

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة ابي بكر بلقايد - تلمسان
Université Aboubakr Belkaïd-Tlemcen
كلية التكنولوجيا
Faculté de TECHNOLOGIE



Mémoire de fin d'études

Pour l'obtention du **diplôme de MASTER**

Filière : **Électronique**

Spécialité : Instrumentation

Réalisé par :

ROUGUEB Hichem

MKEDDER Mohammed El Amin

Intitulé du Sujet

Conception, modélisation et programmation d'un système automatisé de production

(Distribution-Usinage-manutention-livraison).

Etude pratique système automatisé MPS de FESTO du laboratoire de recherche MELT.

Soutenu le 22 Juin 2022, Devant le jury composé de :

Mr HADJ ABDELKADER Amine	-Professeur	-Univ. Tlemcen	-Président
Mr BRIXI NIGASSA Med El Amine	-MCB	-Univ. Tlemcen	-Examineur
Mr HASSAM Ahmed	-MCB	-Univ. Tlemcen	-Encadrant
Mme HERARSI Nour El Houda	-Doctorante	-Univ. Tlemcen	-Co-Encadrante

Année universitaire : 2021/2022

REMERCIEMENTS

Tout d'abord, nous remercions **Allah** le tout puissant de nous avoir donnés le courage et la patience nécessaires à mener ce travail à son terme.

Nous tenons à remercier tout particulièrement nos encadrants **Mr. HASSAM Ahmed** et **Mme. HERARSI Nour El Houda**, pour leurs compétences, les aides qu'ils nous ont apportées, leurs patiences et leurs encouragements. Leur regard critique nous a été très précieux pour structurer le travail et améliorer la qualité des différentes sections.

Nous tenons à remercier les membres de jury **Mr. Hadj Abdelkader Amine** et **Mr. Brixi Nigassa Mohammed El Amine**, l'expression de nos sincères remerciements pour l'honneur qu'ils nous font en prenant le temps de lire et d'évaluer ce travail.

Nous devons un remerciement à tous les enseignants de l'Université **Aboubakr Belkaïd-Tlemcen, faculté de technologie** pour leurs qualités scientifiques et pédagogiques.

Pour finir, nous souhaitons remercier toute personne ayant contribué de près ou de loin à la réalisation de ce travail.

DÉDICACE

“



.... Je dédie ce travail :

*À mon cher père **Mohammed**,*

*À ma chère mère **Nouara**,*

*Qui n'ont jamais cessé, de formuler des prières à mon
égard, de me soutenir et de m'épauler pour que je puisse
atteindre mes objectifs,*

*À mon cher frère **Karime** et ma chère sœur **Karima**,*

*À ma chère soeur **Farah** et son mari **Ilyes** et leur petite
princesse **Houda**,*

*Puisse **Dieu** vous donne santé, bonheur et surtout réussite.*

*À mes chères amis ,**B. Anes**, **A. Oussama**, **Z.
Oussama**,*

Pour leurs aides et supports dans les moments difficiles.

*Sans oublier mon cher binôme **Amine**,*

Pour son soutien moral, sa sympathie et sa patience.

À tous ceux qui me sont chers, à vous tous, Merci.

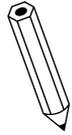
★ ★ ★

”

- *Hichem*

DÉDICACE

“



.... Je dédie ce travail :

*À Mon Cher Père **Youcef**,*

*À Ma Chère Mère **Zahia**,*

*Qui n'ont jamais cessé, de formuler des prières à mon
égard, de me soutenir et de m'épauler pour que je puisse
atteindre mes objectifs,*

*À Mon Cher Frère **Yassine MK**,*

*À Mes Chères sœurs **N MK / H MK**,*

*À Mes Chères Nièces **Salma / Rima**,*

*À Ma Chère Femme **W, BZ** et Mon Chèr Fils **Ahmed
Taha**,*

*À Mon Cher proche **AF, Miraoui**,*

*Puisse **Dieu** vous donne santé, bonheur et surtout réussite.*

*Sans oublier Mon cher binôme **Hichem**,*

Pour son soutien moral, sa sympathie et sa patience.

À tous ceux qui me sont chers, à vous tous, Merci.

★ ★ ★

”

- *Amin MK*

Table des matières

- Remerciements **I**
- Dédicace **II**
- Dédicace **III**
- Introduction générale **1**
- 1 Un aperçu sur l’automatisme 3**
 - 1.1 Introduction 4
 - 1.2 Historique de l’automatisme 4
 - 1.3 C’est quoi l’automatisation ? 4
 - 1.4 La cible à atteindre via de l’automatisme 5
 - 1.5 Système automatisé 5
 - 1.6 Domaines d’application d’un système automatisé 5
 - 1.7 Anatomie d’un système automatisée 6
 - 1.7.1 Partie Commande (P.C) 7
 - 1.7.2 Partie Opérative (P.O) 8
 - 1.7.2.1 Une vue approfondie sur la partie opérative PO 8
 - 1.7.2.1.1 Les effecteurs 8
 - 1.7.2.1.2 Les actionneurs 8
 - 1.7.2.1.3 Les pré-actionneurs 11
 - 1.7.2.1.4 Les capteurs 12
- 1.8 Les différentes technologies utilisées dans un système automatisé 17
 - 1.8.1 Logique câblée 18
 - 1.8.2 Logique programmée 18
- 1.9 Généralités sur les automates programmables industriels (API) 19
 - 1.9.1 Historique[18] 19
 - 1.9.2 Définition d’un Automate Programmable Industriel (API) 19
 - 1.9.3 Présentation des API S7-300 de Siemens 20
 - 1.9.4 Architecture matérielle d’un API 21
 - 1.9.4.1 Aspect extérieur 21
 - 1.9.4.1.1 Configuration compacte 21
 - 1.9.4.1.2 Configuration modulaire 21
 - 1.9.4.2 Structure interne 22
 - 1.9.4.2.1 Unité de traitement 23
 - 1.9.4.2.2 Module d’alimentation 23
 - 1.9.4.2.3 Modules d’entrées/sorties 24

1.9.4.2.4	Module de communication	24
1.9.4.2.5	Mémoire	28
1.9.4.2.6	Outil de programmation	28
1.9.4.2.7	Programmation d'un API	29
1.10	Conclusion	34
2	La supervision et la simulation dans l'industrie	35
2.1	Introduction	36
2.2	La supervision industrielle	36
2.3	Structure de la supervision	37
2.4	Techniques de la supervision	38
2.4.1	Acquisition de données	38
2.4.2	Surveillance	38
2.4.3	Détection	39
2.4.4	Diagnostic	39
2.4.5	Aide à la décision	40
2.4.6	La maintenance	40
2.5	Supervision dans un environnement SCADA	42
2.6	SCADA	42
2.6.1	Évolution du SCADA	42
2.6.2	Architecture du SCADA	43
2.6.3	Composants de base d'un système SCADA	44
2.7	Interface Homme-Machine (IHM)	44
2.7.1	Fonctionnalité de l'interface homme-machine	46
2.7.2	Les avantages d'une IHM	46
2.7.3	La différence entre IHM et SCADA	47
2.7.4	Conception d'IHM pour la supervision	47
2.7.5	Types d'interfaces homme-machine (IHM)	48
2.7.6	Connectivité des IHM avec les API	50
2.7.7	Applications des IHM	50
2.8	Conclusion	51
3	La programmation, simulation et l'injection au niveau des sous-stations (MPS-variant)	52
3.1	Introduction	53
3.2	Description des sous stations du Système MPS variant	53
3.2.1	Sous-station de distribution	53
3.2.1.1	Principaux composants de la sous-station de distribution	54
3.2.1.2	Cahier de charge initial de la sous-station de distribution	55
3.2.1.3	Modifications apportées sur le cahier de charge de la sous- station de distribution	55
3.2.2	Sous-station d'usinage	55
3.2.2.1	Principaux composants de la sous-station d'usinage	56
3.2.2.2	Cahier de charge initial de la sous-station d'usinage	57
3.2.2.3	Modifications apportées sur le cahier de charge de la sous- station d'usinage	57

3.2.3	Sous-station de manipulation	58
3.2.3.1	Principaux composants de la sous-station de manipulation	59
3.2.3.2	Cahier de charge initial de la sous-station de manipulation	59
3.2.3.3	Modifications apportées sur le cahier de charge de la sous-station de manipulation	59
3.2.4	Sous-station de livraison (tri)	60
3.2.4.1	Principaux composants de la sous-station de livraison (tri)	60
3.2.4.2	Cahier de charge initial de la sous-station de livraison . .	61
3.2.4.3	Modifications apportées sur le cahier de charge de la sous-station de tri	62
3.2.5	SADT des sous-stations	63
3.2.5.1	SADT de la sous-station de distribution	63
3.2.5.2	SADT de la sous-station d'usinage	64
3.2.5.3	SADT de la sous station de manipulation	65
3.2.5.4	SADT de la sous station de livraison	66
3.2.6	Pupitre de commande	67
3.3	Aperçu de Step 7	67
3.3.1	Démarrage de Step 7	67
3.3.2	Création d'un projet	68
3.3.3	Configuration matérielle	69
3.3.4	Communication via câble Ethernet	71
3.3.5	Configuration du réseau	71
3.3.6	Modélisation de notre système avec GRAFCET	73
3.3.7	Appel du FB1 Depuis le bloc OB1	74
3.4	Modélisation du MPS Variant	74
3.4.1	Sous-station de distribution	74
3.4.1.1	Déclaration des variables de la sous-station de distribution	74
3.4.1.2	Création du Grafcet de la sous-station de distribution . .	74
3.4.2	Sous-station d'usinage	75
3.4.2.1	Déclaration des variables de la sous-station d'usinage . . .	75
3.4.2.2	Création du Grafcet de la sous-station d'usinage	76
3.4.3	Sous-station de manipulation	76
3.4.3.1	Déclaration des variables de la sous-station de manipulation	76
3.4.3.2	Création du Grafcet de la sous-station de manipulation .	77
3.4.4	Sous-station de livraison (tri)	78
3.4.4.1	Déclaration des variables de la sous-station de livraison .	78
3.4.4.2	Création du Grafcet de la sous-station de livraison	78
3.5	Simulation de notre MPS variant avec S7-PLCSIM	80
3.5.1	Présentation du S7-PLCSIM	80
3.5.2	Simulation de la sous-station de distribution	81
3.5.3	Simulation de la sous-station d'usinage	83
3.5.4	Simulation de la sous-station de manipulation	84
3.5.5	Simulation de la sous-station de livraison (tri)	85
3.6	Test des sous-stations réelles via l'outil L'EASYPORT USB	87
3.6.1	EASYPORT USB	87
3.6.2	EzOPC	88

3.6.3	Branchement de la Sous-Station réelle avec EASYPORT USB	88
3.7	Injection des programmes dans l'API	93
3.8	Résultats obtenus après l'injection de chaque sous-station par son programme	109
3.8.1	Sous-station de distribution	110
3.8.2	Sous-station d'usinage	111
3.8.3	Sous-station de manipulation	111
3.8.4	Sous-station de livraison (tri)	112
3.9	Conclusion	113
4	La conception, simulation et supervision de notre MPS variant	114
4.1	Introduction	115
4.2	WinCC flexible	115
4.2.1	WinCC flexible Runtime	116
4.2.2	Démarrage de WinCC flexible	116
4.2.3	Création d'un projet	116
4.2.4	Sélection du pupitre	117
4.2.5	Intégration du projet avec le SETP7	118
4.2.6	Communication SIMATIC et IHM	120
4.2.7	Création des vues	123
4.2.7.1	Utilisation des objets dans la zone de travail	124
4.3	Création, conception et simulation de nos IHMs	125
4.3.1	IHM de la sous-station de distribution	125
4.3.2	IHM de la sous-station d'usinage	126
4.3.3	IHM de la sous-station de manipulation	127
4.3.4	IHM de la sous-station de livraison (tri)	128
4.4	Simulation des IHMs sous S7-PLCSIM et WINCC FLEXIBLE RUNTIME	128
4.4.1	Sous-station de distribution	129
4.4.2	Sous-station d'usinage	131
4.4.3	Sous-station de manipulation	134
4.4.4	Sous-station de livraison (tri)	136
4.5	Conclusion	139
	Conclusion générale	140
	Bibliographie	144
	Webographie	147

Table des figures

1.1	Chaîne de production en série automatisée destinée à souder la carrosserie des véhicules[6].	6
1.2	Les systèmes automatisés dans l'organisation de la circulation[6].	6
1.3	Planification de fonctionnement d'un système automatisé[2].	7
1.4	Schéma de fonctionnement d'un effecteur[7].	8
1.5	Schéma de fonctionnement d'un actionneur[7].	9
1.6	Les différents actionneurs électriques[11][12][13].	10
1.7	Les différents actionneurs pneumatiques[12][14].	10
1.8	Exemple d'un Vérin hydraulique[12].	11
1.9	Schéma de fonctionnement d'un pré-actionneur[7].	11
1.10	Les différents pré-actionneurs[7].	12
1.11	Principe de fonctionnement d'un capteur[16].	12
1.12	Les types de l'information des capteurs[15].	12
1.13	Divers types de capteurs inductifs[16]	13
1.14	Capteur de proximité capacitif[16].	14
1.15	Principe de fonctionnement d'un capteur photoélectrique à barrage[19]. . .	14
1.16	Principe de fonctionnement d'un capteur photoélectrique réfléchissant[19].	15
1.17	Principe de fonctionnement d'un capteur photoélectrique rétro-réfléchissant[19].	15
1.18	Capteur de pression[20].	16
1.19	Exemple d'un capteur à ampoule REED[21].	16
1.20	Capteur de position mécanique (fin de course)[22].	17
1.21	Principe de fonctionnement d'un thermocouple[15].	17
1.22	Différence entre armoire à relais et armoire à API[24]	18
1.23	Automates programmables industriels de type "Siemens"[27].	20
1.24	Automates programmables industriels de type "Allen-Bradley"[24].	20
1.25	Automate programmable compact S7-1200[29].	21
1.26	Automate modulaire s7-300[30].	22
1.27	Structure interne d'un API[30].	22
1.28	Module CPU Siemens (CPU 314)[31].	23
1.29	Module d'alimentation PS 307-5A[32].	23
1.30	Exemple de réseaux Modbus[33].	25
1.31	La structure du système AS-i[34].	25
1.32	Les principaux composants dans PROFIBUS DP[36].	26
1.33	Les principaux composants dans PROFINET[39].	27
1.34	Configuration du système avec PN/CAN LINK[40].	27
1.35	(a) Représentation d'un circuit électrique. (b) Représentation en Ladder[42].	30
1.36	Cycle d'exécution dans un diagramme Ladder[42].	30
1.37	FBD ou logigramme[43].	31

1.38	Les différentes composantes d'un Grafset[44].	33
2.1	Place de la supervision dans un système automatisé[45].	36
2.2	Poste de supervision[49].	37
2.3	Structure globale d'un système de supervision[50].	38
2.4	Différents types de surveillance[52].	39
2.5	Les différentes formes de maintenance[54].	40
2.6	L'architecture SCADA[58].	43
2.7	Composants du système SCADA[59].	44
2.8	Les organes de visualisation avant l'apparition de l'IHM[61].	45
2.9	Exemple d'une IHM[62].	45
2.10	Fonctionnalité de l'interface homme-machine[63].	46
2.11	Structure simplifiée d'un système interactif[58].	47
2.12	Visualisation avec IHM de type SIMATIC[66].	49
2.13	Surveillance à distance à l'aide d'un appareil mobile[62].	49
2.14	Connectivité d'IHM avec l'API[60].	50
2.15	Exemple d'un écran tactile pour contrôler le processus d'embouteillage[68].	51
3.1	Sous-station de distribution du système FESTO MPS.	53
3.2	Composants principaux de la sous-station de distribution[69]. - (a) Module de transfert. - (b) Magasin à empilage.	54
3.3	Sous-station d'usinage du système FESTO MPS.	56
3.4	Principaux composants de la sous-station d'usinage[69]. - (a) Module table à indexation rotative. - (b) Module de contrôle. - (c) Module de perçage.	57
3.5	Sous-station de manipulation du système FESTO MPS.	58
3.6	Module de manipulateur "PickAlfa"[69].	59
3.7	Sous-station de livraison (tri) du système FESTO MPS.	60
3.8	Principaux composants de la sous-station de tri[69]. - (a) Module de glissière. - (b) Module de bande de tri.	61
3.9	Fonction globale A-0 de la sous-station de distribution.	63
3.10	Fonction globale A0 de la sous-station de distribution.	63
3.11	Fonction globale A-0 de la sous-station d'usinage.	64
3.12	Fonction globale A0 de la sous-station d'usinage.	64
3.13	Fonction globale A3 de la sous-station d'usinage.	65
3.14	Fonction globale A-0 de la sous-station de manipulation.	65
3.15	Fonction globale A0 de la sous-station de manipulation.	65
3.16	Fonction globale A-0 de la sous-station de livraison.	66
3.17	Fonction globale A0 de la sous-station de livraison.	66
3.18	Module du pupitre de commande.	67
3.19	L'icône du logiciel SIMATIC Manager.	67
3.20	Fenêtre d'assistant pour créer rapidement un projet Step 7.	68
3.21	Création d'un nouveau projet.	68
3.22	Fenêtre d'insertion d'une station.	69
3.23	La station SIMATIC 300.	69
3.24	Partie matérielle.	69
3.25	Configuration matérielle (HW Config).	70
3.26	Insertion du Rack-300.	70

3.27	Insertion d'une alimentation PS 307 5A.	70
3.28	Insertion CPU 314C-2 PN/DP	71
3.29	Choix de la communication via câble Ethernet.	71
3.30	Fenêtre de configuration réseau.	72
3.31	Configuration des adresses d'entrées/sorties.	72
3.32	Hierarchie du programme STEP 7.	73
3.33	Création d'un GRAFCET.	73
3.34	Table des variables la sous-station de distribution.	74
3.35	Une partie GS de la sous-station de distribution.	75
3.36	Une partie GPN de la sous-station distribution.	75
3.37	Table des variables la sous-station d'usinage	75
3.38	Une partie GS de la sous-station d'usinage.	76
3.39	Une partie GPN de la sous-station d'usinage.	76
3.40	Table des variables la sous-station de manipulation	77
3.41	Une partie GS de la sous-station de manipulation.	77
3.42	Une partie GPN de la sous-station Manipulation.	77
3.43	Table des variables la sous-station de livraison.	78
3.44	Une partie GS de la sous-station de tri.	79
3.45	Une partie GPN de la sous-station de tri des pièces (a).	79
3.46	Une partie GPN de la sous-station tri des pièces (b).	79
3.47	Une partie GPN de la sous-station tri des pièces (c).	80
3.48	L'interface du simulateur S7-PLCSIM.	81
3.49	Chargement du programme et lancement de la visualisation.	81
3.50	Simulation de la sous-station de distribution partie GS (a).	82
3.51	Simulation de la sous-station de distribution partie GPN.	82
3.52	Simulation de la sous-station de distribution partie GS (b).	83
3.53	Simulation de la sous-station d'usinage partie GS.	83
3.54	Simulation de la sous-station d'usinage partie GPN (a).	84
3.55	Simulation de la sous-station d'usinage partie GPN (b).	84
3.56	Simulation de la sous-station de manipulation partie GPN (a).	85
3.57	Simulation de la sous-station de manipulation partie GPN (b).	85
3.58	Simulation de la sous-station de tri partie GPN (a).	86
3.59	Simulation de la sous-station de tri partie GPN (b).	86
3.60	Simulation de la sous-station de tri partie GPN (c).	86
3.61	Câblage d'un processus réel et un PC à l'aide d'un EasyPort USB[71].	87
3.62	Identification d'EasyPort USB[71].	87
3.63	Mode de connexion de l'interface EzOPC.	88
3.64	Branchement de la sous-station de distribution via EasyPort USB.	89
3.65	Choix des partenaires de communication dans l'interface EzOPC.	89
3.66	La synchronisation entre le Grafcet partie GS, l'interface EzOPC, PLCSIM et l'EasyPort.	90
3.67	La synchronisation entre l'EasyPort et le pupitre de commande (a).	91
3.68	La synchronisation entre l'EasyPort et le pupitre de commande (b).	91
3.69	La synchronisation entre le Grafcet partie GS, le simulateur PLCSIM et l'interface EzOPC.	92
3.70	La synchronisation entre l'EasyPort et le pupitre de commande (c).	92

3.71	La synchronisation entre le Grafcet partie GPN, le simulateur PLCSIM et l'interface EzOPC.	93
3.72	Carte mémoire MMC du type SIEMENS.	93
3.73	L'adaptateur USB MPI[72].	94
3.74	Branchement de l'automate avec le PC via l'adaptateur USB-MPI.	95
3.75	Changement du type de communication à l'interface MPI.	95
3.76	Menu "Paramètres de l'interface PG/PC".	96
3.77	Le choix de la communication MPI dans la fenêtre "PG/PC interface settings".	96
3.78	La communication entre notre projet et l'adaptateur PC MPI est établie.	97
3.79	Vérification de la configuration du réseau de notre projet.	97
3.80	La fenêtre d'état d'Ethernet.	98
3.81	Recherche de la fenêtre propriétés de protocole Internet version 4 (TCP/IPv4).	98
3.82	Fenêtre de propriétés de protocole TCP/IPv4	99
3.83	Changement de type de communication à l'interface Ethernet.	99
3.84	"Paramètres de l'interface PG/PC".	100
3.85	Sélection de la carte réseau du PC.	100
3.86	Communication entre PC et l'automate via Ethernet.	101
3.87	Menu pour ouvrir l'Éditer Station Ethernet.	101
3.88	Fenêtre "Édit Station Ethernet ".	102
3.89	Recherche automatique de l'adresse IP.	102
3.90	Fenêtre "Édit Ethernet Station" après la recherche de l'adresse IP.	103
3.91	Changement de la communication MPI vers Profibus.	104
3.92	Le choix de communication Profibus (a).	104
3.93	Le choix de la communication Profibus (b).	104
3.94	Communication Profibus établie.	105
3.95	La fenêtre du SIMATIC Manager.	105
3.96	La fenêtre de configuration matérielle (HW Config).	105
3.97	Le choix du module cible.	106
3.98	Chargement avec la communication MPI.	106
3.99	Chargement avec la communication Ethernet.	107
3.100	Chargement de la configuration matérielle.	107
3.101	Sélection de tous les blocs.	108
3.102	Chargement des blocs.	108
3.103	Chargement du bloc FB1 (partie fonctionnelle).	108
3.104	Le chargement du programme via la communication MPI.	109
3.105	Notre nouveau système de production automatisé MPS variant	110
3.106	Résultats obtenus pour la sous-station du distribution.	110
3.107	Le voyant "magasin vide" de la sous-station de distribution.	111
3.108	Les résultats obtenus pour la sous-station d'usinage.	111
3.109	Les résultats obtenus pour la sous-station de manipulation.	112
3.110	Les résultats obtenus pour la sous-station de livraison (tri).	112
3.111	Le voyant "glissières pleines" de la sous-station de livraison (tri).	113
4.1	L'icône WinCC flexible sur le bureau de Windows.	115
4.2	L'interface principale du logiciel WinCC flexible.	116

