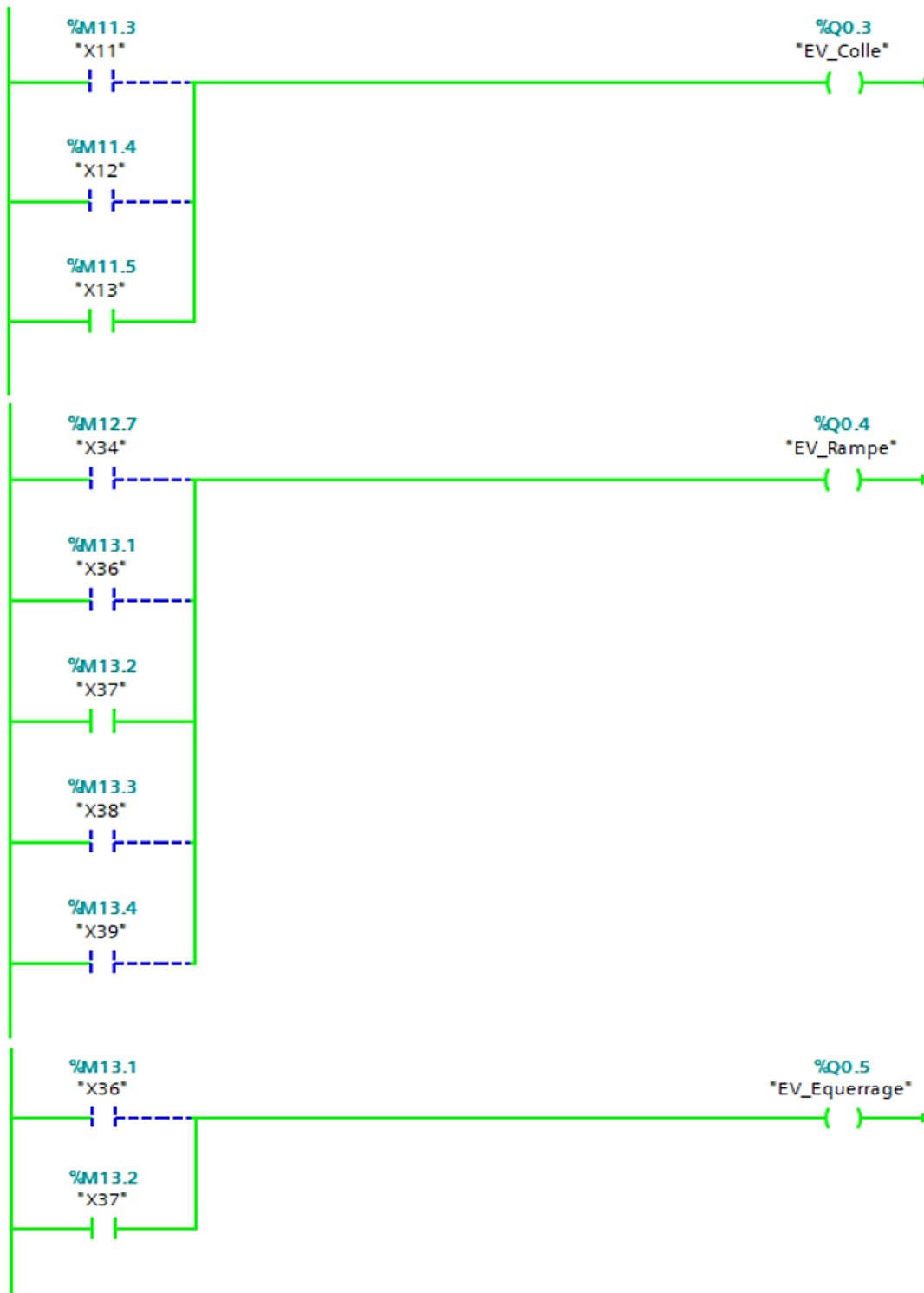
- SORTIES[FC5] (les actions des grafcet)



Réseau 3 :

Commentaire





Réseau 7 :