4.3	Création d'un nouveau projet.	117
4.4	Sélection du pupitre de commande.	117
4.5	Chargement d'un nouveau projet.	118
4.6	L'interface de travail au niveau du WinCC flexible.	118
4.7	L'intégration d'un projet WinCC Flexible à SIMATIC Manager.	119
4.8	Le choix de projet Step 7 qui nous devons lier avec WinCC flexible	120
4.9	L'évidence d'un pupitre opérateur dans notre projet Step 7.	120
4.10	Cliquons sur "Configuration".	121
4.11	Configuration du pupitre de commande.	121
4.12	Fenêtre d'interface Ethernet	122
4.13	Établissement de la connexion Ethernet au niveau du Step 7.	122
4.14	Activation de la liaison au niveau de WinCC flexible.	123
4.15	Fenêtre d'une vue dans le WinCC flexible avec ses divers éléments.	123
4.16	Configuration des objets (a)	125
4.17	Configuration des objets (b)	126
4.18	IHM de la sous station de distribution avant simulation sous Runtime. . .	126
4.19	IHM de la sous-station d'usinage avant simulation sous Runtime.	127
4.20	IHM de la sous-station de manipulation avant simulation sous Runtime. . .	127
4.21	IHM de la sous-station de livraison avant simulation sous Runtime.	128
4.22	L'icône pour lancer le simulateur Runtime	128
4.23	Simulation de la sous-station de distribution (PLCSIM et Runtime) (a). . .	129
4.24	Simulation de la sous-station de distribution (PLCSIM et Runtime) (b). . .	129
4.25	Simulation de la sous-station de distribution (PLCSIM et Runtime) (c). . .	130
4.26	Simulation de la sous-station de distribution (PLCSIM et Runtime) (d). . .	130
4.27	Simulation de la sous-station d'usinage (PLCSIM et Runtime) (a).	131
4.28	Simulation de la sous-station d'usinage (PLCSIM et Runtime) (b).	131
4.29	Simulation de la sous-station d'usinage (PLCSIM et Runtime) (c).	132
4.30	Simulation de la sous-station d'usinage (PLCSIM et Runtime) (d).	132
4.31	Simulation de la sous-station d'usinage (PLCSIM et Runtime) (e).	133
4.32	Simulation de la sous-station d'usinage (PLCSIM et Runtime) (f).	133
4.33	Simulation de la sous-station de manipulation (PLCSIM et Runtime) (a). .	134
4.34	Simulation de la sous-station de manipulation (PLCSIM et Runtime) (b). .	134
4.35	Simulation de la sous-station de manipulation (PLCSIM et Runtime) (c). .	135
4.36	Simulation de la sous-station de manipulation (PLCSIM et Runtime) (d). .	135
4.37	Simulation de la sous-station de manipulation (PLCSIM et Runtime) (e). .	136
4.38	Simulation de la sous-station de manipulation (PLCSIM et Runtime) (f). .	136
4.39	Simulation de la sous-station de livraison (PLCSIM et Runtime) (a). . . .	137
4.40	Simulation de la sous-station de livraison (PLCSIM et Runtime) (b). . . .	137
4.41	Simulation de la sous-station de livraison (PLCSIM et Runtime) (c). . . .	138
4.42	Simulation de la sous-station de livraison (PLCSIM et Runtime) (d). . . .	138
4.43	Simulation de la sous-station de livraison (PLCSIM et Runtime) (e). . . .	139

LISTE DES ACRONYMES

MELT	<i>Manufacturing Engineering Laboratory of Tlemcen</i>
API	<i>Automates programmables industriels</i>
PLC	<i>Programmable logic controllers</i>
SPA	<i>Systèmes de Production Automatisés</i>
P.O	<i>Partie Opérative</i>
P.C	<i>Partie Commande</i>
CPU	<i>Central Processing Unit</i>
TOR	<i>Tout Ou Rien</i>
E/S	<i>Entrées/Sorties</i>
MPI	<i>Multi Point Interface</i>
S7	<i>Step 7</i>
SIMATIC	<i>Siemens Automatique</i>
SADT	<i>Structured Analysis and Design Technique</i>
CONT	<i>Schéma à contacts</i>

Table des figures

LOG	<i>Logigramme</i>
LIST	<i>Liste d'instruction</i>
GRAPH	<i>Graphique</i>
GS	<i>GRAFCEt de Sécurité</i>
GPN	<i>GRAFCEt de Production Normale</i>
FB	<i>Bloc Fonctionnel</i>
OB	<i>Bloc d'organisation</i>
DB	<i>Bloc de données</i>
IHM	<i>Interface Homme-Machine</i>
SCADA	<i>Supervisory Control And Data Acquisition</i>
MPS	<i>Modular Production System</i>
CP	<i>Communication Processor</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
PG	<i>Programming Terminal</i>
WinCC	<i>Windows contrôle center</i>

INTRODUCTION GÉNÉRALE

Dans le monde du vingt et unième siècle, les concurrences, les compétitions et les exigences sont de plus en plus rudes et strictes entre les marchés et dans toutes les disciplines et surtout dans le secteur industriel et manufacturier, pour cela et pour avoir la quantité et la qualité de produits demandées et exigées par les clients, les industriels doivent moderniser leurs installations, leurs ateliers et leurs machines pour qu'ils puissent suivre les développements technologiques rapides et gigantesques.

À cet égard, les industries modernes doivent avoir un degré de flexibilité et d'automatisation très important, donc l'obligation d'utiliser des équipements et des systèmes modernes et intelligents tels que les systèmes de contrôles et de régulations industriels (PID), les systèmes de contrôle distribué (DCS) et les automates programmables industriels (API), sans oublier d'avoir des systèmes d'acquisition de données et de supervision SCADA performants afin de commander et de superviser tous les paramètres et les variables en temps réel et avoir une vision globale du comportement de toute l'installation industrielle.

Le travail réalisé dans ce mémoire porte sur la conception, la modélisation et la programmation d'un système automatisé de production avec le développement d'IHMs de supervision du fonctionnement du système étudié. Pour cela nous allons développer une nouvelle variante du système MPS de FESTO qui se compose de quatre stations qui sont la station de distribution, d'usinage, de manutention et de livraison, ces stations se trouvent dans le laboratoire de recherche de productique MELT de l'université de Tlemcen. Le système MPS de FESTO est considéré comme un petit atelier flexible de production didactique et de recherches scientifiques dans le domaine du génie industriel. Le système est très moderne, il contient quatre stations divisées en deux sous stations chacune. Toutes ces sous-stations sont équipées d'automates programmable SIEMENS, il existe deux autres stations, la station qualité équipée aussi d'API SIEMENS et une station robotisée qui contient un robot manipulateur anthropomorphe MITSUBISHI.

Ce mémoire comporte quatre chapitres :

Le premier chapitre est consacré à donner un aperçu général et global sur les systèmes automatisés de production SAP et les automatismes. Nous allons présenter la composition et l'architecture générale des systèmes automatisés de production où nous allons

d'abord parler des principaux composants de la partie opérative tel que les capteurs, les actionneurs, les pré-actionneurs et les effecteurs, ensuite nous allons présenter la partie commande et ces différentes composantes, en commençant par les automates Programmables industriels où nous allons parler de leurs architectures, leurs types, structures internes ainsi que leurs composantes, ensuite nous parlons des logiciels utilisés pour la configuration et la programmation des automates en donnant un aperçu sur les langages de programmation de ces derniers. Nous terminons ce chapitre par une conclusion.

Le second chapitre est dédié à la supervision des systèmes industriels, où nous allons donner la structure globale et les différentes techniques utilisées dans un système de supervision comme l'acquisition des données, la détection, le diagnostic, l'aide à la décision, la maintenance. Ensuite, nous allons entamer les systèmes SCADA, parler de leurs évolutions, leurs architectures et leurs composants de base. En fin, nous présentons les interfaces homme-machine (IHM), où nous présentons les différentes fonctionnalités des IHMs, leurs avantages, donner la différence entre l'IHM et le SCADA. Nous parlons aussi de la conception des IHMs, de leurs types, de connexion des IHMs avec les APIs et nous terminons par les applications des IHMs.

Dans la première partie du troisième chapitre, nous exposons et nous décrivons le système MPS variant que nous allons proposer à partir du système MPS de FESTO original. Nous allons présenter tous d'abord les quatre sous-stations qui forment le nouveau système de production automatisé MPS variant, en suite, nous présentons le cahier de charge et le fonctionnement du nouveau système proposé. Dans la deuxième partie de ce chapitre, on va réaliser suivant le cahier de charge proposé la configuration et les programmes de chaque sous-station sur le logiciel STEP 7 de SIEMENS. Les programmes seront simulés tout d'abord sur le logiciel S7-PLCSIM qui fait partie de la suite des logiciels de SIEMENS, puis on va ajouter l'équipement EasyPort USB qui remplacera l'API. Dans cette partie, nous allons connecter l'EasyPort USB avec le PC, injecter les programmes effectués et voir si le fonctionnement du système est correct. Nous terminons le chapitre par la présentation des résultats.

Le quatrième et dernier chapitre sera consacré au développement d'IHMs pour la supervision du système MPS variant proposé. Pour cela, nous allons utiliser le logiciel d'ingénierie, de conception et de configuration d'IHM WinCC Flexible du constructeur SIEMENS. Au préalable, nous allons présenter le logiciel WinCC Flexible et son interface, puis nous allons présenter en détail les différentes étapes utilisées pour la création d'un projet de supervision, comment créer une IHM et comment intégrer un projet STEP 7 avec WinCC Flexible. En fin de chapitre, nous allons présenter les simulations des interfaces WinCC Flexible de toutes les sous stations du système MPS variant étudié.

CHAPITRE

1

UN APERÇU SUR L'AUTOMATISME

1.1 Introduction

Aujourd'hui, le développement des technologies a permis d'introduire des systèmes d'automatisation industrielle dans presque tous les domaines de la fabrication.

Un système est considéré comme automatisé quand le processus permettant de passer d'une position initiale à une position finale s'effectue sans aucune intervention humaine, et ce comportement se répète chaque fois que les conditions qui caractérisent la position initiale sont satisfaites.

L'objectif principal d'un système d'automatisation est de réduire l'intervention humaine. Un opérateur humain est sujet à des erreurs et à la fatigue, ce qui peut entraîner divers problèmes. L'adaptation d'un système d'automatisation produira des avantages substantiels sur le rendement, le taux de production, la sécurité et la qualité.

1.2 Historique de l'automatisme

On croit que le terme automatisation a été inventé dans les années 1940 par un ingénieur de Ford Motor Company décrivant divers systèmes avec lesquels les actions et les contrôles automatiques ont été substitués à l'effort humain et l'intelligence.

À cette époque, les dispositifs de commande étaient de nature électromécanique. La logique a été effectuée au moyen de relais et de minuteriers reliés à la rétroaction humaine aux points de décision. En câblant les relais, les minuteriers, les boutons poussoirs et les capteurs de position mécaniques ensemble, des séquences de mouvement logiques simples pourraient être réalisées en allumant et éteignant les moteurs et les actionneurs[1].

1.3 C'est quoi l'automatisation ?

Après avoir consulté plusieurs documents, plusieurs livres, articles et sites web, nous avons trouvé plusieurs définitions de l'automatisation, dont on peut citer :

- L'automatisation veut dire l'utilisation des procédés visant à réduire ou remplacer la prise de décision et les activités manuelles de commande et les réponses des êtres humains, dans tout le processus de fabrication, et à perfectionner l'utilisation des matières et des énergies par l'application de nouvelles technologies[2].

- Autrement dit, l'automatisation, c'est de transférer tout ou partie des tâches de coordination, précédemment réalisés par des exploitants humains, dans un ensemble d'éléments techniques appelé partie commande (P.C). Cette partie préserve les connaissances pratiques pour les exploitants afin d'acquérir la séquence d'actions à mener sur les matériaux de travail dans le but de clarifier la valeur ajoutée. Il utilise un ensemble d'informations collectées par la partie opérative (P.O) pour développer la séquence de commandes nécessaires pour obtenir les actions requises. Donc l'automatisation permet généralement de rendre les entreprises et les personnes plus concurrentielles.

Le type de systèmes ou d'entreprise qui utilise l'automatisation et les automatismes sont dits des Systèmes Automatisés de Production (SPA)[2][3].

1.4 La cible à atteindre via de l'automatisme

L'automatisme a ciblé plusieurs objectives dans les SPA quelle que ce soit leur nature, nous vous citons quelques objectives qui sont touchées :

- La recherche des coûts plus bas pour le produit en réduisant les frais de main d'œuvre, économiser les matières premières et l'énergie, etc[4].
- La recherche d'une meilleure qualité du produit et l'augmentation de la productivité en limitant l'intervention humaine, l'enlèvement de tâches répétitives.
- Éliminer le travail dangereux ou difficile, compliqué, ainsi simplifier les efforts humains et faire accroître la sécurité via l'amélioration des conditions de travail[5].

1.5 Système automatisé

Un système automatisé est l'ensemble qui remplace un humain en décidant et en prenant des mesures après avoir reçu des instructions d'un opérateur. Cette substitution permet la mise à disposition de main-d'œuvre, du temps et des coûts, tout en améliorant l'exactitude et la précision du travail effectué. Il améliore l'accès, l'efficacité et la fiabilité des services fournis[6].

1.6 Domaines d'application d'un système automatisé

Avec le développement très rapide des technologies comme l'informatique, la robotique, les systèmes électroniques embarqués entre autres, les SPA sont devenue une obligation et une nécessité.

On les trouve dans tous les types d'industrie moderne pour réaliser des emplois dangereux ou particulièrement douloureux ou répétitifs[6].



FIG. 1.1 : Chaîne de production en série automatisée destinée à souder la carrosserie des véhicules[6].

Nous trouvons aussi les systèmes automatisés dans l'organisation et la gestion de la circulation et des systèmes de transport comme les trains, les métros, feux de circulation, etc.



FIG. 1.2 : Les systèmes automatisés dans l'organisation de la circulation[6].

Nous trouvons les systèmes de production automatisés dans le domaine du bâtiment et de la construction, dans les systèmes de chauffage et de la climatisation, la distribution de l'électricité, la sécurité, les ascenseurs, ...

1.7 Anatomie d'un système automatisée

Un système automatisé est généralement composé de deux parties qui coopèrent : Partie Commande (P.C) et une Partie Opérative (P.O). L'architecture d'un système au-

tomatisé peut se décomposer selon la figure 1.3.

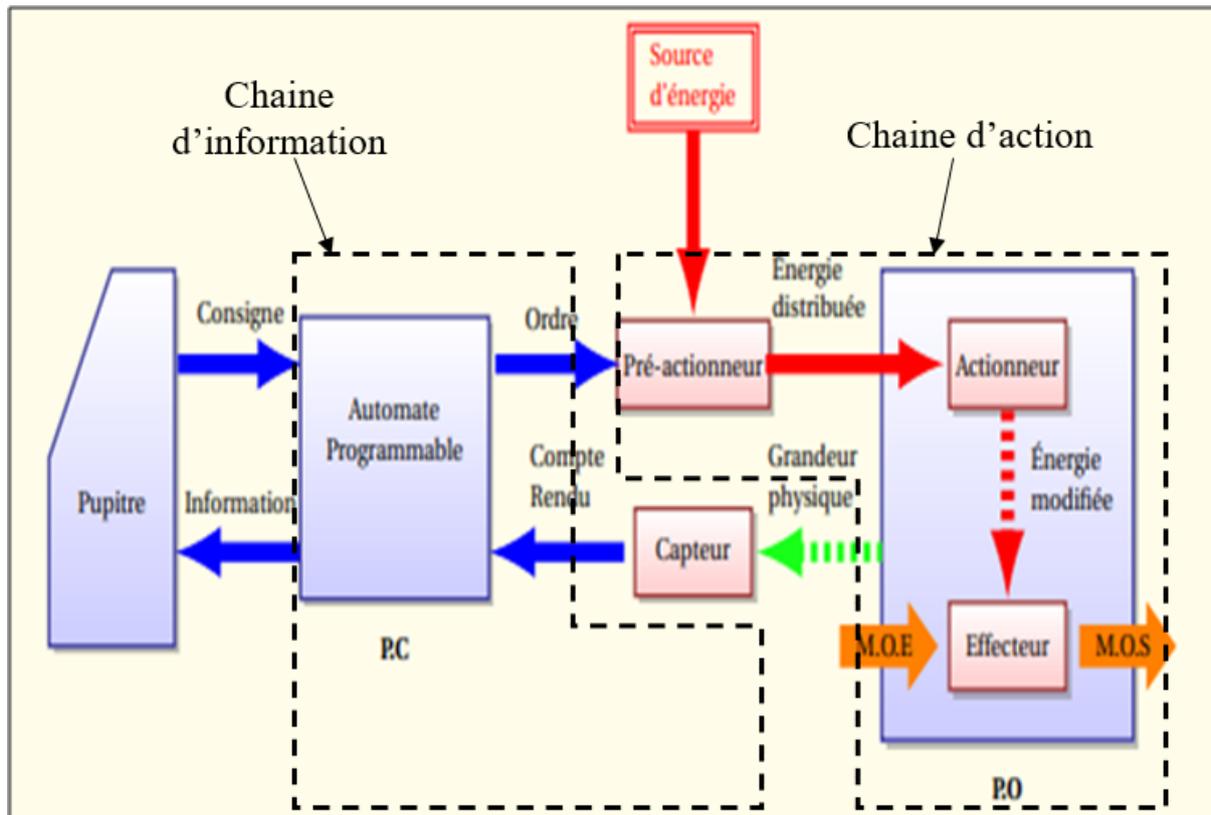


FIG. 1.3 : Planification de fonctionnement d'un système automatisé[2].

1.7.1 Partie Commande (P.C)

Elle joue le rôle du cerveau de tous les systèmes automatisés, elle reçoit les informations et les données des capteurs et selon les programmes écrits dans les automates programmables elle détermine des ordres à envoyer aux pré-actionneurs et actionneurs pour garantir le pilotage et la coordination des tâches du processus durant son fonctionnement[4][7]. Toutes les données qui entrent dans la partie opérative sous formes d'informations viennent à partir des capteurs, des détecteurs et des consignes qui représentent les entrées et qui sont transmises vers cette même (PC)[8]. La partie commande se compose de [6] :

- L'automate ou bien l'unité de traitement.
- Les interfaces d'entrées qui convertit les informations venant des capteurs placés sur la partie opérative en informations d'amplitude et de nature accordable avec les caractéristiques technologiques de l'automate.
- Les interfaces de sorties qui convertit les informations produit par l'unité de traitement en informations d'amplitude et de nature accordable avec les caractéristiques technologiques de pré-actionneurs.

1.7.2 Partie Opérative (P.O)

Cette partie reçoit les ordres de la partie commande, et qui vont adresser des comptes rendus grâce aux actionneurs. La P.O est composé des pré-actionneurs, des actionneurs, des effecteurs et des capteurs qui permettant de mesurer l'état du système. Elle reçoit des messages et envois des consignes vers la partie commande. La partie opérative consomme l'énergie électrique, hydraulique ou pneumatique. C'est la partie visible du système[2][8]. La partie opérative est composée des[6] :

- Effecteurs qui ont une relation directe avec la matière à traiter.
- Actionneurs qui fournissent à l'unité de production des sources d'énergie utilisable et exploitable grâce à la transformation de cette énergie mouvements utiles pour la production.
- Pré-actionneurs qui ont comme rôle la mise en énergie des actionneurs avec les différents types d'énergie (électrique, hydraulique, pneumatique).
- Capteurs qui créent, à partir de différents types d'informations, l'entrée de la partie commande.

1.7.2.1 Une vue approfondie sur la partie opérative PO

1.7.2.1.1 Les effecteurs

Un effecteur est un assemblage qui utilise de l'énergie soit de lui-même ou bien à l'aide d'un actionneur, afin de produire un effet utile sur la matière d'œuvre en lui donnant une certaine valeur ajoutée[6][7]. Nous pouvons trouver les effecteurs dans : la production, la manutention, l'assemblage, le remplissage, l'emballage.

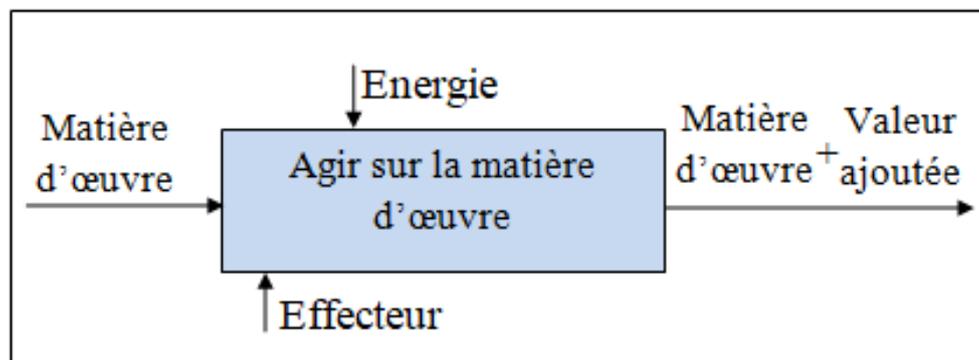


FIG. 1.4 : Schéma de fonctionnement d'un effecteur[7].

1.7.2.1.2 Les actionneurs

L'actionneur représente un élément de la partie opérative qui reçoit des ordres de la partie commande, afin de fournir une puissance (moteurs, vérins, résistances, électroaimants). Ces actionneurs transforment l'énergie d'entrée (électrique, pneumatique, hydraulique) en énergie de sortie utilisable par un effecteur pour l'exploiter sous une action bien

définie selon l'utilité, le plus souvent en énergie mécanique (la rotation, la translation, l'aspiration, ...) comme nous pouvons la voir sous forme de l'énergie thermique, sonore ou lumineuse pour obtenir une action définie[9]. Un actionneur est un changeur électromécanique désigné pour mettre en mouvement un système mécanique à partir d'une commande électrique ou autre pour transmettre une énergie en une autre[10].

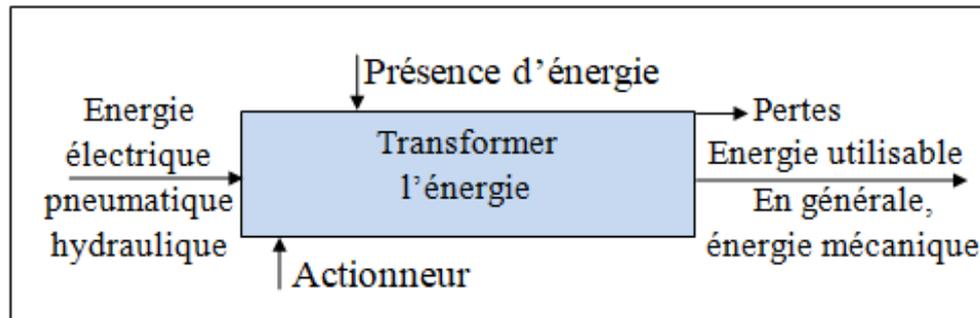


FIG. 1.5 : Schéma de fonctionnement d'un actionneur[7].

A. Les types d'actionneurs[11][12]

Les actionneurs sont classés selon l'énergie utilisée, électrique, pneumatique, hydraulique.

A.1 Les actionneurs électriques

Convertissent l'énergie électrique en énergie mécanique. Il existe plusieurs types d'actionneurs électriques :

- Les moteurs

Produire un mouvement de rotation par transformation de l'énergie électrique en énergie mécanique, par le biais de l'axe de sortie du moteur. Exemples de différents moteurs :

1. Les moteurs asynchrones triphasés.
2. Les moteurs à courant continu (MCC).
3. Les moteurs pas à pas.
4. Servomoteur.

- La résistance chauffante

Lorsqu'un courant électrique circule dans un élément résistif, nous obtiendrons la transformation de l'énergie électrique en énergie calorifique (chaleur).

- L'électroaimant

Il transforme l'énergie électrique en énergie mécanique.

Pour plus de résumer, nous avons choisi la figure 1.6 ci-dessous comme synthèse qui englobe quelques actionneurs électriques.

			
Moteur asynchrone triphasé.	Moteur à courant continu (MCC).	Moteur pas à pas.	Servomoteur.
			
Résistance chauffante.		L'électroaimant.	

FIG. 1.6 : Les différents actionneurs électriques[11][12][13].

A.2 Les actionneurs pneumatiques

Un actionneur pneumatique est un dispositif qui convertit l'énergie de l'air comprimé en travail mécanique. Parmi les actionneurs pneumatiques les plus utilisés dans les systèmes automatisés il y a trois types :

1. Les vérins pneumatiques.
2. Les ventouses pneumatiques.
3. Les moteurs pneumatiques.

		
Vérin pneumatique.	Ventouse.	Moteur pneumatique.

FIG. 1.7 : Les différents actionneurs pneumatiques[12][14].

A.3 Les actionneurs hydrauliques

Un actionneur hydraulique est un dispositif qui convertit l'énergie hydraulique en énergie mécanique. Parmi ces actionneurs hydrauliques :

1. Les vérins hydrauliques.
2. Les pompes.



FIG. 1.8 : Exemple d'un Vérin hydraulique[12].

1.7.2.1.3 Les pré-actionneurs

C'est l'interface de puissance entre la partie commande et la partie opérative. C'est un élément dont le rôle est de distribuer de l'énergie quelle que ce soit son type via un ordre de la partie commande. Ils commandent exclusivement l'établissement et l'interruption, la circulation de l'énergie entre une source et un actionneur[7].

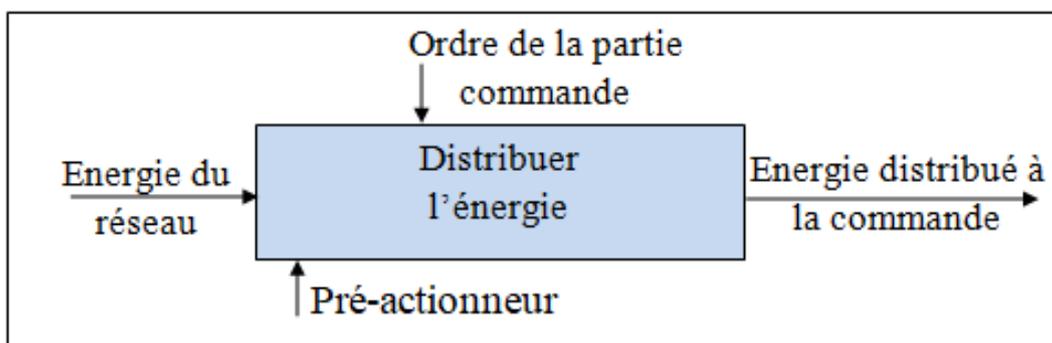


FIG. 1.9 : Schéma de fonctionnement d'un pré-actionneur[7].

Les pré-actionneurs les plus utilisés sont[10] :

1. Les contacteurs, les relais électromagnétiques : pour les actionneurs de type électriques.
2. Les distributeurs : pour les actionneurs de type pneumatiques ou hydrauliques.

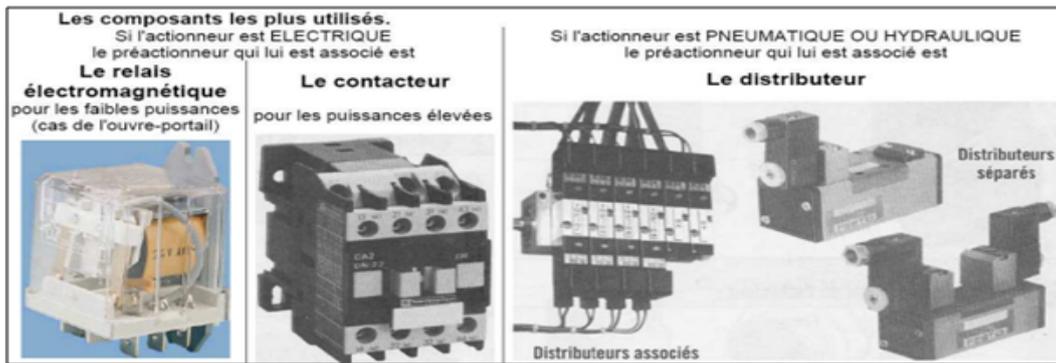


FIG. 1.10 : Les différents pré-actionneurs[7].

1.7.2.1.4 Les capteurs

Un capteur est un dispositif qui détecte une grandeur physique, puis il va la traduire en grandeur électrique[15].

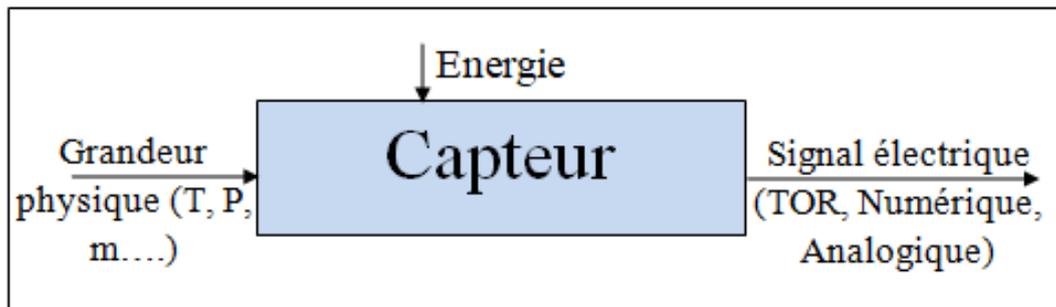


FIG. 1.11 : Principe de fonctionnement d'un capteur[16].

La nature de type de sortie d'un signal électrique dans un capteur :

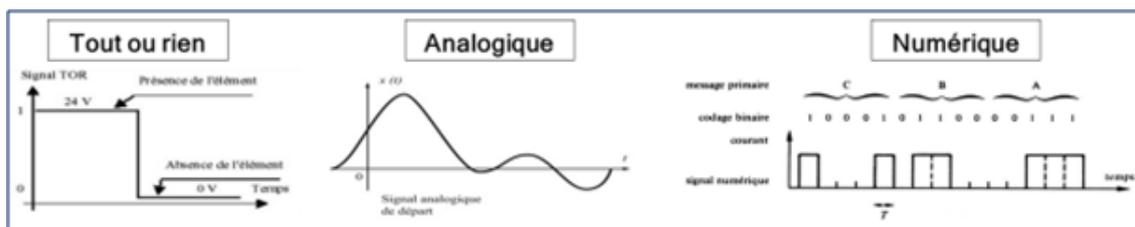


FIG. 1.12 : Les types de l'information des capteurs[15].

a. Classification des capteurs

Il existe deux types de capteur actif est passif :

- Capteurs actifs

Il génère un signal électrique sous l'effet de grandeur physique, il ne nécessite pas une source d'alimentation extérieure[15]. Exemple : thermocouples, capteur de vitesse (phototransistors), photodiodes.

- **Capteurs passifs**

Un capteur dont laquelle la variation de la grandeur physique est représenté par une variation de l'impédance, afin d'exploiter cette variation il faut une source d'alimentation extérieure[15]. Exemples : inductance de fin de course, thermistance, capteur de niveau capacitif, résistance à fil de platine (sonde Pt100).

- b. Les différents capteurs**

- b.1. Capteur de proximité**

Il est utilisé pour détecter la présence des objets. L'interaction entre le capteur et sa « cible » est traduite le plus souvent par un champ (magnétique, électrique, électromagnétique) ou par un capteur infrarouge, ou par une caméra associée à un système d'analyse de l'image. L'élément détecté doit être proche du capteur ou éclairer par une source rayonnante. Les capteurs de proximité sont utilisés en mode analogique ou binaire [10][17].

- **Capteur de proximité inductif**

Un capteur de proximité inductif est un capteur qui génère un champ magnétique à son extrémité et peut détecter tout objet conducteur situé à une certaine distance, selon le type de capteur. Les capteurs inductifs génèrent un champ magnétique oscillant à l'extrémité de leur tête de détection. Ce champ est généré par une self et un condensateur installés en parallèle. Lorsqu'un objet métallique pénètre dans ce champ, le champ est perturbé puis le champ oscillatoire décroît. Ce changement est exploité par un amplificateur qui fournit un signal de sortie, le commutateur du capteur[16].



FIG. 1.13 : Divers types de capteurs inductifs[16]

- **Capteur de proximité capacitif**

Les capteurs capacitifs sont des capteurs de proximité qui détectent les objets métalliques ou isolants. Lorsqu'un objet entre dans le champ de détection des électrodes sensibles du capteur, il cause des oscillations en modifiant la capacité de couplage du condensateur. Le principe de fonctionnement des capteurs de proximité capacitifs est

basé sur l'utilisation du changement de capacité d'un condensateur dans un circuit oscillant RC. Une fois que le matériau s'approche du capteur de proximité, la capacité du condensateur augmente. Il en résulte un changement utilisable dans l'oscillation du circuit RC. Ce changement de capacité dépend principalement de la distance, de la taille et de la constante diélectrique du matériau[16].



FIG. 1.14 : Capteur de proximité capacitif[16].

b.2. Capteur photoélectrique

Les cellules photoélectriques, ou capteurs photoélectriques comme on les appelle également, contiennent un émetteur qui envoie de la lumière et un photo-détecteur (récepteur) qui détecte la lumière entrante. Les cellules photoélectriques existent en trois versions principales : à barrage, réfléchir et rétro-réfléchir. La différence entre ces versions réside dans l'emplacement du photo-détecteur (récepteur) et dans la capacité de l'objet à détecter, à réfléchir la lumière[18].

- **Capteur photoélectrique à barrage**

Ce type de capteur est basé sur un émetteur et un récepteur dans des unités séparées qui sont placées de chaque côté de l'objet à détecter. Lorsqu'un objet passe entre les unités, le faisceau lumineux ne peut atteindre le photo-détecteur. L'avantage de ce type est que la distance de détection peut être grande (plusieurs dizaines de mètres, selon la lentille de l'émetteur), même s'il est très difficile d'installer un émetteur et un récepteur en ligne lorsque la séparation atteint 10 m ou plus. La difficulté réside dans le fait qu'il existe deux unités, qui nécessitent toutes deux une alimentation électrique[18].

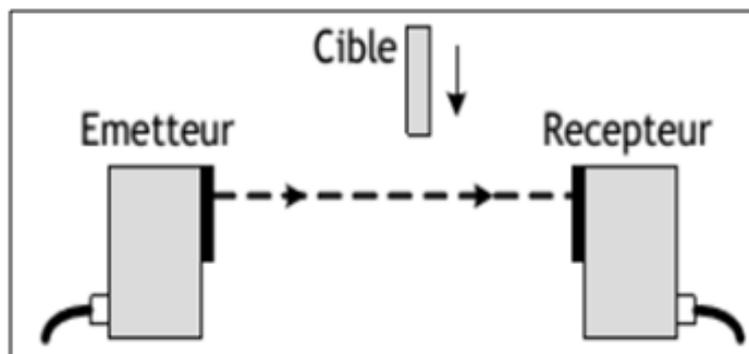


FIG. 1.15 : Principe de fonctionnement d'un capteur photoélectrique à barrage[19].

- **Capteur photoélectrique réfléchissant**

Dans le type réfléchissant, le photo-détecteur est intégré dans la même unité que l'émetteur. Si l'objet est d'une nature telle qu'il peut réfléchir la lumière, le photo-détecteur enregistrera le passage de l'objet. La forme et la couleur de l'objet, ainsi que la couleur de ce qui se trouve derrière l'objet, jouent un rôle important dans le bon fonctionnement du détecteur. Pour une séparation donnée, un objet blanc peut être plus simple à détecter qu'un objet gris ou noir. Si l'arrière-plan est d'une couleur plus claire que l'objet, la détection peut être quasi impossible[18].

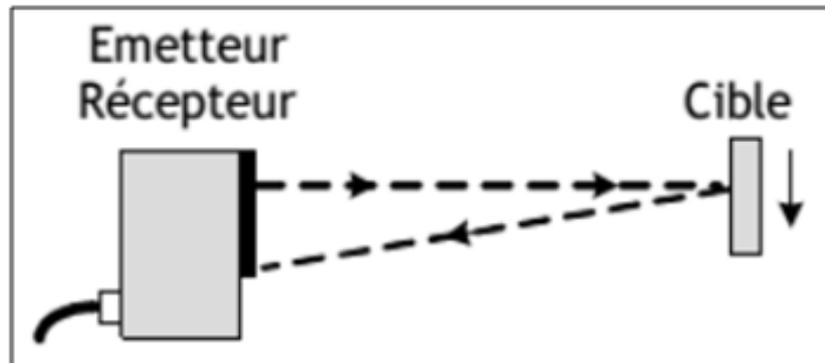


FIG. 1.16 : Principe de fonctionnement d'un capteur photoélectrique réfléchissant[19].

- **Capteur photoélectrique rétro-réfléchissant**

Si l'objet ne convient pas à la lumière réfléchi, une surface réfléchissante peut être installée sur le côté opposé. Dans ce cas, le photo-détecteur recevra de la lumière jusqu'à ce qu'un objet interrompe le faisceau. Cette variante est appelée rétro-réfléchissant. Ici aussi, l'écart de détection peut être très important, jusqu'à 20 m. L'utilisation de ce type peut être difficile si l'objet présente une surface lisse ou réfléchissante, comme une fenêtre ou un objet métallique poli. Le capteur peut alors confondre l'objet avec la surface réfléchissante[18].

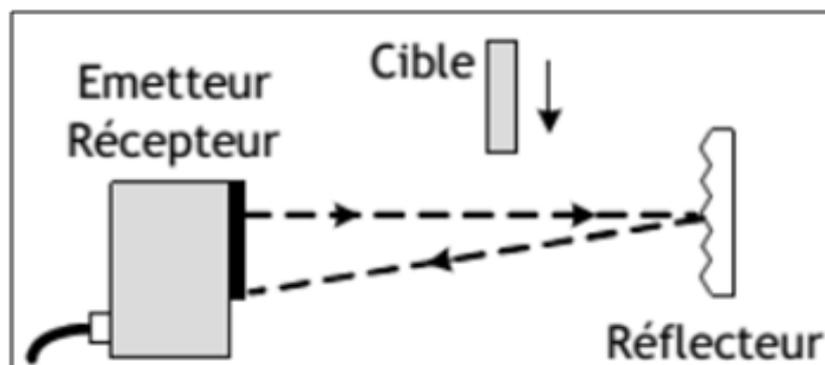


FIG. 1.17 : Principe de fonctionnement d'un capteur photoélectrique rétro-réfléchissant[19].

b.3. Capteur de pression

Capteur de pression est un dispositif destiné à transférer les variations de pression en variations de tension électrique. Ils sont utilisés dans de nombreuses applications de systèmes de contrôle ou de régulation[10].

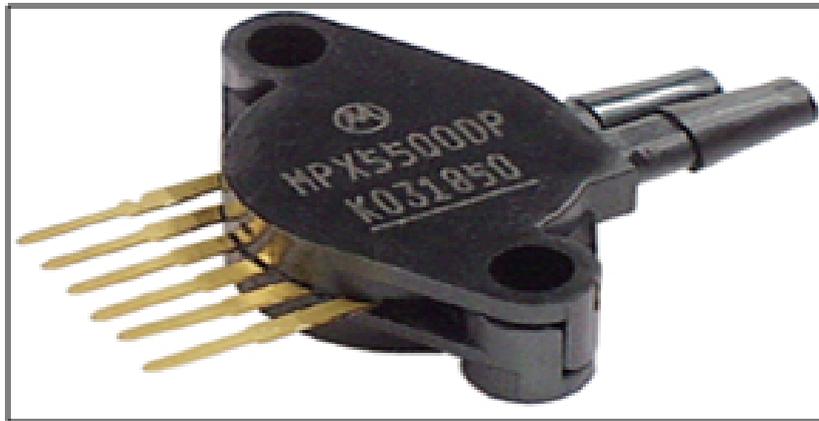


FIG. 1.18 : Capteur de pression[20].

b.4. Capteur magnétique (Capteur à ampoule REED)

L'ampoule REED ou l'interrupteur à lames souples est constitué de deux ou trois lames ferromagnétiques enfermées dans un tube de verre rempli de gaz inerte, qui entrent en contact entre eux par le biais d'un champ magnétique[21].

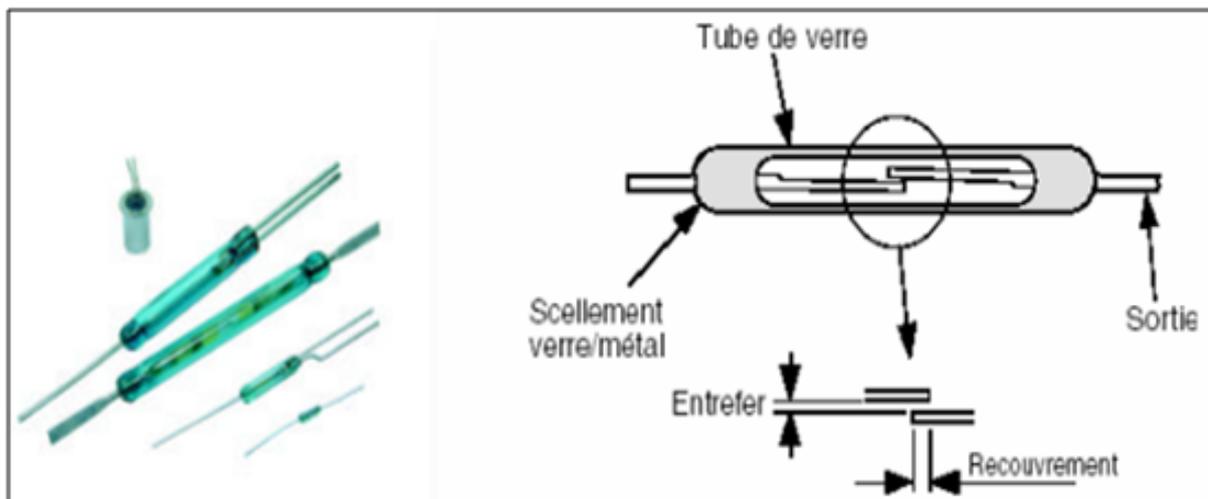


FIG. 1.19 : Exemple d'un capteur à ampoule REED[21].

b.5. Capteur de position mécanique (fin de course)

Un capteur est dit mécanique s'il est actionné par des quantités physiques telles que la pression, la température ou la position[17].



FIG. 1.20 : Capteur de position mécanique (fin de course)[22].

b.6. Capteurs de température (Thermocouple)

Deux métaux A et B de natures différentes, sont reliés par deux jonctions (formant ainsi un thermocouple). Par l'effet Seebeck, le thermocouple génère une différence de potentiel qui dépend de la différence de température entre les jonctions. Les thermocouples ne mesurent pas une température, mais une différence de température[20].

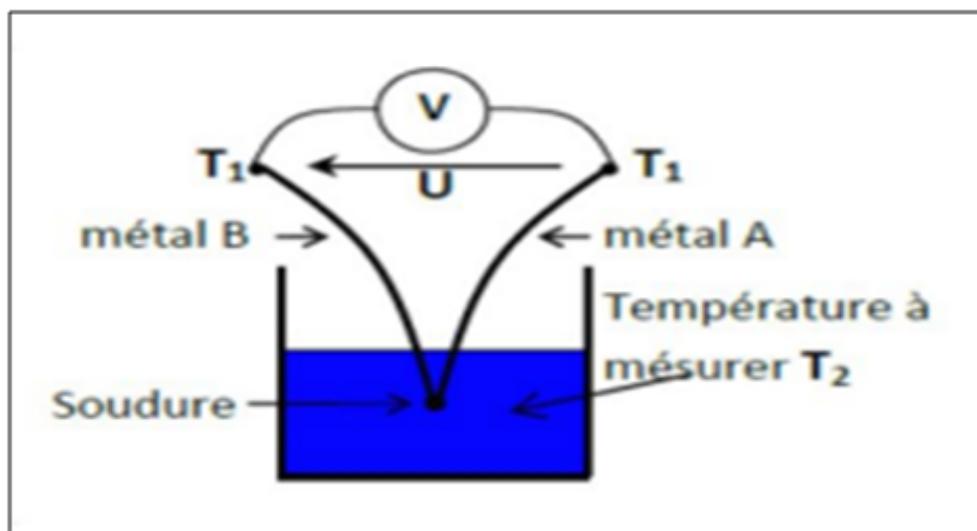


FIG. 1.21 : Principe de fonctionnement d'un thermocouple[15].

1.8 Les différentes technologies utilisées dans un système automatisé

Il existe deux types de technologies :

1.8.1 Logique câblée

C'est l'ancienne technologie des automatismes dans l'industrie, elle met en œuvre des contacts, des relais, des bobines, etc. sous l'assemblage d'une armoire électrique. Cette technologie est inutilisée maintenant au profit des technologies plus modernes et flexibles[23].

1.8.2 Logique programmée

La logique programmée nécessite les outils d'informatiques qui sont venus à la place de la logique câblée, celle-ci est très facilement adaptable aux besoins et aux évolutions du processus, c'est-à-dire changement du mode de commande qui peut se faire par un simple changement du programme, elle est de plus en plus systématiquement employée grâce à l'aspect de programmation de la loi de commande. Parmi les solutions qui se trouvent maintenant dans l'industrie[6][23] :

- Automate programmable industriel API.
- Régulateur industriel.
- Microcontrôleur.

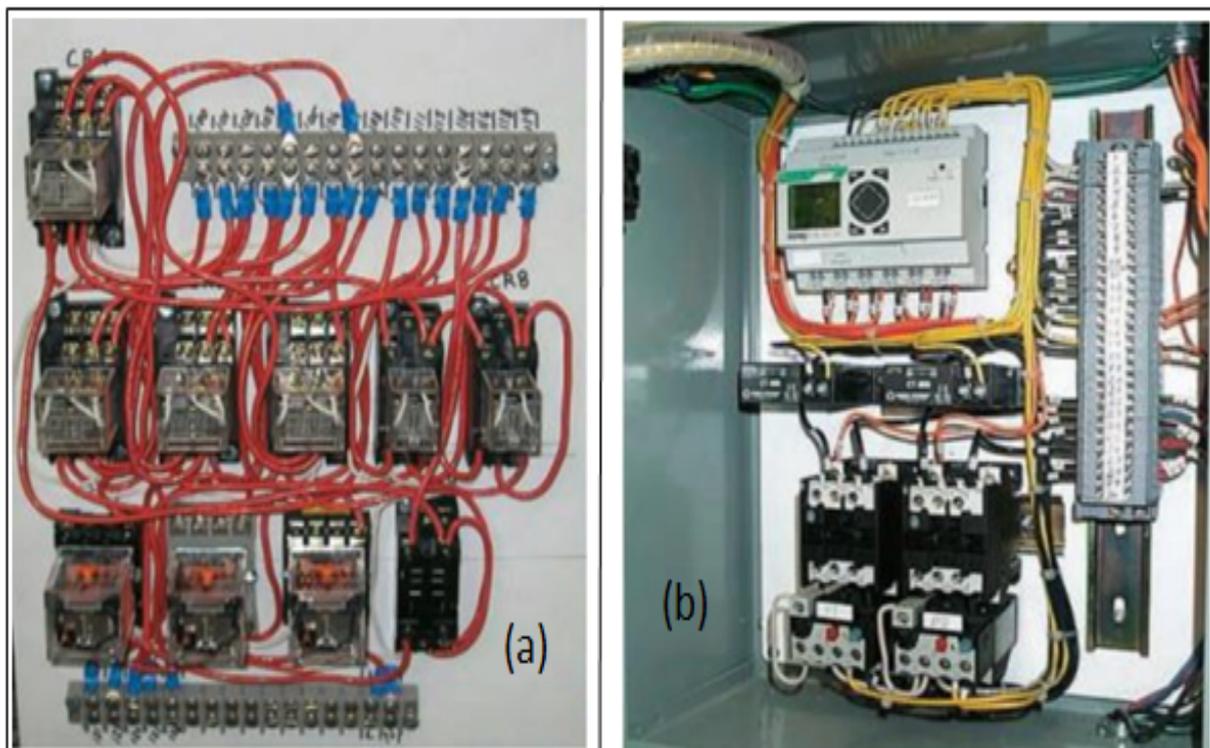


FIG. 1.22 : Différence entre armoire à relais et armoire à API[24]

1.9 Généralités sur les automates programmables industriels (API)

1.9.1 Historique[18]

Les API ont été introduits pour la première fois à la fin des années 1960. Leur invention était de remplacer les grands circuits de relais séquentiels pour le contrôle des machines.