Commentaire



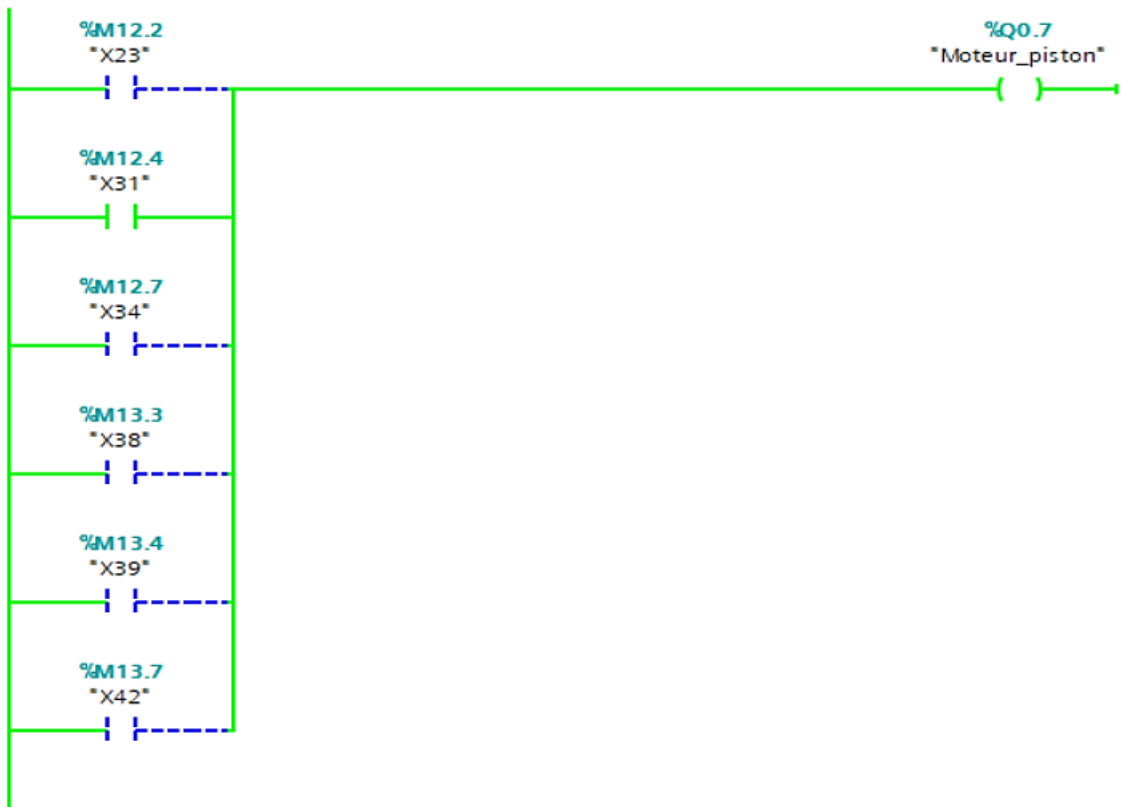
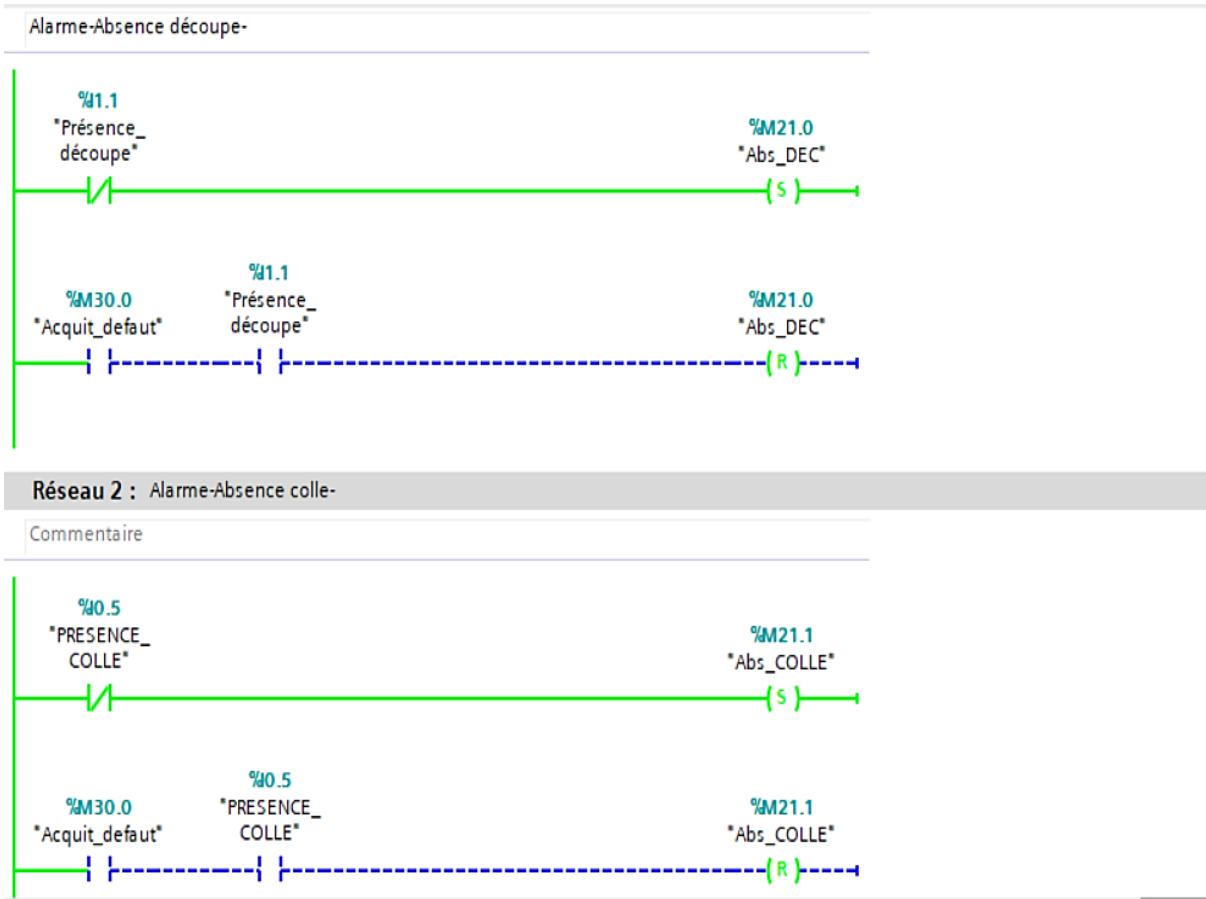