Les premiers API étaient relativement simples en ce sens que leur fonction était de remplacer la logique de relais et rien d'autre. Progressivement, leurs capacités se sont améliorées de plus en plus et des fonctions telles que des compteurs et des délais ont été ajoutés. L'étape suivante du développement a été l'entrée/sortie analogique et les fonctions arithmétiques telles que les comparateurs et les additionneurs.

Avec le développement de la technologie des semi-conducteurs et des circuits intégrés, les commandes programmables sont devenues largement utilisées dans l'industrie. En particulier, lorsque les microprocesseurs sont entrés sur le marché au début des années 70, le développement s'est accéléré.

Les API d'aujourd'hui viennent avec des outils de développement sous forme de logiciel avec chaque fonction prête à l'emploi imaginable. Par exemple, les codes de programme pour la gestion des communications ainsi que les fonctions de traitement telles que les régulateurs intégrateurs/dérivés proportionnels, les servocommandes, le contrôle axial, etc. En d'autres termes, le rythme de développement est le même qu'avec le PC.

1.9.2 Définition d'un Automate Programmable Industriel (API)

L'automate programmable industriel est un dispositif électronique sous une forme spéciale de microcontrôleur basé sur un processeur qui utilise une mémoire programmable pour stocker des instructions et pour mettre en œuvre des fonctions telles que la logique, le séquençage, le chronométrage, le comptage et l'arithmétique afin de contrôler les machines et les processus et est conçu pour être exploité par des ingénieurs ayant peut-être une connaissance limitée des ordinateurs et des langages informatiques.

Ils ne sont pas conçus pour que seuls les programmeurs puissent configurer ou modifier les programmes. La capacité d'un automate programmable se définit par trois éléments[25][26] :

- Le nombre d'entrées-sorties
- La capacité de la mémoire programme.
- Les fonctions auxiliaires de commandes : temporisateurs, compteurs, registres...

Dans le monde de l'industrie, on trouve plusieurs grandes marques des API's, parmi eux :

- SIEMENS
- Allen-Bradley

- Mitsubishi
- Schneider Electric
- Rockwell
- Delta, Moeller, Omron, IDEC



FIG. 1.23 : Automates programmables industriels de type "Siemens"[27].



FIG. 1.24 : Automates programmables industriels de type "Allen-Bradley"[24].

1.9.3 Présentation des API S7-300 de Siemens

C'est l'automate modulaire par excellence, offrant une grande souplesse d'utilisation et des performances particulièrement élevées. Il autorise un très large choix de modules d'E/S TOR, analogiques et de fonctions métier qui permettent l'adaptation optimale à toutes les

tâches d'automatismes. Pour le type de communication l'appareil S7-300 permet toutes les communications standards (AS-I, liaison série, Profibus, Profinet, Ethernet...)[28].

1.9.4 Architecture matérielle d'un API

1.9.4.1 Aspect extérieur

Il existe deux types courants de conception mécanique pour les systèmes API, le boîtier fixe (compacte) et les types modulaires/rack[26].

1.9.4.1.1 Configuration compacte

Dans cette configuration, nous trouvons la fusion des unités d'alimentation, le processeur, la mémoire, les entrées et les sorties dans un seul boîtier. C'est-à-dire que les entrées/sorties sont fixes, car dans ce cas, les entrées/sorties et le CPU sont packagés ensembles.

L'avantage de cette configuration, c'est le faible coût de l'API, mais l'inconvénient : Si une des unités tombe en panne, c'est tout le système qui doit être remplacé[24][26].



FIG. 1.25 : Automate programmable compact S7-1200[29].

1.9.4.1.2 Configuration modulaire

Le processeur, les unités d'alimentation et les entrées/sorties sont des modules séparés (modules) dans un ou plusieurs racks. C'est-à-dire l'API est divisée en plusieurs modules. La configuration modulaire permet d'avoir une grande flexibilité[24][26].

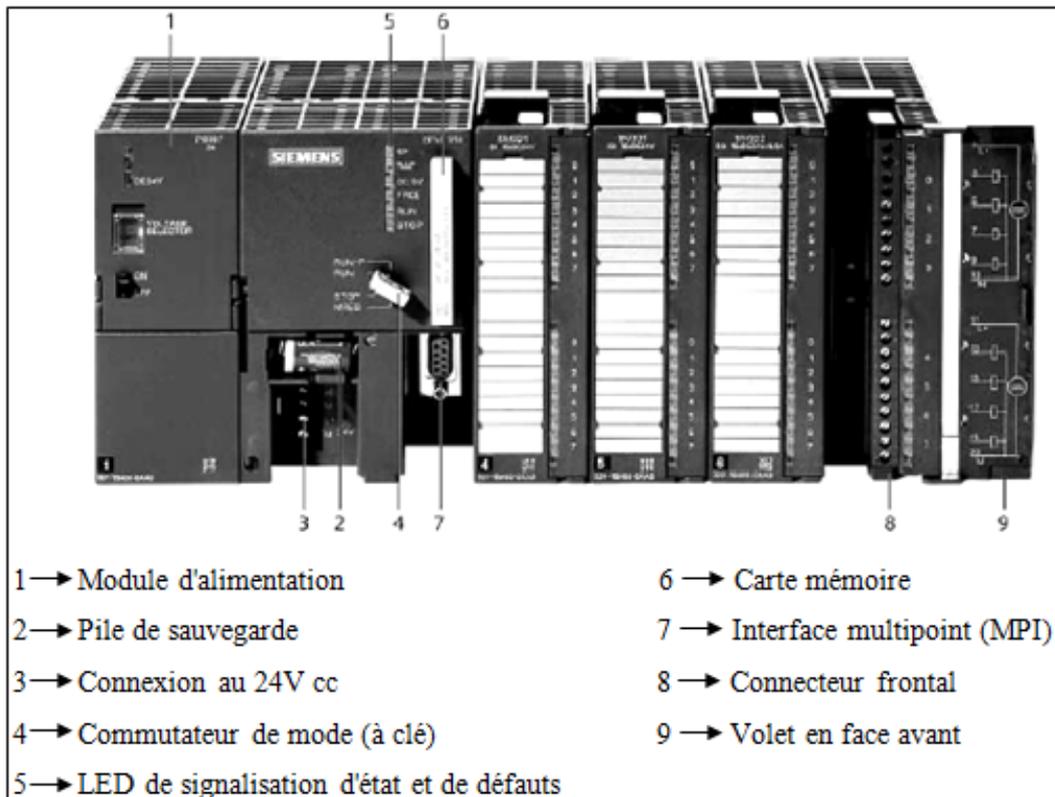


FIG. 1.26 : Automate modulaire s7-300[30].

1.9.4.2 Structure interne

En général, un système API comporte les composants fonctionnels de base suivants : unité de traitement, mémoire, module d'alimentation, les interfaces d'entrées/sorties, interface de communication et l'outil de programmation [26].

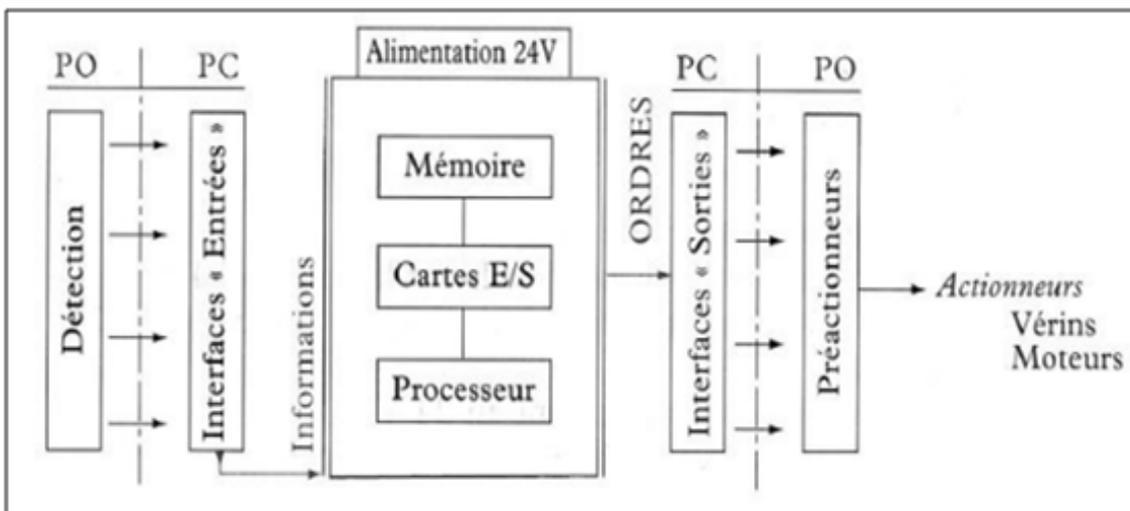


FIG. 1.27 : Structure interne d'un API[30].

1.9.4.2.1 Unité de traitement

C'est le cerveau de l'API. Ici sont effectués toutes les instructions et les calculs, et il contrôle le flux de l'information et comment le programme fonctionne. Normalement, le CPU fait partie du bloc physique et contient la mémoire, les ports de communication, les voyants d'état et parfois l'alimentation[18].



FIG. 1.28 : Module CPU Siemens (CPU 314)[31].

1.9.4.2.2 Module d'alimentation

Tous les API doivent être connectées à une alimentation. Habituellement, l'alimentation est un module interchangeable, mais certains API plus petits ont l'alimentation comme une partie intégrée du processeur et du module de communication. Ils Permettent de distribuer l'énergie aux différents modules. La plupart des fabricants fournissent donc des alimentations en plusieurs versions : 220V AC, 120V AC, et 24V DC[18].



FIG. 1.29 : Module d'alimentation PS 307-5A[32].

1.9.4.2.3 Modules d'entrées/sorties

Il s'agit du contact entre un API et le monde extérieur. Dans un API, toutes les entrées et les sorties se déroulent en blocs ou modules conçus pour recevoir différents types de signaux et transmettre des signaux dans différents formats. Il y a des blocs d'entrée pour les signaux numériques, les signaux analogiques, les éléments thermiques, les thermocouples et les encodeurs, etc. Il y a aussi des blocs de sortie pour les signaux numériques et analogiques ainsi que des blocs à des fins spéciales. Chaque entrée et sortie à une adresse unique qui peut être utilisée dans le code du programme [18]. Les modules d'entrées/sorties sont :

- Modules d'entrées TOR (Tout Ou Rien).
- Modules de sorties TOR (Tout Ou Rien).
- Modules d'entrées Analogiques.
- Modules de sorties Analogiques.

1.9.4.2.4 Module de communication

Cette unité intègre un ou plusieurs protocoles pour gérer les communications. Tous les automates ont une connexion pour un câble de programmation et souvent pour un panneau opérateur, une imprimante ou un réseau. Diverses normes physiques sont utilisées, tant pour le port de programmation que pour les ports de connexion à d'autres équipements. Les API actuels sont généralement programmées à partir d'un PC ordinaire avec un outil de programmation développé pour ce type particulier d'API. Il n'est pas toujours nécessaire d'avoir une connexion directe entre l'automate et le PC pour transférer le code du programme à l'automate. Toutefois, c'est actuellement l'approche la plus courante, du moins pour les petits systèmes. Parfois, la programmation peut être effectuée via un réseau composé de plusieurs automates et d'autres équipements ou via Ethernet. Certains automates disposent également d'un serveur web intégré. Le développement des bus d'instrumentation a permis aux fabricants d'automates de fournir des solutions intégrées ou modulaires pour les communications via un grand nombre de protocoles différents. Par exemple : Modbus, le bus AS-interface, PROFIBUS et PROFINET /Industriel Ethernet, CAN bus, etc.[18].

a. Modbus

Modbus est un protocole de communication pour les réseaux d'automates programmables industriels (API). Il fonctionne en mode maître/esclave pour échanger des trames. Le protocole Modbus peut être utilisé [33] :

- Directement sur une liaison série de types RS-422 ou RS-485 ou TTY (boucle de courant) à vitesse et distance variables.
- Via TCP/IP avec Ethernet, cela s'appelle Modbus TCP/IP ou Modbus TCP.

- Via Modbus Plus (ou Modbus+), est un réseau à passage de jetons à 1 Mb/s qui transmet des trames Modbus et d'autres services spécifiques au réseau.

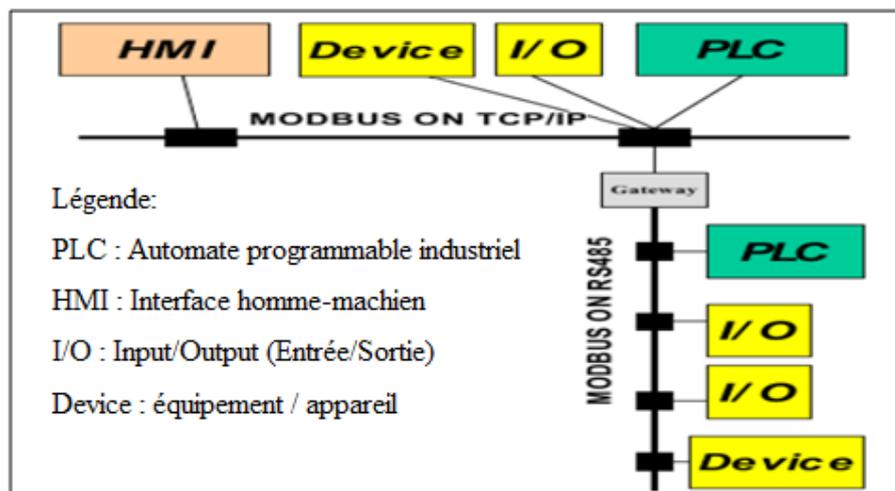


FIG. 1.30 : Exemple de réseaux Modbus[33].

b. Le bus AS-interface

Le protocole de réseau ASI est utilisé dans des applications de réseau simple et discret ne nécessitant pas plus de 124 dispositifs d'E/S sur le terrain. Ces 124 dispositifs d'entrée et de sortie peuvent être connectés à un maximum de 31 nœuds dans une topologie en arbre, en étoile ou en anneau. Les dispositifs d'E/S se connectent à l'automate ou à l'ordinateur personnel via le bus par l'intermédiaire d'une interface de contrôleur hôte. Les dispositifs typiquement compatibles avec le protocole ASI sont les détecteurs de proximité, les interrupteurs de fin de course, les capteurs photoélectriques et les dispositifs de terrain standard disponibles dans le commerce. Le système AS-i est constitué d'un maître, d'une alimentation et de stations connectées, appelées esclaves. Le maître AS-i gère l'échange de données avec les esclaves de manière cyclique (polling)[34].

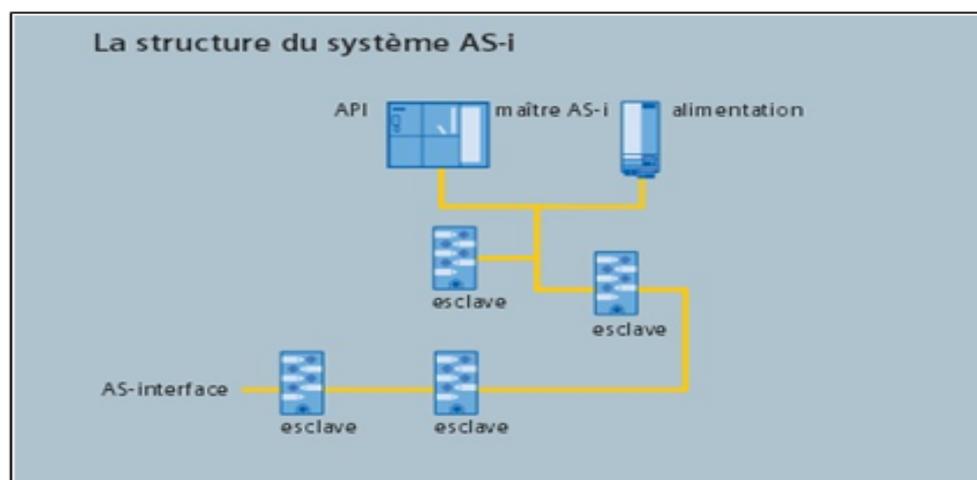


FIG. 1.31 : La structure du système AS-i[34].

c. PROFIBUS

Profibus (Process Field Bus) est un réseau de bus de processus numérique capable de communiquer des informations entre un contrôleur maître (ou hôte) et un dispositif de champ de processus intelligent et esclave, ainsi que d'un hôte à un autre. Profibus se compose en fait de trois réseaux inter-compatibles avec différents protocoles conçus pour répondre à des exigences d'application distinctes. Les trois types de réseaux Profibus sont les suivants[35] : Profibus-FMS (Fieldbus Message Specification), Profibus-DP (Decentralised Peripheral) et Profibus-PA (Process Automation),

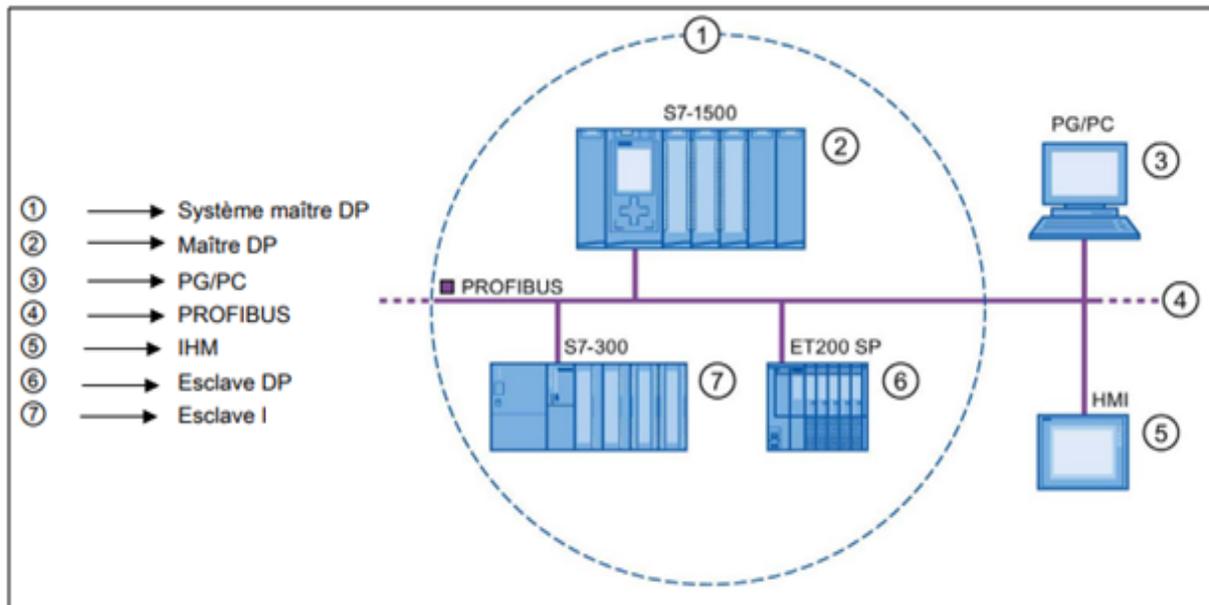


FIG. 1.32 : Les principaux composants dans PROFIBUS DP[36].

d. PROFINET /Industriel Ethernet

Profinet est une norme de communication industrielle pour un système de contrôle industriel. Comme Profibus, ce protocole de communication est principalement utilisé par les automates Siemens. Profinet est un système industriel basé sur Ethernet. En utilisant Profinet, nous pouvons connecter des équipements tels que PLC, HMI, Distributed I/O, différents types de transmetteurs, capteurs, actionneurs, etc. sur un seul réseau. Profinet offre un temps de réponse plus rapide, de sorte que la collecte des données devient encore plus importante. Le Profinet industriel est livré avec un blindage qui offre de meilleures performances dans un environnement électriquement bruyant[37]. Une des applications de PROFINET est de remplacer PROFIBUS par un réseau basé sur Ethernet. Toutes les interfaces proposées pour PROFIBUS sont également proposées pour PROFINET. L'avantage évident est le débit de données plus élevé d'Ethernet et son extension avec le câblage en fibre optique et, dernièrement, avec la transmission sans fil Wi-Fi. Il existe des versions spécifiques de PROFINET pour les applications de sécurité, et pour les applications synchrones à haute vitesse dans le contrôle des machines avec lesquelles PROFINET-RT peut être utilisé, ou pour les applications à très haute vitesse dans l'usinage des métaux où une variation matérielle spéciale de la puce PROFINET est utilisée pour supporter PROFINET-iRT, isochrone, temps réel[38].

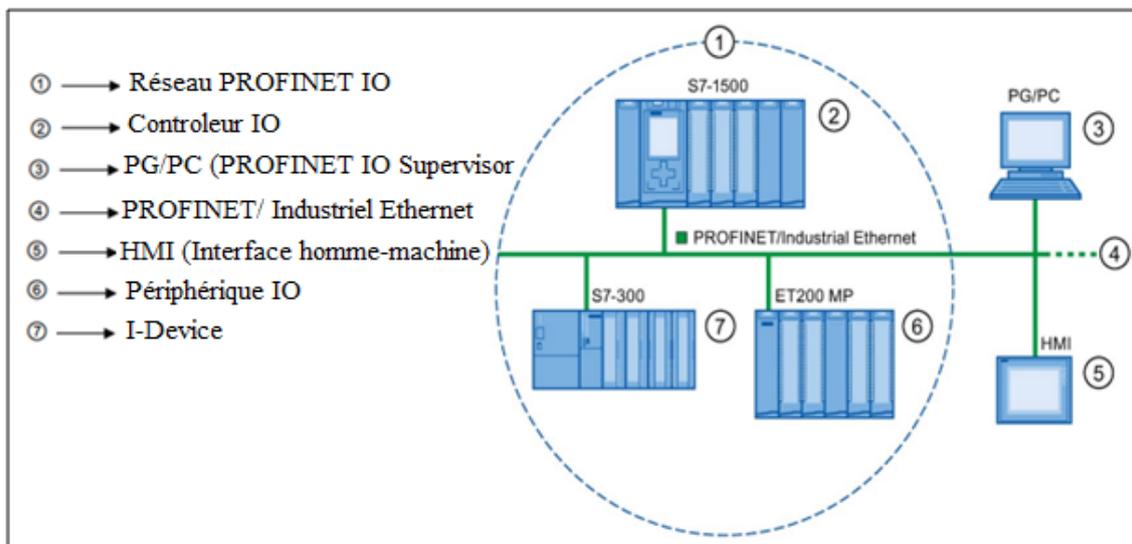


FIG. 1.33 : Les principaux composants dans PROFINET[39].

e. CANbus

Les réseaux CANbus sont des réseaux de bus de périphériques basés sur la technologie très répandue de la puce électronique CAN, qui est utilisée à l'intérieur des automobiles pour contrôler les composants internes, tels que les freins et autres systèmes. Il a été ensuite généralisé pour l'automatisme et les applications de contrôle. Un réseau CANbus est un système de protocole ouvert comportant des messages de longueur variable (jusqu'à 8 octets), un arbitrage non destructif et une gestion avancée des erreurs. Un câble à quatre fils plus deux fils de protection pour l'alimentation, deux pour la transmission du signal et un « cinquième » fil de protection assure la liaison de communication avec les dispositifs de terrain. Cette communication peut être maître/esclave ou pair à pair. La vitesse du réseau (vitesse de transmission des données) dépend de la longueur du câble du tronç[35]. La figure ci-dessous montre une configuration de base du système avec un PN/CAN LINK comme passerelle de communication entre un réseau PROFINET et un réseau CAN.

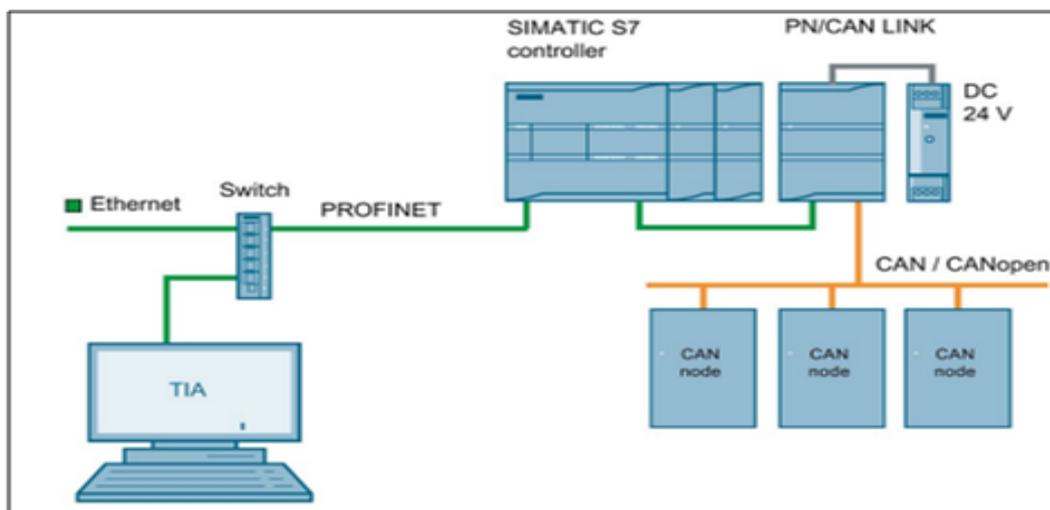


FIG. 1.34 : Configuration du système avec PN/CAN LINK[40].

f. Communication MPI

L'interface multipoint - Siemens (MPI) est une interface spécifique aux automates programmables industriels SIMATIC S7 de Siemens. Il est utilisé pour la connexion de stations de programmation (PG ou PC), de pupitres opérateur, et pour d'autres appareils de la gamme SIMATIC. Cette technologie a été à l'origine du développement du protocole Profibus. MPI est conforme à RS-485, la nature des communications multipoints[41].

1.9.4.2.5 Mémoire

La taille de la mémoire varie d'une marque d'API à l'autre, mais la mémoire peut souvent être élargie en installant une carte mémoire supplémentaire, par exemple une carte SD. Un API possède généralement les unités de mémoire suivantes[18] :

- ROM (Read-Only-Memory)

Utilisée pour une sauvegarde permanente du système d'exploitation et des données systèmes. Étant donné que les informations stockées dans une mémoire morte ne peuvent pas être supprimées, une mémoire morte programmable effaçable (EPROM) est utilisée à cette fin.

- RAM (Random-Access-Memory)

Pour le stockage de programmes. C'est parce qu'une RAM est très rapide. Étant donné que l'information dans une RAM ne peut pas être maintenue sans courant, les PLC ont une batterie afin que le code du programme ne soit pas perdu en cas de panne de courant. Certains PLCs ont également la capacité de stockage de programme dans une EPROM. RAM sont également utilisées lorsque le code de programme est en cours d'exécution. Certains API offrent la possibilité d'insérer de la mémoire supplémentaire.

1.9.4.2.6 Outil de programmation

Bien que la manière d'entrer le programme de contrôle dans l'automate ait changé depuis l'arrivée des premiers automates sur le marché, les fabricants d'automates ont toujours maintenu une interface humaine facile pour l'entrée du programme. Cela signifie que les utilisateurs ne doivent pas passer beaucoup de temps à apprendre comment entrer un programme, mais qu'ils peuvent plutôt passer leur temps à programmer et à résoudre le problème de contrôle. La plupart des automates sont programmés à l'aide d'instructions très similaires. La seule différence peut être la mécanique associée à la saisie du programme dans l'automate, qui peut varier d'un fabricant à l'autre. Cela implique à la fois le type d'instruction utilisé par chaque automate particulier et la méthodologie d'entrée de l'instruction à l'aide d'un dispositif de programmation[35]. Les deux types de base de dispositifs de programmation sont les suivants :

- Les mini-programmateurs

- Les ordinateurs personnels

Tous les fabricants d'API possèdent leur propre logiciel fourni avec l'API, par exemple : (Siemens travaille avec le logiciel SIMATIC STEP 7, Schneider travaille avec le logiciel ZELIO), ce qui permet d'éditer, de créer, de modifier et de télécharger le programme dans l'automate[24].

1.9.4.2.7 Programmation d'un API

Un programme est une suite d'instructions qui permettent de guider l'API dans l'exécution du processus. Siemens propose SIMATIC STEP 7 ou bien TIA Portal, qui est entièrement conforme à la norme internationale CEI 61131-3 relative aux langages de programmation des automates programmables, Schneider travaille avec le logiciel ZELIO, Allen-Bradley travaille avec le logiciel RSLogix 5000. Avec STEP 7, les programmeurs peuvent choisir entre différents langages de programmation. La norme, qui est actuellement suivie à un degré plus ou moins élevé par la plupart des grands fabricants d'automates, comprend différents langages de programmation[24] :

- Ladder ou langage contact (LD).
- Logigramme ou FBD (Function Block Diagram).
- Instruction List (IL).
- Structured Text (ST).
- Grafcet ou SFC (Sequential Function Charts).

LD, SFC et FBD sont des langages de programmation graphique, tandis que IL et ST sont des langages textuels.

a. Ladder ou langage contact (LD)

Le diagramme Ladder, ou tout simplement LD, est encore largement utilisé par de nombreux programmeurs d'automates. Et ce, malgré le fait que ST et SFC sont des langages plus efficaces dans la plupart des contextes. La raison pour laquelle le LD est toujours autant utilisé est qu'il est simple à comprendre et qu'il est basé sur les schémas de câblage électrique traditionnels (diagrammes de relais). Le LD consiste essentiellement en un ensemble d'instructions qui exécutent les types de fonctions de contrôle les plus élémentaires : logique, contrôle du temps et comptage, ainsi que des opérations mathématiques simples. La plupart des fabricants d'automates permettent aujourd'hui d'exécuter des fonctions supplémentaires avancées en LD, souvent intégrées à d'autres langages tels que FBD et ST. Pour les petites commandes, le LD peut donc être un excellent choix de langage de programmation. Les fonctions de base nécessaires à la mise en œuvre de petites applications peuvent être apprises relativement rapidement et la présentation graphique peut être comprise intuitivement[18].

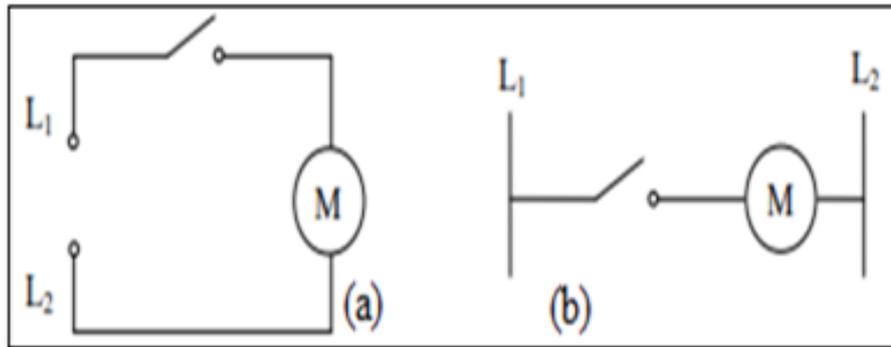


FIG. 1.35 : (a) Représentation d'un circuit électrique. (b) Représentation en Ladder[42].

Le diagramme Ladder est constitué de deux lignes verticales représentant les rails d'alimentation. Les circuits sont reliés par des lignes horizontales. Pour dessiner un diagramme en Ladder, certaines conventions sont adoptées [26] :

- Les lignes verticales du diagramme représentent les rails d'alimentation entre lesquels les circuits sont connectés.
- Un diagramme en Ladder est lu de gauche à droite et de haut en bas, le premier réseau est lu de gauche à droite. Ensuite, le deuxième réseau en bas est lu de gauche à droite et ainsi de suite. Lorsque l'API est en mode exécution, il passe par l'ensemble du programme Ladder jusqu'au dernier réseau, puis reprend rapidement au début. Cette procédure de passer par tous les échelons du programme est appelé un cycle.

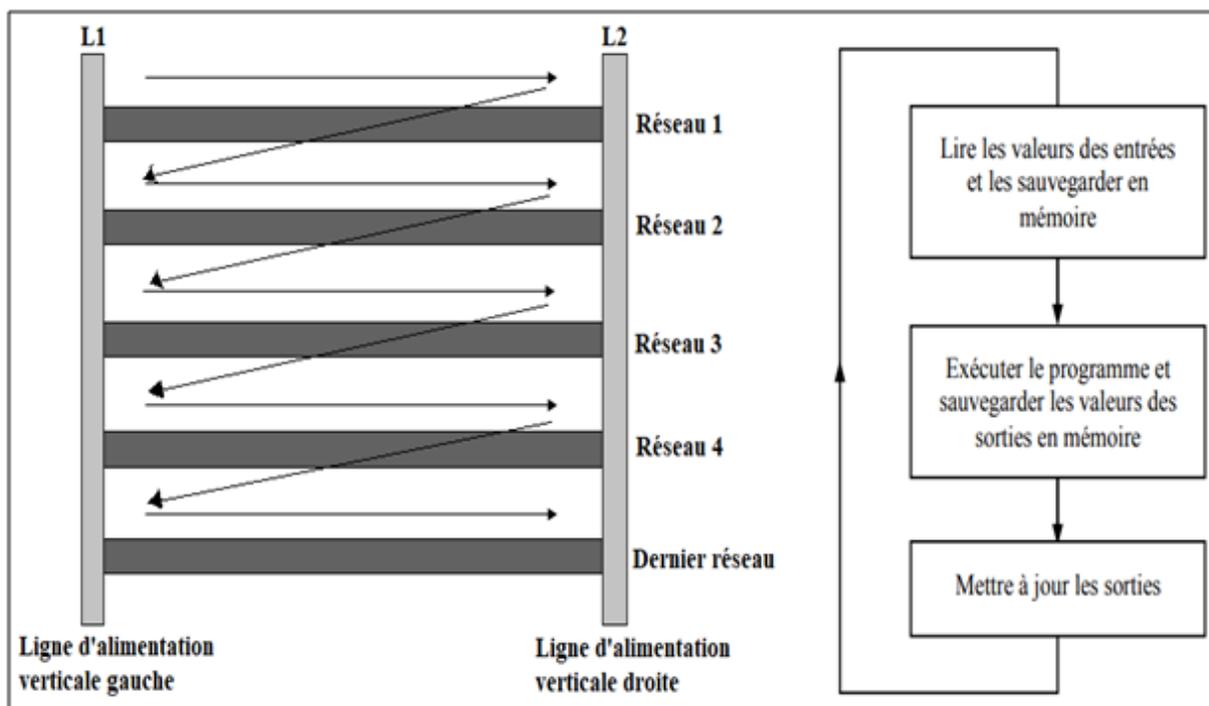


FIG. 1.36 : Cycle d'exécution dans un diagramme Ladder[42].

- Chaque réseau doit commencer par une ou plusieurs entrées et se terminer par au moins une sortie. Le terme entré est utilisé pour une action de commande. Le terme

sorti est utilisé pour un dispositif connecté à la sortie d'un API, par exemple, un moteur.

- Les dispositifs électriques sont représentés dans leur état normal, soit par un interrupteur normalement ouvert ou fermé.
- Les entrées et les sorties sont toutes identifiées par son nom et leur adresse, la notation utilisée dépendant du fabricant de l'automate.

b. Logigramme ou FBD (Function Block Diagram)

FBD est un langage graphique qui, décrit de manière très simplifiée, est basé sur la connexion de fonctions et de blocs fonctionnels. Le langage comprend, entre autres, l'utilisation de fonction logique standard telle qu'AND, OR, NOT, etc. Et de blocs fonctionnels (FB) tels que des minuteries et des compteurs, mais des fonctions et des FB auto-construits peuvent également être définis. De nombreuses personnes ayant peu de connaissances en électronique numérique pensent donc que c'est un excellent langage à utiliser ou, en tout cas pour commencer. Le bloc FBD est représenté de la manière illustrée à la Figure 1.39 avec le nom de la fonction écrit dans la case[18].

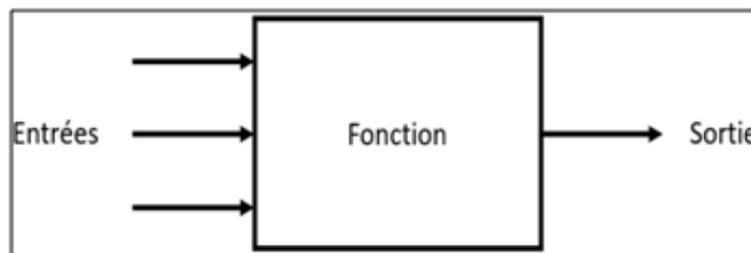


FIG. 1.37 : FBD ou logigramme[43].

c. Instruction List (IL)

IL est un langage de bas niveau de type assembleur. Même si l'utilisation d'un langage de bas niveau tel que l'IL présente des inconvénients, l'avantage de ce langage est qu'il ne nécessite pas une grande puissance informatique. Si ce langage continue d'être utilisé, c'est parce qu'il existe, avec le langage LD, depuis plus longtemps que les autres langages de la norme. De nombreux automates plus anciens ne peuvent être programmés qu'avec IL/LD. Il peut donc y avoir des cas où l'on peut utiliser le IL, par exemple, lorsque le code de programme écrit en IL est extrait d'un ancien automate pour être modifié ou analysé. Le langage IL a cependant des capacités et des applications limitées, car il est difficile à apprendre et peu compréhensible lorsque les tâches de programmation sont nombreuses ou complexes. Sur les anciens automates, le code en IL (ou LD) était programmé et transféré à l'unité centrale via un panneau spécial. Il n'était pas nécessaire (ou possible), comme c'est le cas aujourd'hui, de connecter un PC pour la programmation et le diagnostic[18].

d. Structured Text (ST)

Comme son nom l'indique, ST est un langage basé sur le texte. Contrairement aux IL (l'autre langage textuel de la norme), il s'agit d'un langage de haut niveau dans lequel de nombreuses opérations et instructions peuvent être effectuées avec une seule ligne de commande. Si nous devons le comparer à d'autres langages de haut niveau, ST ressemblerait le plus à Pascal ou à C. ST a été spécialement développé pour programmer des fonctions arithmétiques complexes, manipuler des tableaux, et travailler avec des objets mots et du texte[18].

e. Grafcet ou SFC (Sequential Function Charts)

Le langage SFC (Sequential Function Chart) est le dernier des langages définis dans la norme CEI 61131-3, défini comme étant capable de diviser des problèmes complexes en unités plus petites et plus faciles à manipuler et de décrire le flux de contrôle entre ces unités. L'ordre d'exécution de ces unités dépend des conditions statiques définies par le programme et des conditions dynamiques définies par les E/S. Les unités elles-mêmes peuvent être programmées à l'aide d'autres langages de programmation des API. Le premier langage largement adopté pour décrire un processus à travers un ensemble d'états et de conditions transitoires est le Grafcet, ce langage est proposé par de nombreux fabricants européens d'automates programmables, qui étaient alors principalement intégrés au langage SFC. Les processus qui peuvent être décrits par un comportement séquentiel caractérisé par des étapes sont particulièrement bien adaptés pour décrire leurs opérations dans ce langage[44].

Le GRAFCET (GRAphe Fonctionnel de Commande par Etape Transition) est un outil graphique qui permet de décrire le fonctionnement d'un automatisme dans le temps de manière claire et sans ambiguïté.

Une séquence dans Grafcet se compose de trois éléments principaux : Les étapes, les transitions et les actions[44].

1. Etape

Une étape correspond à une phase durant laquelle une action est réalisée pendant une certaine durée (même petite, mais jamais nulle). L'action doit être stable, ce qui veut dire qu'on fait la même chose pendant toute la durée de l'étape. Dans le Grafcet une étape est soit ACTIVE ou INACTIVE. Les étapes qui sont actives au début du processus sont appelées étapes initiales qui seront représentées par un carré double[44].

2. Action

Une ou plusieurs actions peuvent être associées à chaque étape distincte[2].

3. Transition

Contiennent des conditions qui doivent être satisfaites pour que le contrôle passe d'un état à un autre. Les conditions de la transition déterminent quand l'étape précédente

est désactivée et l'étape suivante activée. Pour qu'une transition puisse être testée par l'automate, toutes les étapes situées directement au-dessus de la transition doivent être actives. Lorsque la transition est satisfaite, l'étape précédente est d'abord désactivée, avant que l'étape suivante ne soit immédiatement activée. Deux transitions ne peuvent pas être placées côte à côte sans qu'une étape ne les sépare[2][34].

4. Réceptivité

Est une condition logique associée à une transition, peut être vraie ou fausse[2].

5. Les liaisons orientées

Indique le chemin d'évolution du Grafcet et relie les étapes aux transitions et les transitions aux étapes. Ils sont horizontaux ou verticaux. Le sens général d'évolution du Grafcet est du haut vers le bas. Les flèches doivent être utilisées dans d'autres situations ou lorsqu'elles peuvent conduire à une meilleure compréhension[34].

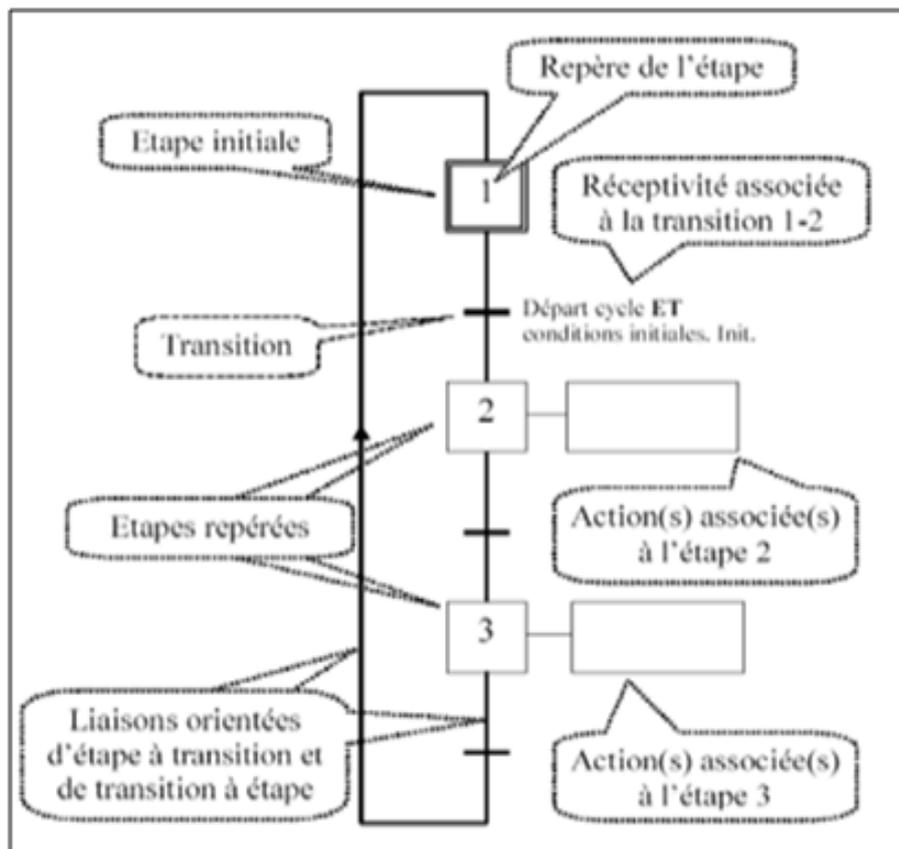


FIG. 1.38 : Les différentes composantes d'un Grafcet[44].

Un grafcet dispose d'un fonctionnement dynamique orienté par cinq règles, qui spécifient les causes et les effets des transitions de passage[44].

• Situation initiale

Un Grafcet doit contenir au moins une étape active au démarrage du système et à partir de laquelle le Grafcet va évoluer. Cette étape s'appelle l'étape initiale.

- **Franchissement d'une transition**

Une transition est dite franchissable lorsque toutes les étapes précédentes reliées à cette transition sont actives et la réceptivité associée à cette transition est correcte. Si la transition est acceptable, elle sera nécessairement passée.

- **Activation et désactivation d'une étape**

Le passage d'une transition provoque l'activation de toutes les étapes instantanément suivantes et la désactivation de toutes les étapes instantanément précédentes.

- **Évolution simultanée**

Plusieurs transitions en même temps franchissables à un instant t sont simultanément franchies.

- **Évolution des étapes actives**

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

1.10 Conclusion

Dans ce chapitre, nous avons parlé des systèmes automatisés de production de manière globale, où nous avons donné un très bref historique, quelques définitions, les domaines de leurs applications, ensuite nous avons expliqué de quoi est composé un système automatisé où nous avons parlé de la partie commande et de la partie opérative. Dans la deuxième partie du chapitre, nous avons parlé des capteurs, des pré-actionneurs, des actionneurs et des effecteurs qui appartiennent tous à la partie opérative. La troisième partie du chapitre a été consacré à la partie commande, où nous avons commencé par présenter les automates programmables, leur architecture et leurs composantes, ensuite ont parlé des principaux bus de communication et de terrain utilisés par les automates. Enfin, nous avons terminé le chapitre par parler des langages de programmation utilisés par les automates programmables industriels.

CHAPITRE

— 2 —

LA SUPERVISION ET LA SIMULATION
DANS L'INDUSTRIE

2.1 Introduction

La supervision offre une solution complète de la gestion des installations industrielles. Elle améliore le rendement, la productivité et la qualité des opérations, elle réduit les temps d'arrêt, elle permet la maintenance à distance et renforce la sécurité des personnes et des équipements.

Nous présenterons dans ce chapitre une vue d'ensemble de la supervision et de son utilisation au sein des installations industrielles, ainsi que certaines définitions utiles dans ce domaine, ensuite nous exposerons quelques notions de base de la supervision dans un environnement SCADA. À la fin, nous parlerons d'un composant de base d'un système SCADA, qui est l'IHM.