Figure B. 7 : Les sorties d'API sous TIA PORTAL

ALARMES[FC5]



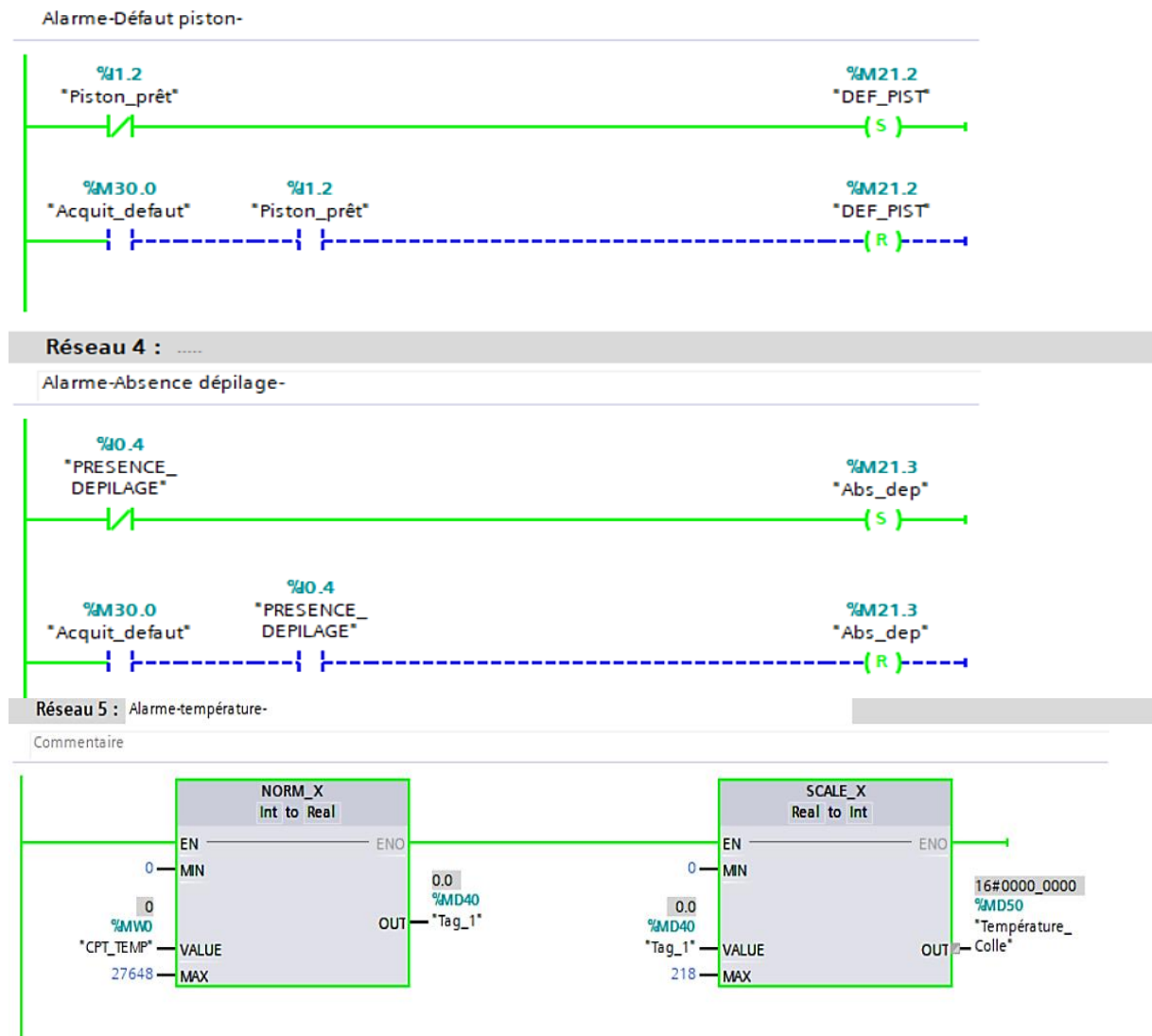
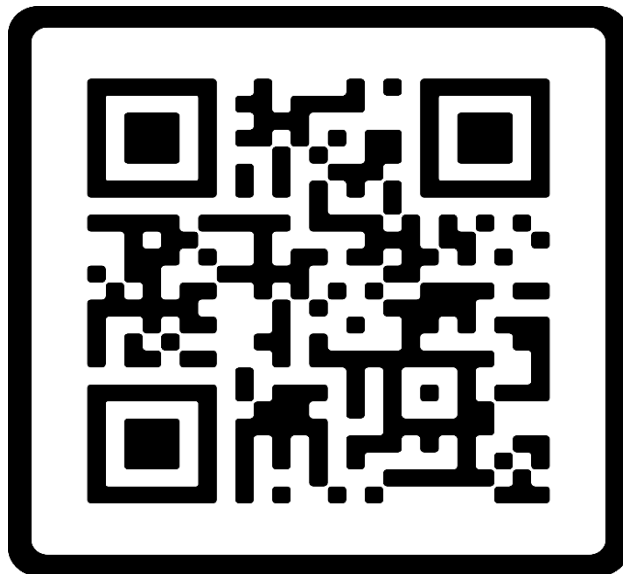