2.2 La supervision industrielle

La supervision industrielle représente pour un opérateur humain ou pour un système extérieur, une image des performances du système pour le contrôler à distance, elle permet de suivre en temps réel une installation ou une machine industrielle, elle fournit un affichage dynamique du processus avec les différentes alarmes de défaut et les événements qui se produisent pendant le fonctionnement de la machine, elle offre la possibilité de centraliser la vision et le pilotage des organes physiques (capteurs, actionneurs)[45]. L'affichage des informations relatives aux actionneurs et aux capteurs permet de découvrir l'état de fonctionnement du processus. Les défauts peuvent être représentés graphiquement et textuellement afin de mieux comprendre. Tous les paramètres analogiques (températures, pressions, etc.) peuvent être visualisés sous forme de courbes et de diagrammes. Enfin, toutes les données peuvent être enregistrées dans un système de gestion base de données à intervalles réguliers [46].

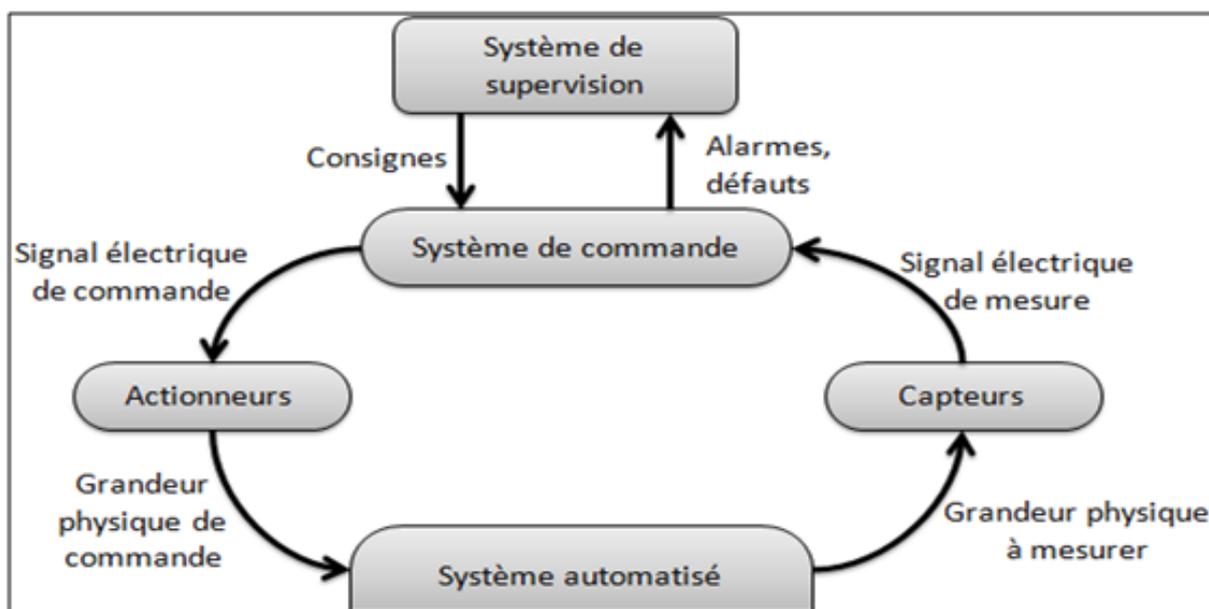


FIG. 2.1 : Place de la supervision dans un système automatisé[45].

La supervision dans la signification de "supervision industrielle" est l'interface de dialogue entre le système automatisé d'une part et l'opérateur qui surveille le système d'autre part. Cette interface, souvent appelée Interface Homme Machine (IHM), permet de visualiser les données contenues dans le système de contrôle et d'obtenir les requêtes des opérateurs[45].

L'affichage des IHMs de supervision sous forme graphique facilite la tâche de l'utilisateur pendant le processus de supervision. Plusieurs logiciels de conception d'applications communément appelés SCADA ont été conçus pour faciliter la conception des systèmes de supervision[47]. La supervision offre beaucoup d'avantages[48] :

- Contrôle de la disponibilité des services et des fonctions, avec la surveillance des processus à distance.
- Contrôler l'utilisation des ressources et vérifier leur adéquation.
- Traitement des données.
- Détection, localisation, diagnostic et prévention des défaillances.



FIG. 2.2 : Poste de supervision[49].

2.3 Structure de la supervision

La supervision est le moyen utilisé par l'opérateur comme outil de référence, mais peut également dialoguer directement avec le système de commande/contrôle. Il s'agit de surveiller le fonctionnement d'un procédé afin d'amener et de maintenir celui-ci à son

point de fonctionnement idéal en assurant des fonctions de collecte et de visualisation des informations, de surveillance, de diagnostic, de détection et d'aide à la décision pour l'adaptation, la reconfiguration ou la maintenance[46].

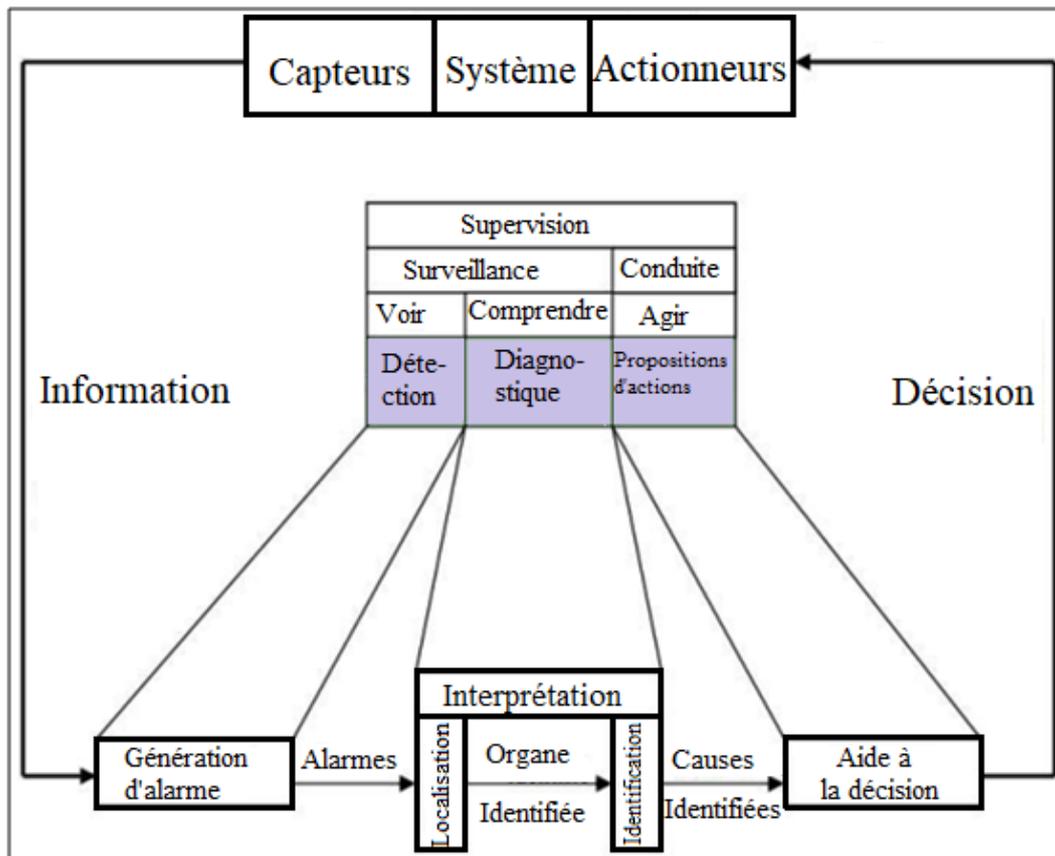


FIG. 2.3 : Structure globale d'un système de supervision[50].

2.4 Techniques de la supervision

La conception d'un système de supervision nécessite la maîtrise des techniques suivantes :

2.4.1 Acquisition de données

L'acquisition des données, c'est la phase initiale de la supervision, car elle consiste à collecter, vérifier et garantir le flux d'informations sur l'état du système vers le poste de pilotage, cette tâche est effectuée sans arrêt et à chaque moment et en temps réel. Elle comprend l'utilisation de capteurs pour mesurer les différentes variables du processus. Ces informations seront utilisées dans les rapports de résidus pour compléter l'étape de détection [50].

2.4.2 Surveillance

La surveillance est un appareil permettant l'analyse de l'état du système et la fourniture des indicateurs. La surveillance comprend la détection et la classification des dé-

faillances en observant l'évolution du système, avant leur diagnostic en localisant les défauts et en identifiant les causes profondes[51]. La surveillance se sert des informations du système pour montrer l'état de fonctionnement, puis par la suite détecter les changements. La surveillance intervient dans la phase d'exploitation, même si elle est prise en compte dès la phase de conception. Elle peut également détecter et diagnostiquer les pannes et les erreurs du système. Si une défaillance se produit, elle informe le module de service et le module de supervision[50]. Il existe deux types de surveillance : la surveillance du système opérant et la surveillance de la commande. La surveillance du système opérant comprend deux types de surveillance : la surveillance curative et la surveillance prédictive[52].

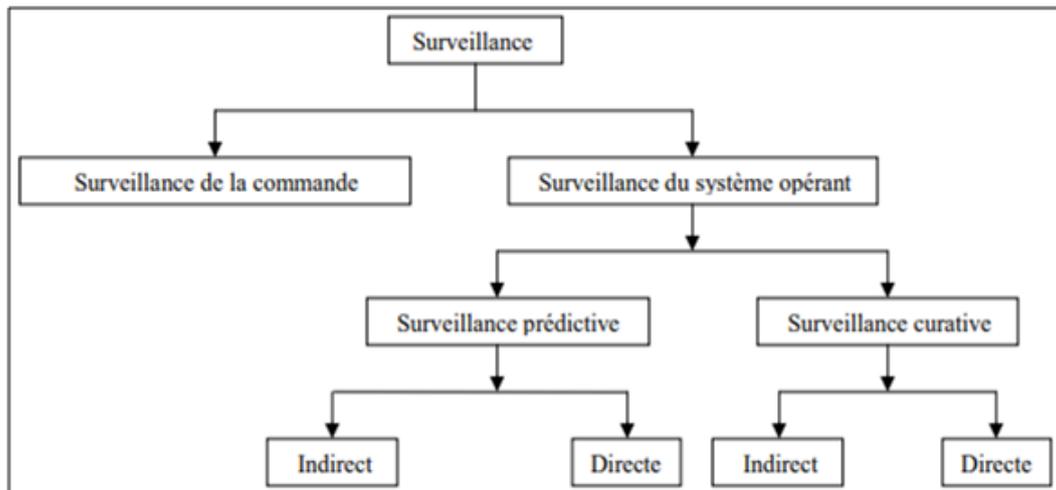


FIG. 2.4 : Différents types de surveillance[52].

2.4.3 Détection

La détection est essentielle, car il doit être possible de décider si le système est dans une situation normale de fonctionnement ou non en comparant les valeurs résiduelles aux seuils précédemment définis. Cette étape s'effectue à l'aide des tests statistiques ou à l'aide d'un seuillage[50]. Le but de cette étape est de détecter les pannes du système, la capacité de classer les situations qui peuvent être observées comme normales ou anormales, et de surveiller les conditions de fonctionnement liées au vieillissement des composants du processus et aux changements de l'environnement. Il s'agit d'un élément important dans la gestion des systèmes de production[51].

2.4.4 Diagnostic

Cette étape permet de localiser le composant défectueux et d'identifier la cause qui a provoqué la défaillance à l'aide d'un raisonnement logique basé sur un ensemble d'informations issues d'un contrôle ou d'un test. La fonction de diagnostic a pour but d'explorer les raisons et de localiser les éléments qui ont conduit à une observation particulière[50][51]. Cette procédure de diagnostic se compose de deux fonctions de base :

- **La localisation**

Cette étape se déroule dès qu'elle détecte un défaut, il est nécessaire de déterminer quel composant du système est défaillant. Il devrait isoler le sous-système qui fonctionne mal et rechercher quel composant a un défaut[53].

- **L'identification**

Déterminer la nature du défaut et les causes du problème, en établissant une relation de causalité entre les symptômes et les éléments défaillants[53].

2.4.5 Aide à la décision

Dans certaines circonstances, même l'opérateur le plus expérimenté et le plus qualifié commet des erreurs en prenant des décisions inadéquates, cela est dû à l'épuisement de l'opérateur, à un nombre important d'alarmes qui se produisent en même temps et que l'opérateur n'est pas en mesure de gérer ou simplement à une interprétation erronée de sa part. L'aide à la décision permet d'aider l'opérateur à prendre la bonne décision face à n'importe quelle situation, en suggérant une liste d'actions qui pourraient rétablir les conditions idéales du système[50].

2.4.6 La maintenance

La maintenance est l'étape qui se fait habituellement après celle de la prise de décision, elle consiste à maintenir ou à rétablir les performances des composants ou du système de manière globale, pour la réalisation de sa tâche requise, ces activités sont une combinaison d'activités techniques, administratives et de gestion. Quant au cours d'une intervention préventive un composant interne de l'équipement est repéré ou constaté défectueux, sa réparation ou son remplacement est considéré comme une maintenance corrective. S'il s'avère qu'il n'est pas en mauvais état, mais détérioré, même au-delà de la valeur de défaillance potentielle, sa réfection ou son changement est de type préventif[50].

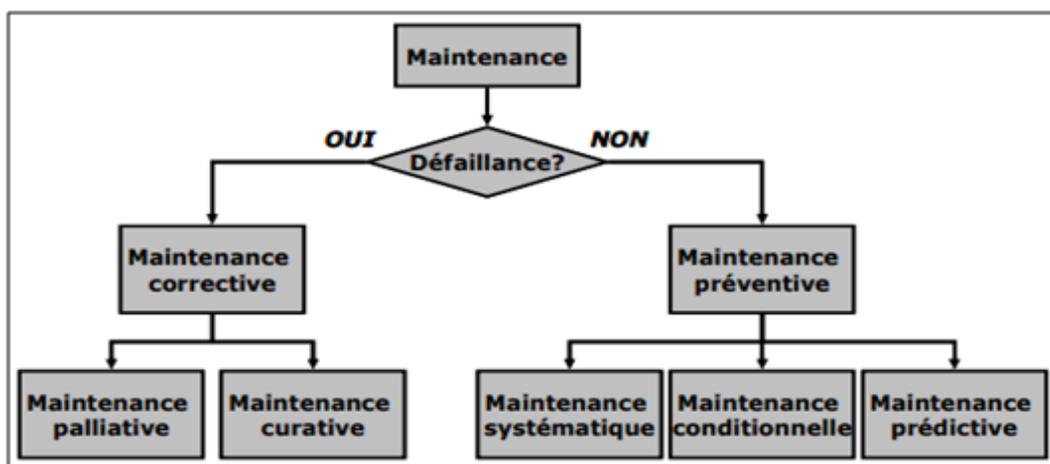


FIG. 2.5 : Les différentes formes de maintenance[54].

- **Maintenance préventive**

La maintenance préventive consiste en des interventions de maintenance visant à réduire la probabilité d'une défaillance ou la dégradation d'un composant de l'outil de production, ces interventions sont effectuées avant que ne survienne une défaillance qui générerait la production. Cette notion de maintenance est fondée sur une vérification régulière en fonction de critères prédéterminés de l'outil de production, de façon à évaluer le bon état de fonctionnement de celui-ci. Les techniques de maintenance préventive comprennent trois niveaux[54] :

- 1. Maintenance préventive systématique**

La maintenance préventive systématique a lieu avant même l'observation d'une simple détérioration, parce qu'elle se fait à la fin de la durée de vie des composants sans prendre en considération l'état du système à ce moment-là, le remplacement se fait de manière systématique[50][54].

- 2. Maintenance préventive conditionnelle**

La maintenance préventive conditionnelle a pour but de rétablir les fonctionnalités des composants estimés potentiellement défectueux, après étude et comparaison de leurs états à un critère défini au préalable[50][54].

- 3. Maintenance préventive (prévisionnelle)**

La maintenance préventive (prévisionnelle) a pour but de différer la défaillance et de programmer les interventions, elle est associée à la détérioration des composants et à son évolution au cours du temps[50][54].

- **Maintenance corrective**

La maintenance corrective est généralement considérée comme la forme de base de la maintenance puisque l'intervention a lieu "en urgence" une fois que la défaillance s'est produite, pour restaurer le fonctionnement correct du système, de manière permanente ou temporaire. Il existe deux types de maintenance corrective[54] :

- 1. Maintenance palliative**

La maintenance palliative sert à restaurer les fonctionnalités exigées du système de manière temporaire, généralement ce type de maintenance est toujours suivi d'une maintenance curative, dont le rôle est de gagner du temps pour la planification et la réalisation des tâches de maintenance curative[50][54].

- 2. Maintenance curative**

La maintenance curative consiste à remettre le système en bon état de fonctionnement de façon permanente par le biais de certaines actions, ces interventions peuvent être des réparations, des substitutions de composants ou des arrangements qui élimineront les défaillances[50][54].

2.5 Supervision dans un environnement SCADA

Les systèmes SCADA sont largement utilisés dans l'industrie pour le contrôle, la surveillance et l'acquisition des données des processus industriels. Les systèmes SCADA ont fait des progrès considérables ces dernières années en termes de fonctionnalité, d'évolutivité, de performance et d'ouverture, de sorte qu'ils constituent une alternative au développement interne, même pour des systèmes de contrôle très exigeants et très complexes [55]. Le système SCADA travaille en acquérant des données à partir de l'installation, ces données sont présentées sur une interface graphique en utilisant un langage assez proche du langage humain, ces actions sont effectuées en temps réel, de sorte que les systèmes SCADA fournissent à l'opérateur un maximum d'informations pour avoir une vision exacte et globale du système complet. Ils offrent un degré de sécurité très élevé, tant pour le personnel que pour l'installation, ainsi que la réduction des coûts d'exploitation, les avantages offerts par SCADA sont obtenus grâce à une combinaison d'outils souples et résistants[50].

2.6 SCADA

SCADA « Supervisory Control And Data Acquisition » est un acronyme qui signifie le contrôle et la supervision par acquisition des données, permettent le contrôle, la gestion et la surveillance des systèmes de contrôle de processus et d'automatisation de la fabrication par la collecte et l'analyse de données en temps réel. Les capacités des systèmes SCADA ont évolué, passant du simple remplacement des lumières et des boutons poussoirs à la gestion de systèmes très complexes de contrôle de processus et d'arrêt de sécurité critique. L'intelligence des systèmes SCADA a progressé au point que leur fonctionnement automatisé nécessite moins d'opérateurs, c'est-à-dire moins de supervision humaine que les méthodes de contrôle précédentes. Les applications des systèmes SCADA sont diverses. Ils sont utilisés dans la plupart des processus industriels comme la production et la distribution d'électricité dans le contexte des réseaux intelligents, le domaine des industries du gaz et du pétrole, dans les industries agro-alimentaires, dans les industries métallurgiques, dans les stations de traitement et d'épuration des eaux entre autres[56].

2.6.1 Évolution du SCADA

La portée du SCADA a beaucoup évoluée depuis ses débuts dans les années 1960, avec le développement des microprocesseurs et l'arrivée des ordinateurs le contrôle informatique des opérations de fabrication est devenu possible. Au fur et à mesure que les capacités techniques des ordinateurs, des systèmes d'exploitation et des réseaux se sont améliorés, la gestion organisationnelle a poussé à une meilleure connaissance de l'état en temps réel des opérations de l'usine à distance[57]. Au cours des années 1980 et 1990, SCADA évolué encore une fois avec une large utilisation de réseaux locaux (LAN), qui a permis aux systèmes SCADA d'être connecté à d'autres systèmes et introduire des logiciels IHM sur PC.

2.6.2 Architecture du SCADA

Les systèmes SCADA sont des systèmes de contrôle-commande axés sur la supervision. Un système SCADA est caractérisé par une architecture matérielle et logicielle.

L'architecture matérielle du SCADA se compose communément de stations clientes, de serveurs de données, des Automates Programmables Industriels (API), les systèmes physiques contrôlés. Les stations clientes traitent essentiellement l'interaction homme-machine. Le rôle du serveur de données est de gérer et d'obtenir des informations fournies par les automates programmables, qui sont basés sur les données des capteurs, enfin les automates programmables vont opérer sur le processus physique par le biais des actionneurs.

L'architecture logicielle d'un système SCADA comporte le logiciel embarqué dans les divers composants de l'architecture physique. Le client SCADA se compose notamment d'une interface homme-machine (IHM), communément appelée "IHM de supervision", laquelle permet d'afficher toutes les données relatives à l'état du système contrôlé, et autoriser aux utilisateurs de surveiller l'état du système et de le contrôler à distance. Le client comprend aussi un gestionnaire d'alarmes et de scripts[58]. Le serveur SCADA est le responsable de la coordination entre l'IHM et les programmes de contrôle. Il se compose d'un gestionnaire d'événements et comporte divers protocoles de communication. Le gestionnaire d'événements assure le transfert des informations entre le client SCADA et les APIs[58].

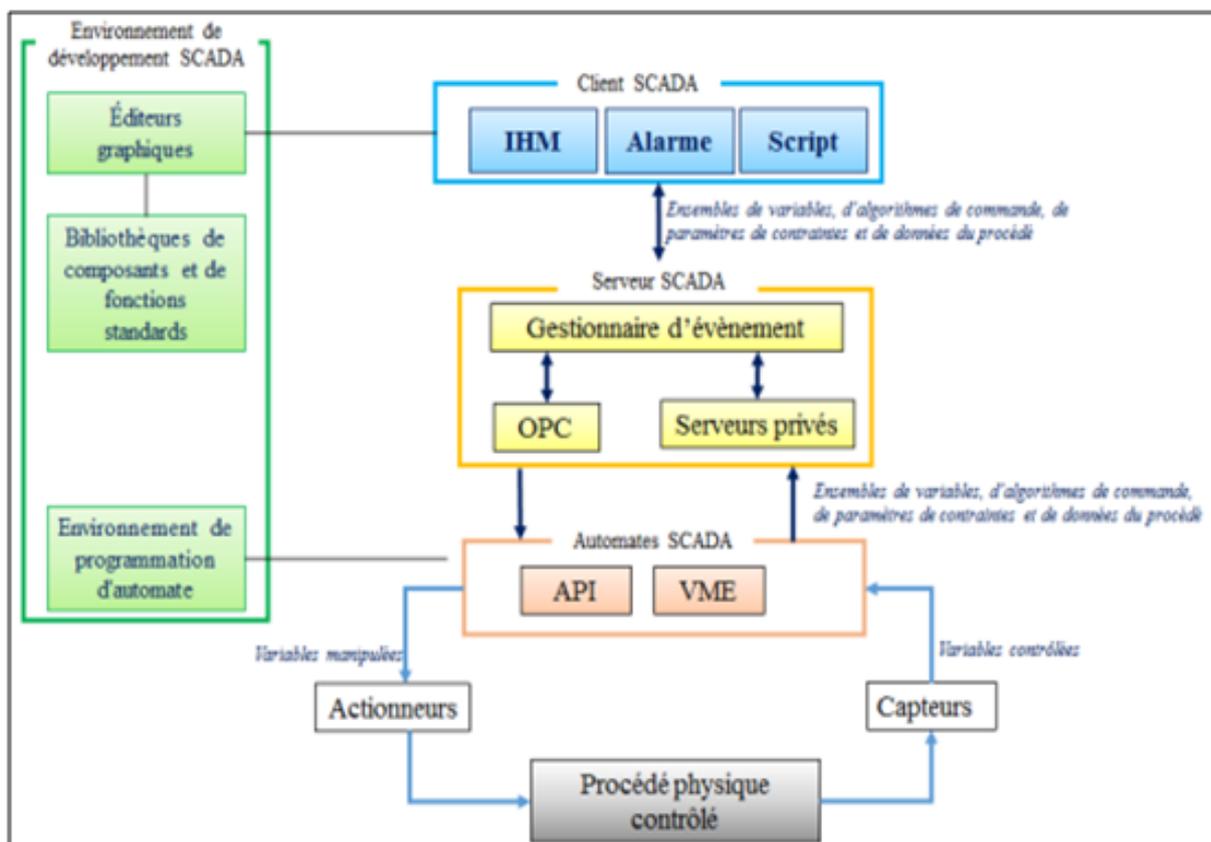


FIG. 2.6 : L'architecture SCADA[58].

2.6.3 Composants de base d'un système SCADA

La plupart des systèmes SCADA sont constitués de plusieurs composants qui communiquent sur un réseau. Un système SCADA de base se compose des éléments suivants[59] :

- Interface Homme-Machine (IHM).
- Système de surveillance (Supervisory System).
- Unités de terminal à distance RTU (Remote Terminal Unit).
- Automates Programmables Industriels (API ou bien PLC).
- Infrastructure de communication.
- La programmation des systèmes SCADA.

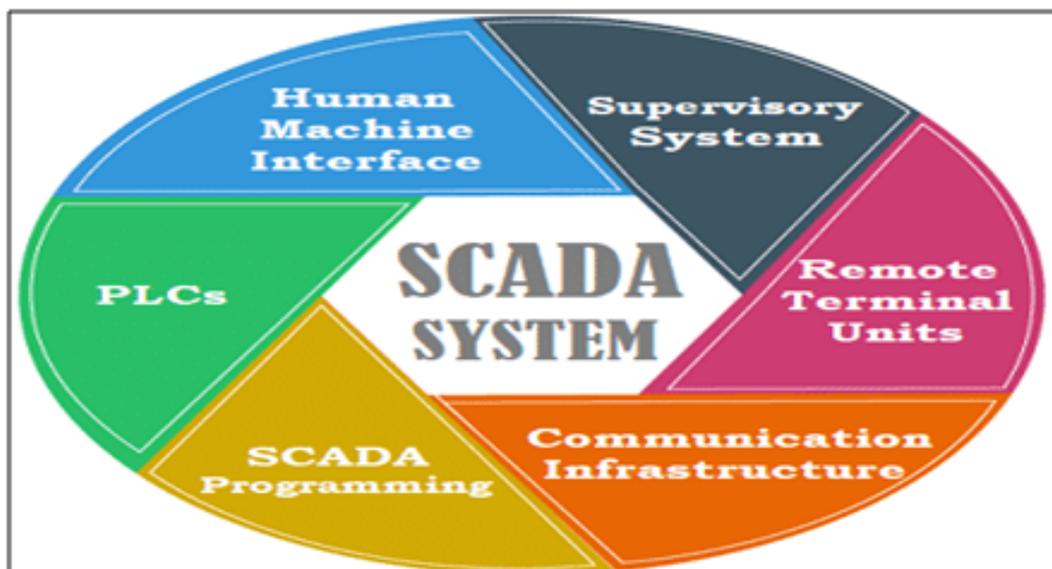


FIG. 2.7 : Composants du système SCADA[59].

Dans notre travail nous allons nous intéresser aux interfaces homme-machine dédiées à la supervision des systèmes industriels.

2.7 Interface Homme-Machine (IHM)

Avant l'apparition de l'interface homme machine, l'utilisateur ou l'opérateur ne disposait d'aucune sorte d'affichage visuel. Il devait se contenter de regarder en permanence la machine ou le processus, ou les informations sonores et les alarmes, et utiliser les boutons poussoirs du panneau de commande pour contrôler le processus[60].



FIG. 2.8 : Les organes de visualisation avant l'apparition de l'IHM[61].

Une interface homme-machine (IHM) est l'appareil qui présente les données du processus à un opérateur humain et par lequel ce dernier contrôle le processus. Il est le moyen par lequel l'utilisateur (opérateur) interagit avec le système SCADA. En termes simples, l'IHM fournit une représentation informatique claire et facile à comprendre de ce qui est, en fait, contrôlé ou surveillé par le système SCADA. Une IHM peut fonctionner sur diverses plateformes, notamment des ordinateurs de bureau, laptops, des tablettes, smartphones ou des écrans tactiles dédiés aux panneaux[56].



FIG. 2.9 : Exemple d'une IHM[62].

Les IHMs de la génération actuelle ne remplacent pas seulement les boutons poussoirs et les voyants lumineux du passé. En fait, elles offrent une interface utilisateur très simple et très facile à utiliser même pour des systèmes SCADA complexes[56].

2.7.1 Fonctionnalité de l'interface homme-machine

Les interfaces homme-machine sont utilisées comme moyen pour qu'un opérateur interagisse avec PLC et RTU. Ils remplacent les interrupteurs et autres dispositifs de commandes électriques activés manuellement, par des représentations graphiques de commandes numériques. Les IHMs permettent aux opérateurs de démarrer et d'arrêter des cycles, d'ajuster les points de consignes et d'exécuter d'autres fonctions requises pour ajuster et interagir avec un processus de contrôle. Les IHMs étant basées sur un logiciel, elles remplacent les fils et les commandes physiques par des paramètres logiciels, ce qui permet de les adapter et de les ajuster très facilement[63]. Les fonctions de base des IHM sont les suivantes[64] :

- Affichage des données
- Surveillance des temps de production.
- Surveillance des indicateurs clés de performances.
- Surveillance des entrées et des sorties de la machine.

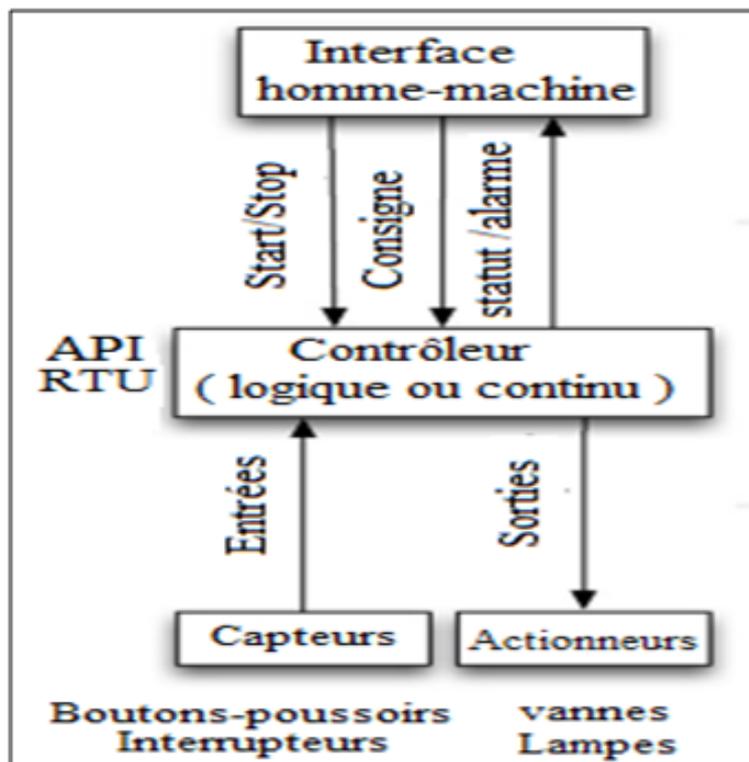


FIG. 2.10 : Fonctionnalité de l'interface homme-machine[63].

2.7.2 Les avantages d'une IHM

Les IHMs fournissent un certain nombre d'avantages aux organisations industrielles actuelles, notamment[65] :

- **Amélioration de la visibilité** Elles fournissent une meilleure visibilité des opérations tout le temps, avec la possibilité de visualiser l'installation sur un seul tableau de bord.
- **Augmentation de l'efficacité** La visualisation et l'analyse des données permettent d'améliorer l'efficacité des procédures.
- **Diminution des temps d'arrêt** Grâce aux alertes sur le tableau de bord central, la réponse aux problèmes devient plus rapide, cela réduit les temps d'arrêt.

2.7.3 La différence entre IHM et SCADA

IHM et SCADA sont parfois confondus en raison de leurs similitudes et du fait qu'ils travaillent ensemble. En effet, une IHM fait fréquemment partie d'un système SCADA. Ils font tous deux, partie d'un système de contrôle industriel plus vaste, mais chacun offre des fonctionnalités et des possibilités différentes. Tandis que les IHM sont axées sur la transmission visuelle d'informations pour aider l'opérateur à superviser un processus industriel, les systèmes SCADA ont une plus grande capacité de collecte de données et de fonctionnement du système de contrôle. Au contraire aux systèmes SCADA, les IHM ne recueillent pas et n'enregistrent pas d'informations et ne se connectent pas à des bases de données. Les IHM constituent plutôt un outil de communication efficace qui fonctionne en tant que partie intégrante ou parallèle d'un système SCADA[62].

2.7.4 Conception d'IHM pour la supervision

Les interfaces de supervision industrielle sont des systèmes interactifs, c'est-à-dire que l'opérateur peut réagir avec le système par le biais d'une IHM. La conception des systèmes interactifs est basée sur la disjonction entre la partie graphique et l'application. L'architecture d'un système interactif est communément constituée de ces modules : dialogue, présentation et application [58].

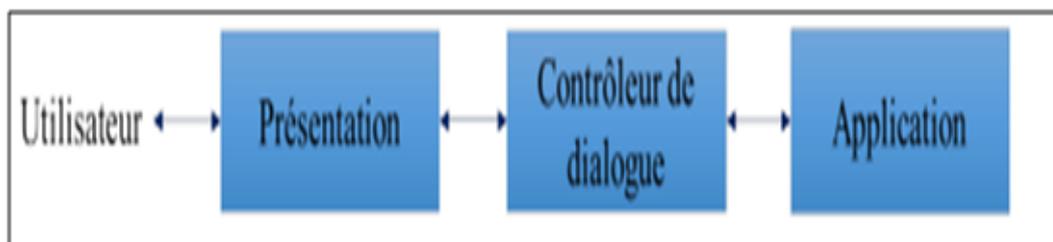


FIG. 2.11 : Structure simplifiée d'un système interactif[58].

La présentation est la partie graphique présentée à l'utilisateur. Elle se compose de nombreux outils par exemple les boutons, qui autoriser à l'utilisateur de réagir avec l'interface. Le contrôleur de dialogue est l'élément principal dans la conception des systèmes interactifs. Il agit comme un coordinateur entre la présentation et l'application. En réalité, il permet d'invoquer les fonctions nécessaires à l'accomplissement d'une tâche demandée

par l'utilisateur. Il affiche également l'état du système à l'utilisateur par le biais d'objets d'affichage. L'application rassemble l'ensemble des données et des fonctions qui permettent au système d'évoluer et d'effectuer les tâches demandées par l'utilisateur. Dans les systèmes SCADA, l'application se compose du processus physique, de serveurs de données et de programmes de contrôle[58].

2.7.5 Types d'interfaces homme-machine (IHM)

Les interfaces homme-machine (IHM) peuvent prendre diverses formes, qu'il s'agisse d'affichages dans la machine, d'écrans d'ordinateur, d'écrans tactiles ou de dispositifs mobiles, mais quel que soit leur format ou leur terme, son objectif est d'approvisionner des informations sur les performances et l'évolution des procédés de production des systèmes et des stations industriels[64].

Durant la dernière décennie, l'évolution des besoins opérationnels et commerciaux a suscité des développements intéressants dans le domaine des IHMs. Aujourd'hui, il est de plus en plus courant de voir des formes évoluées d'IHM, telles que des IHMs haute performance, des écrans tactiles et des dispositifs mobiles, à côté de modèles plus traditionnels. Ces interfaces modernisées créent davantage de possibilités d'interaction et d'analyse des équipements[62].

- **IHM de haute performance**

Les utilisateurs et les opérateurs se tournent progressivement vers les IHMs hautes performances, c'est une méthode de concevoir des IHMs qui permettent l'interaction rapide et efficace. En dirigeant l'attention uniquement sur les indicateurs les plus nécessaires ou les plus critiques de l'interface, cette technique de conception aide l'utilisateur à voir et à répondre plus efficacement aux problèmes, et de prendre des décisions mieux éclairées. Les écrans IHM haute performance sont simples, propres et dépourvus de tout graphique ou commandes inutiles. Les autres éléments de conception, tels que la couleur, la taille et l'emplacement, sont discrètement utilisés dans le but d'optimiser l'expérience de l'utilisateur[62].

- **Écrans tactiles**

Les écrans tactiles et les IHM mobiles sont deux modèles de progrès technologiques ç'a grandi avec l'arrivée des smartphones. À la place des boutons et d'interrupteurs, les IHM modernisées permettent aux opérateurs d'effleurer ou de toucher l'écran physique pour accéder aux commandes. Les écrans tactiles sont notamment importants lorsqu'ils sont utilisés avec une IHM mobile, qui est déployée soit par le biais d'une IHM/SCADA basée sur le Web, soit par une application. L'IHM mobile offre plusieurs avantages aux opérateurs, particulièrement un accès immédiat aux informations de l'IHM [62].



FIG. 2.12 : Visualisation avec IHM de type SIMATIC[66].

- **Surveillance à distance**

La surveillance à distance mobile offre plus de flexibilité et une meilleure accessibilité aux exploitants et aux responsables. Avec cette fonctionnalité, un ingénieur du système de contrôle situé hors site peut, vérifier la température d'un entrepôt sur un dispositif portable, ce qui élimine la nécessité d'une supervision sur place en dehors des heures de travail[62].



FIG. 2.13 : Surveillance à distance à l'aide d'un appareil mobile[62].

2.7.6 Connectivité des IHM avec les API

Les usines reçoivent leurs machines de différentes parties du monde avec des automates différents. Une bonne Interface homme machine doit avoir la capacité ou le support du protocole de communication pour parler à différents types d'automates (API)[60]. L'ingénieur système qui conçoit un pupitre opérateur doit programmer chaque curseur et bouton à une adresse d'entrée ou de sortie spécifique d'un API. Donc ça devrait être cohérent et comporter le même langage ou protocole. Les protocoles les plus courants dans l'industrie sont donc Modbus, Ethernet et Profibus. Ils permettent de connecter les automates programmables, les IHM/SCADA, les machines et autres dispositifs dans le processus de production[64].



FIG. 2.14 : Connectivité d'IHM avec l'API[60].

2.7.7 Applications des IHM

Les IHM sont utilisées dans diverses industries, notamment les usines de fabrication, les distributeurs automatiques, l'industrie alimentaire et celle des boissons, l'industrie pharmaceutique et les services publics, parmi de nombreuses autres applications. Les IHM, ainsi que les automates programmables, sont l'épine dorsale de la chaîne de production dans ces industries. L'intégration d'IHM dans tout système de fabrication peut améliorer considérablement les opérations. L'IHM permet de superviser le contrôle et l'acquisition de données dans l'ensemble du système, de sorte que l'opérateur peut modifier les paramètres à sa guise. Par exemple, dans les processus d'embouteillage, les IHM sont utilisées pour contrôler tous les aspects de la chaîne de fabrication, tels que la vitesse, l'efficacité, la détection et la correction des erreurs[67].



FIG. 2.15 : Exemple d'un écran tactile pour contrôler le processus d'embouteillage[68].

2.8 Conclusion

Dans ce chapitre, nous avons présenté une vue globale et générale de la supervision industrielle comme sa structure et les différentes techniques utilisées pour sa conception, nous avons parlé aussi de son utilité au sein des installations industrielles. Par la suite nous avons exposé quelques notions de base sur la supervision dans un environnement SCADA où nous avons abordé son évolution, son architecture et ses différentes composantes élémentaires comme les IHMs, où nous avons donné une idée sur leurs principales fonctionnalités, avantages, types, manière de connectivité et applications dans le domaine industriel.

CHAPITRE

3

LA PROGRAMMATION, SIMULATION
ET L'INJECTION AU NIVEAU DES
SOUS-STATIONS (MPS-VARIANT)

3.1 Introduction

Dans la première partie de ce chapitre, nous allons d'abord présenter une description des quatre sous-stations qui font part du nouveau système de production automatisé MPS variant, que nous proposons et que nous allons étudier. Nous allons présenter les principes de fonctionnement avec le cahier des charges que nous allons proposer pour les quatre sous-stations qui vont former ce nouveau système MPS. La deuxième phase du chapitre va être consacrée à la réalisation des cahiers des charges du système MPS variant proposé qui se compose des sous-stations : Distribution, Production, Manipulation et Trie, ainsi qu'à la configuration et la programmation du système en utilisant STEP7, puis la simulation avec S7-PLCSIM ensuite EasyPort USB.

3.2 Description des sous stations du Système MPS variant

3.2.1 Sous-station de distribution

La station de distribution FESTO dispose d'un magasin dans lequel des pièces de différentes couleurs sont transportées vers d'autres stations. Elle sépare les pièces à usiner du magasin à empilage en poussant celle du bas à l'aide d'un vérin Mono stable. Le tube de magasin d'empilage peut stocker jusqu'à neuf pièces de couleurs différentes (métallique, rouge et noir), ce magasin est doté d'un capteur optique à barrage B4, ce capteur a pour rôle de détecter la présence des pièces au niveau du magasin d'empilage[69].

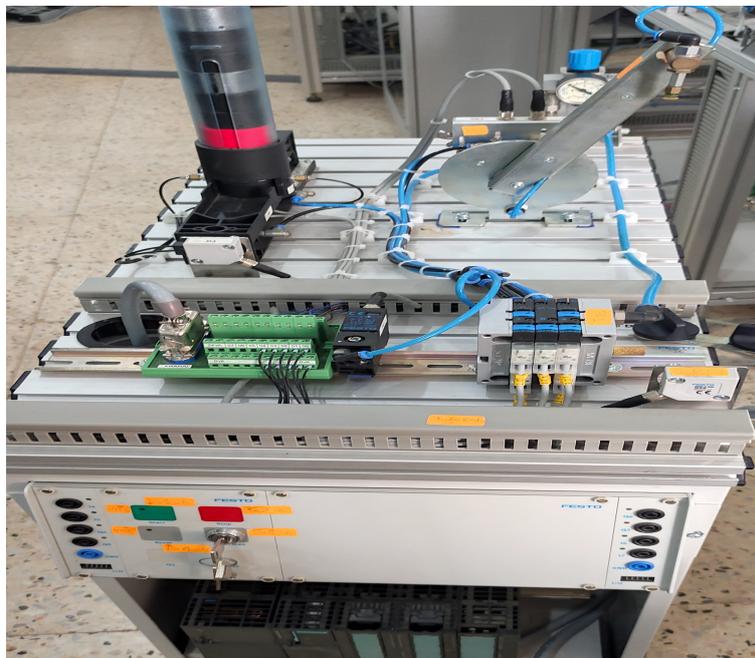


FIG. 3.1 : Sous-station de distribution du système FESTO MPS.

Les pièces doivent être insérées du côté supérieur ouvert du magasin. Un vérin d'éjection 1M1 éjecte les pièces à usiner une par une, la position du vérin d'éjection est détectée

Chapitre 3. La programmation, simulation et l'injection au niveau des sous-stations (MPS-variant)

au moyen de deux capteurs de fin de course électromagnétique 1B1 (Vérin d'éjection rentré) et 1B2 (Vérin d'éjection sorti). Le transfert des pièces à usiner vers la station suivante se fait au moyen d'un module de transfert avec un bras rotatif avec une ventouse pneumatique. Les pièces à usiner sont attrapées au moyen d'une ventouse qui permet d'aspirer les pièces à usiner via 2M1 et de les refouler via 2M2. Les deux opérations sont indiquées par un capteur 2B1 qui assure l'aspiration de la pièce. La figure 3.1 illustre la sous-station de distribution. Les pièces à usiner sont transportées par un vérin oscillant (le 3M2 responsable de la translation vers la station suivante et le 3M1 responsable de la translation vers le magasin), la limite de translation du vérin oscillant est indiquée par deux capteurs de fin de course électromécanique 3B1 coté magasin et 3B2 coté de station suivant.

3.2.1.1 Principaux composants de la sous-station de distribution

Il existe deux modules dans la sous-station de distribution :

- **Magasin à empilage**

Le magasin à empilage peut contenir jusqu'à 8 pièces. Les pièces doivent être insérées du côté ouvert en haut. Un vérin d'éjection pousse la pièce la plus basse du magasin. Cette position sert de point de transfert vers le module de changement. La disponibilité de la pièce dans le magasin est contrôlée au moyen d'un capteur à barrage. La position du vérin d'éjection est détectée électriquement par des capteurs électromagnétiques[69].

- **Module de transfert**

Le module de transfert est un dispositif de transport pneumatique. Il saisit la pièce qui a été poussée hors du magasin à l'aide d'une ventouse à aspiration. Un détecteur de vide vérifie si une pièce a été prise. Le bras de l'unité de transfert, qui est actionné par un vérin rotatif, transporte la pièce à usiner vers la station suivante et peuvent être transférées suivant un angle compris entre 0° et 180° . La détection de la position finale s'effectue au moyen des capteurs[69].

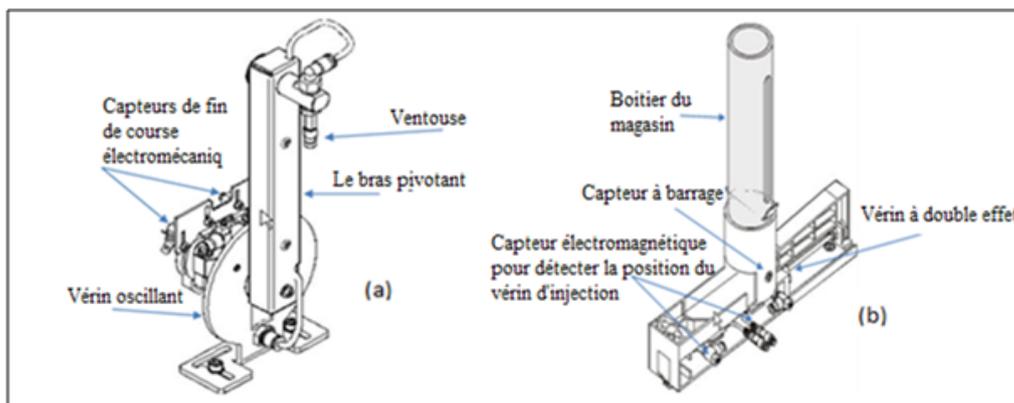


FIG. 3.2 : Composants principaux de la sous-station de distribution[69]. - (a) Module de transfert. - (b) Magasin à empilage.

3.2.1.2 Cahier de charge initial de la sous-station de distribution

Au départ, nous avons le magasin de distribution rempli de pièces à usiner, vérin d'éjection sorti, le vérin oscillant se trouve en position "Magasin", et le vide désactivé. Lorsque des pièces sont identifiées dans le magasin et qu'on appuie sur la touche START, le vérin oscillant tourne jusqu'à la position "Station suivante" sans aucun transfert de pièces, c'est-à-dire qu'il effectue un cycle à vide, le vérin d'éjection fait sortir une pièce à usiner depuis le magasin. Le vérin oscillant tourne à la position "Magasin". Le vide est activé, quand la pièce est bien aspirée. Le vérin d'éjection libère la pièce à usiner et revient à sa position initiale. Le vérin oscillant tourne jusqu'à la position "Station suivante", le vide est désactivé, le vérin oscillant tourne jusqu'à la position "Magasin" pour ramener une autre pièce.