Figure B. 8 : Le bloc FC5 des alarmes





ABSTRACT

Over the past few years, automation, which has a long history, has become a major concern for many companies. This project was suggested to us by the company CEVITAL, which is focused on improving the control part of a tray filling machine and automating it through the migration from an obsolete Schneider TSX Micro programmable logic controller to an S7-1200 (CPU1214C) PLC from SIEMENS. The supervisory system was also improved using a KTP1200 touchscreen.

TIA Portal was used as a programming tool, a software that combines Step 7 and WinCC. Thanks to this software, we were able to design a programmable solution for this processing line.

Keywords : Programmable logic controller (PLC), TIAPortal, tray filling machine, Migration supervisory system, Automation.

RÉSUMÉ

Au fil des dernières années, l'automatisation, qui a déjà une longue histoire, est devenue une préoccupation majeure pour de nombreuses entreprises. Ce projet nous a été suggéré par l'entreprise CEVITAL, qui se concentre sur l'amélioration de la partie de commande d'une barquetteuse et son automatisation grâce à travers la migration d'un automate programmable obsolète Schneider TSX Micro vers un automate S7-1200(CPU1214C) de la société SIEMENS. Le système de supervision a également été amélioré en utilisant un écran tactile KTP1200.

Le TIA Portal a été employé en tant qu'outil de programmation, un logiciel qui combine le Step 7 et le WinCC. Grâce à ce logiciel, nous avons pu concevoir une solution programmable pour cette chaîne de traitement.

Mots Clés : Automate programmable industriel (API), TIAPortal, Barquetteuse, Migration, système de supervision, Automatisation.

ملخص

على مدار السنوات الأخيرة أصبحت الأتمتة التي تعود جذورها الى زمن بعيد، مصدر قلق للعديد من الشركات. تم اقتراح هذا المشروع علينا من قبل شركة سيفيتال التي تسعى جاهدة الى تحسين جزء التحكم لالة صناعة علب كرتونية، يتم ذلك من خلال الهجرة من المبرمج الصناعي الالي القديم من طراز شنايدر "TSX Micro" الى S7-1200 من شركة سيمنس، إضافة لذلك قمنا بتعزيز نظام الاشراف و برمجة شاشة رقمية تعمل باللمس من "KTP1200"

قمنا باستخدام برنامج TIA PORTAL كأداة برمجة، الذي يجمع بين برنامجي STEP7 و WinCC، بفضلها

تمكنا من تصميم حل قابل للبرمجة لسلسلة المعالج