3.2.1.3 Modifications apportées sur le cahier de charge de la sous-station de distribution

Le fonctionnement actuel de la sous-station de distribution n'est pas optimal, à cause du cycle à vide créé par le vérin oscillant (il y a donc un gaspillage d'énergie). Il y a également une perte d'énergie dans le vérin rotatif qui empêche la sortie des pièces lorsqu'il est en position magasin ainsi lorsqu'il fait un cycle à vide durant le lancement de mise en marche de cette dernière. Nous devons dans la modification que nous proposons dans le cahier de charge améliorer le fonctionnement de cette sous-station via la proposition d'une solution plus optimale.

L'amélioration que nous proposons dans le nouveau cahier de charge se résume en ce qui suit : A la situation initiale, le vérin d'éjection en position sorti, le vérin oscillant se trouve en position « Station suivante "contrôle" », et le vide est désactivé. Lorsque le bouton poussoir START est enfoncé, le vérin d'éjection pousse une pièce à usiner depuis le magasin, puis le vérin oscillant est déplacé en position "magasin". Si la pièce est correctement aspirée, le vérin oscillant tourne jusqu'à la position "Station suivante", quand il arrivera à la station suivante, il jette la pièce. Si nous étions en mode automatique, le cycle se répèterait de manière cyclique, en mode manuel, lorsque le cycle se termine, il faut appuyer sur le bouton START. Le voyant "magasin vide" s'allume si le magasin est vide.

3.2.2 Sous-station d'usinage

La sous-station d'usinage est utilisée pour programmer deux opérations exécutées sur un plateau à indexation rotatif (le contrôle et le perçage). Elle utilise uniquement des actionneurs électriques. La pièce à usiner dans la station d'usinage est détectée à l'aide d'un capteur (Part-AV). Le plateau à indexation rotative possède 6 réceptacles semi-ouverts et elle est commandé par un moto-réducteur à courant continu M2. Son positionnement se fait au moyen d'un circuit à relais. La position du plateau d'indexation est détectée par un capteur inductif B3. Sur le plateau à indexation rotatif, les pièces à usiner sont testées (M5 : pour contrôler la pièce à usiner) et percées (M1: moteur de l'unité de perçage en marche, K3: descente de l'unité de perçage, K4: monter de l'unité

de perçage) les deux processus s'exécute en parallèle. Une bobine équipée d'un induit contrôle si les pièces à usiner sont placées dans la bonne position (pendant le processus de contrôle). Ainsi que pendant le perçage, la pièce à usiner est serrée par un vérin de serrage (M4 : pièce fixée)[69]. La figure 3.3 illustre la sous-station d'usinage.

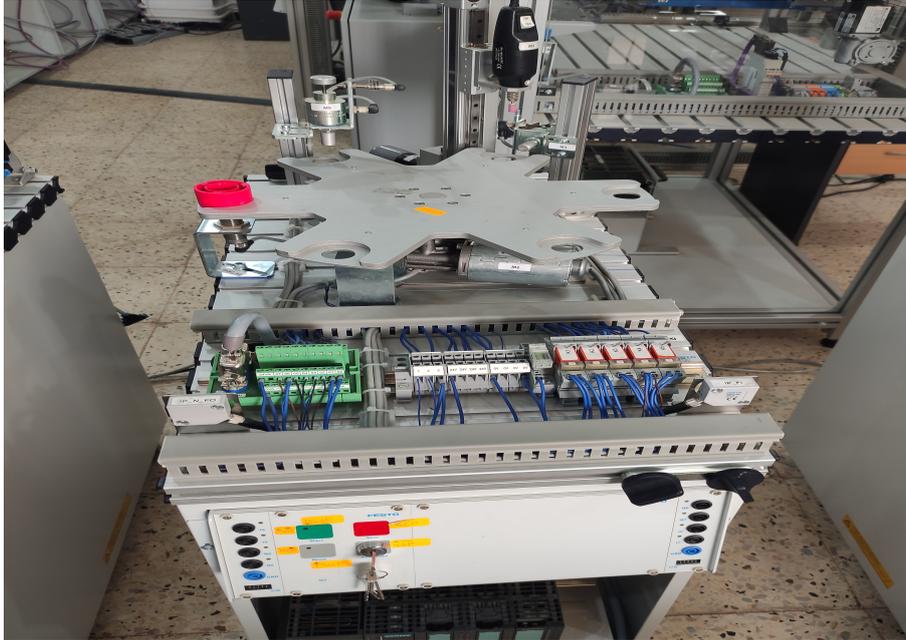


FIG. 3.3 : Sous-station d'usinage du système FESTO MPS.

3.2.2.1 Principaux composants de la sous-station d'usinage

La station d'usinage peut être subdivisée en trois modules partiels :

- **Module table à indexation rotative**

Le module de la table à indexation rotative est actionné par un moto-réducteur à courant continu. Les six positions du plateau rotatif sont déterminées par les vis de positionnement sur la table rotative et sont détectées au moyen d'un capteur inductif. Chacun des 6 réceptacles semi-ouverts du plateau est pourvu d'un trou en son centre pour faciliter la détection de pièce à usiner au moyen d'un capteur de proximité capacitif[69].

- **Module de contrôle**

Le rôle de ce module est de vérifier le bon positionnement d'une pièce à usiner. Si le trou est orienté vers le haut alors l'insertion de la bobine de contrôle atteint sa position basse finale. Un capteur de proximité est activé par un écrou situé à la fin supérieure de l'induit[69].

- **Module de perçage**

L'objectif du module de perçage est de percer un trou au milieu d'une pièce à usiner. Un vérin de serrage à double effet est utilisé pour fixer la pièce à usiner. Les actions

d'avance et de retour de la perceuse sont effectuées au moyen d'un axe linéaire avec entraînement par courroie crantée. La détection de la position finale s'effectue au moyen de capteurs de fin de course électriques. L'unité de perçage est alimentée par un moteur 24 V DC et le nombre de tours n'est pas réglable[69].

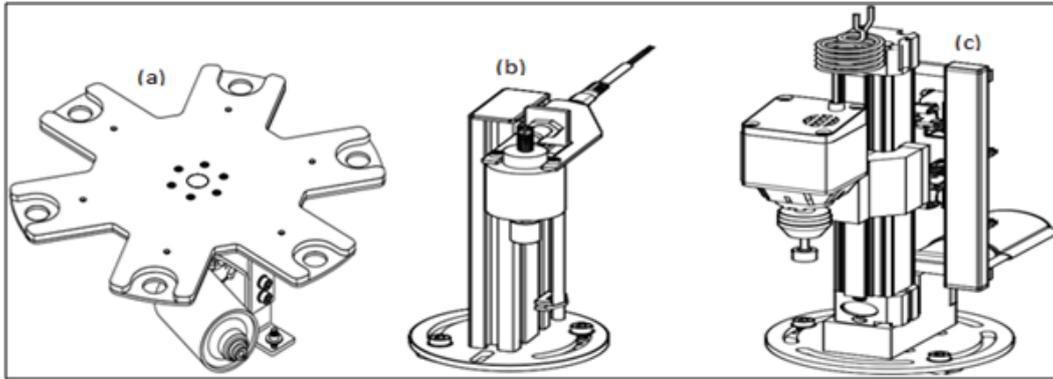


FIG. 3.4 : Principaux composants de la sous-station d'usinage[69]. - (a) Module table à indexation rotative. - (b) Module de contrôle. - (c) Module de perçage.

3.2.2.2 Cahier de charge initial de la sous-station d'usinage

Initialement, l'induit de la bobine de contrôle est en haut, la perceuse en position haute et son moteur est à l'arrêt, le dispositif de serrage en position rentrée et le plateau à indexation rotative en position équilibrée. Le plateau d'indexation rotatif est tourné de 60° et pour que ce plateau puisse démarrer, la pièce doit se trouver dans le réceptacle "Entré matériau".

Si une pièce à usiner est détectée dans le réceptacle "Entrée du matériau" et si le bouton-poussoir START est enfoncé, le plateau à indexation rotative est tourné de 60° jusqu'à ce qu'il atteigne l'unité de contrôle. Lorsque la pièce à usiner est présente devant l'unité de contrôle, le plateau d'indexation s'arrête, l'induit de la bobine de contrôle descend et vérifie si la pièce est posée avec l'ouverture vers le haut. Le plateau à indexation rotative est tourné de 60° si le résultat du contrôle est OK.

En présence de la pièce dans l'unité de perçage, le plateau d'indexation s'arrête, le dispositif de serrage serre la pièce et le moteur de la perceuse est mis en marche, l'axe linéaire déplace la perceuse vers le bas. Quand la perceuse atteint sa position inférieure, puis elle remonte vers sa position initiale (position supérieure) par l'axe linéaire, le moteur de la perceuse est mis hors tension et le dispositif de serrage est rétracté et le plateau à indexation rotative est tourné de 60° . Lorsque le contrôle et le perçage sont terminés, la pièce est en position de transfert vers la station suivante. En mode automatique le cycle de traitement peut être relancé dès qu'une pièce arrive sur le réceptacle 1.

3.2.2.3 Modifications apportées sur le cahier de charge de la sous-station d'usinage

Dans le fonctionnement actuel, le traitement d'une pièce dans la station d'usinage se fait sur un seul passage via la machine de contrôle et la machine de perçage, dans

Chapitre 3. La programmation, simulation et l'injection au niveau des sous-stations (MPS-variant)

le cahier de charge que nous proposons, nous avons supposé de faire un seul traitement dans ces machines n'est pas suffisant, donc nous proposons d'ajouter un deuxième traitement, à cause du problème d'ajustement de la table dans le centre des différents capteurs de présences. À la fin du deuxième traitement la pièce va être transféré vers la station suivante.

Selon le fonctionnement habituel, la position du plateau d'indexation est détectée par un capteur inductif B3, parfois le capteur ne nous donne pas l'information correcte sur le positionnement de la table ce qui provoque des problèmes durant le fonctionnement. La solution que nous avons envisagée à ce problème est d'ajouter un temporisateur pour détecter la position du plateau d'indexation d'une manière exacte.

3.2.3 Sous-station de manipulation

Cette station sert d'intermédiaire entre les stations usinage et livraison, elle va être utilisée pour la manutention des pièces à usiner à l'aide d'un reboot cartésien, qui se déplace selon deux axes XX' et YY'. Ce déplacement est effectué soit entre les stations, soit par le rejet des pièces dans les glissières de la station manipulation. Cette station est équipée d'un dispositif de manipulation flexible "Manipulateur" à deux axes XX' et YY. Le mouvement sur l'axe XX', pour atteindre la station amont et aval, est effectué à l'aide de deux moteurs (1M1) et (1M2) respectivement. Le dispositif de manipulation extrait les pièces noires, métalliques et rouges de manière aléatoire à l'aide d'une pince pneumatique (2M1 sortir la pince et 3M1 ouvrir la pince). Ses pièces sont amenées à la station suivante au moyen du manipulateur et de la pince. La figure 3.5 illustre la sous-station de manipulation[69].

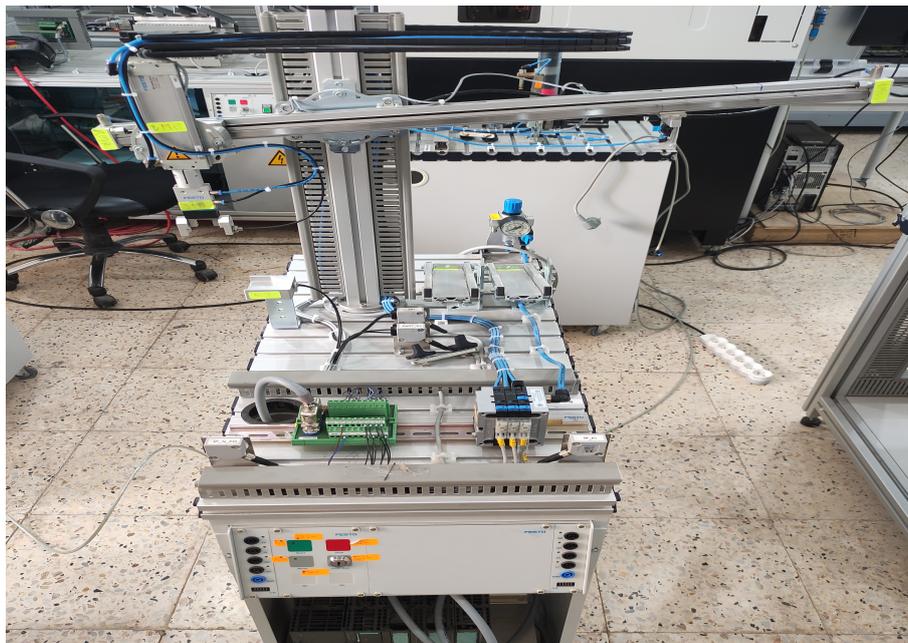


FIG. 3.5 : Sous-station de manipulation du système FESTO MPS.

3.2.3.1 Principaux composants de la sous-station de manipulation

- Module de manipulateur "PickAlfa"

Le module PickAlfa est le composant principal de la station de manipulation. Un axe linéaire pneumatique avec régulation souple de la fin de course et un amortissement permettent un positionnement rapide[69]. Le vérin de levage de l'axe Z est un vérin linéaire plat avec une détection des extrémités de la course. Le vérin de levage est équipé d'une pince pneumatique (il est utilisé pour attraper et relâcher les pièces selon les besoins). Le capteur optique incorporé dans la mâchoire de la pince permet la détection des pièces.

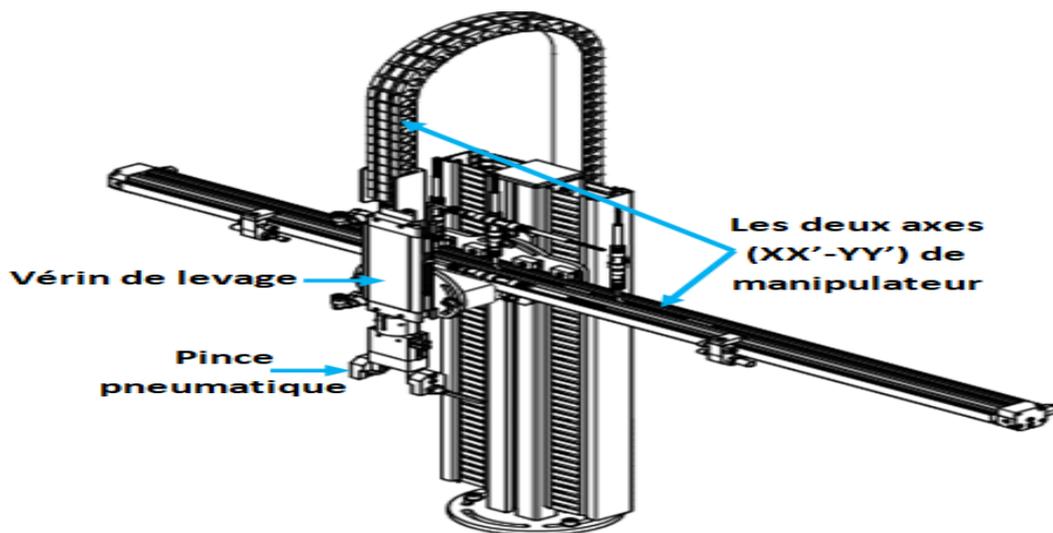


FIG. 3.6 : Module de manipulateur "PickAlfa"[69].

3.2.3.2 Cahier de charge initial de la sous-station de manipulation

En condition initiale, nous trouvons l'axe linéaire en position "Station en amont", le vérin de levage rentré (pince en haut et ouverte). Si une pièce est détectée dans le réceptacle et que l'on appuie sur le bouton START, le vérin de levage sort avec le maintien de l'ouverture de la pince. La pince attrape la pièce et se ferme. Quand la pièce est détectée au niveau de la pince, le vérin de levage rentre, le robot cartésien se déplace vers la station en aval, la translation du robot cartésien se fait à l'aide des capteurs de fin de course électromagnétique. Le vérin de levage sort, la pince s'ouvre et libère la pièce. Lorsque la tige du vérin revient à sa position initiale, la station retourne à la station amont.

3.2.3.3 Modifications apportées sur le cahier de charge de la sous-station de manipulation

Comme nous l'avons dit dans la description, cette station joue le rôle d'intermédiaire entre deux sous-stations (usinage et livraison) et qui va transporter les pièces à usiner à l'aide d'un robot cartésien qui se déplace via deux axes XX' et YY' . Après calibrage des différents capteurs de la sous-station manipulation et pour faire une liaison synchronisée

Chapitre 3. La programmation, simulation et l'injection au niveau des sous-stations (MPS-variant)

entre les autres sous-stations, nous allons proposer l'état initial comme suit : L'axe linéaire en position station en amont (station d'usinage), le vérin de levage rentré (pince en haut et ouverte). Lorsque la pièce est détectée dans la sous-station d'usinage et après l'appuie sur le bouton START, le vérin de levage sort avec une pince ouverte. Quand la pièce est détectée au niveau de la pince, cette dernière attrape la pièce et se ferme. Le vérin de levage rentre, le robot se déplace vers la station en aval (station de livraison). Le vérin de levage sort, la pince s'ouvre et après un temps d'attente, elle libère la pièce. Lorsque la tige du vérin revient à sa position initiale, le robot de la sous-station retourne à la station en amont (station d'usinage).

3.2.4 Sous-station de livraison (tri)

La fonction de la station de tri est de détecter la couleur des pièces entrantes et les dépose sur l'un des trois glissières en fonction de la couleur (noir, rouge ou métallique). Chaque glissière peut contenir jusqu'à 6 pièces. Deux déviations (1M1 et 2M1) actionnées par des vérins à faible course via un mécanisme d'inversion permettent de sélectionner la glissière sur laquelle les pièces doivent être placées[69]. Un seul capteur de réflexion surveille les trois niveaux de charge des glissières avec un inconvénient qui provoque l'arrêt du système lorsque l'une des trois glissières se charge. Le transport des pièces se fait par le module de bande de tri (K1). La figure 3.7 illustre la sous-station de livraison.

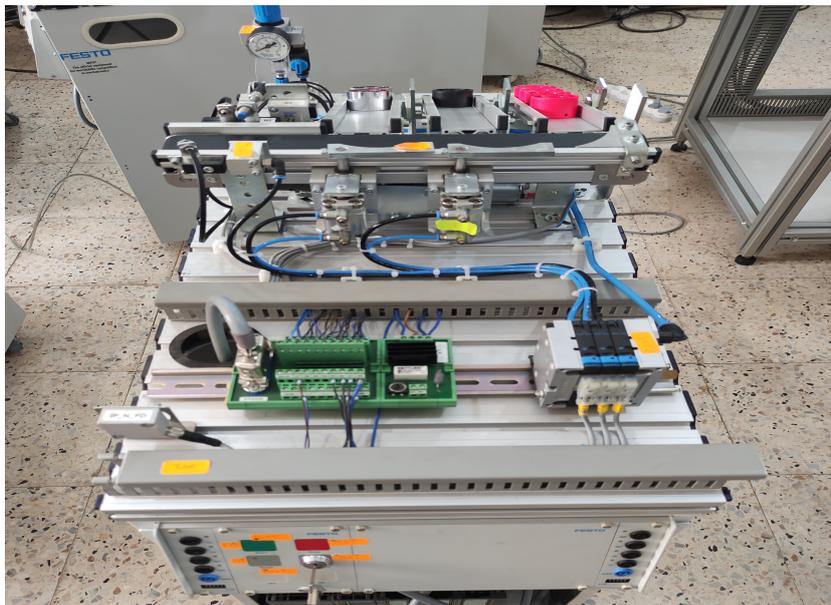


FIG. 3.7 : Sous-station de livraison (tri) du système FESTO MPS.

3.2.4.1 Principaux composants de la sous-station de livraison (tri)

- Module de glissière

Le module de glissière permet de transporter ou de stocker des pièces. Ce module peut être utilisé de manière universelle grâce au réglage variable de l'inclinaison et de la hauteur[69].

- **Module de bande de tri**

Le module de la bande de tri permet de transporter les pièces. Des vérins à faible course intégrés permettent de commuter deux dérivations de matériel. Il est donc possible de trier les pièces selon leurs caractéristiques ou leurs types. Le convoyeur de triage est actionné à l'aide d'un motoréducteur à courant continu[69].

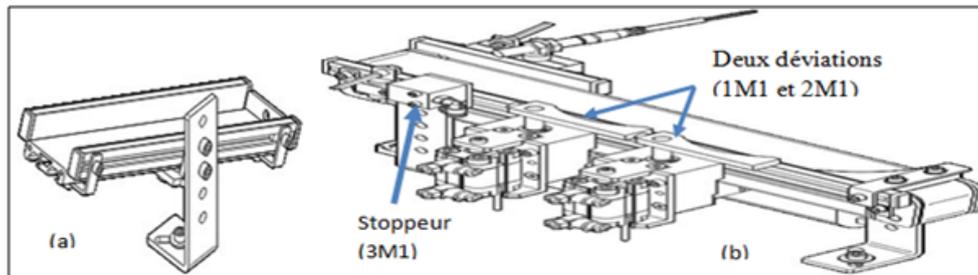


FIG. 3.8 : Principaux composants de la sous-station de tri[69]. - (a) Module de glissière.
- (b) Module de bande de tri.

3.2.4.2 Cahier de charge initial de la sous-station de livraison

Au départ, aucune pièce à usiner n'est présente au début de la bande, le stoppeur est sorti, les deux dérivations sont rétractées et le moteur de la bande est mis en arrêt. Si une pièce à usiner est identifiée par le capteur (Part-AV) et que l'on appuie sur la touche START, le moteur de la bande s'active. Lorsque la pièce à usiner arrive aux capteurs d'identification de couleur/matériau, elle va être bloquée par le stoppeur pendant un certain temps afin d'être identifiés. Les pièces vont être déposées sur les trois glissières en fonction de leur nature (rouge, métallique, noir).

- Dans le cas où la pièce à usiner est rouge, on la dépose sur la glissière en début de bande :

Lorsque la pièce rouge est identifiée via un capteur optique à proximité, le stoppeur se rétracte et permet à la pièce de passer, la dérivation 1 sort de manière à faire glisser la pièce dans la glissière qui se trouve au début de la bande puis se rétracte. Ensuite le moteur de la bande sera désactivé.

- Dans le cas où la pièce à usiner est métallique, on la dépose sur la glissière en milieu de bande :

Lorsque la pièce métallique est identifiée via un capteur inductif à proximité, le stoppeur se rétracte et permet à la pièce de passer, la dérivation 2 sort de manière à faire glisser la pièce dans la glissière du milieu de la bande puis se rétracte. Ensuite le moteur de la bande sera désactivé.

- Dans le cas où la pièce à usiner est noire, on la dépose sur la glissière située en fin de bande :

Lorsque la pièce n'est pas identifiée par aucun capteur ni par le capteur inductif à proximité ni par le capteur optique à proximité, alors elle c'est une pièce noire, le stoppeur se rétracte et permet à la pièce de passer jusqu'à ce qu'elle arrive à la fin de la bande avec laquelle elle va glisser dans la dernière glissière. Puis le moteur de la bande sera désactivé.

Si une des glissières est pleine ou si son capteur ne reçoit pas de réflexion, le système s'arrête et le voyant de la glissière pleine s'allume.

3.2.4.3 Modifications apportées sur le cahier de charge de la sous-station de tri

L'inconvénient du fonctionnement actuel de cette station, c'est que si l'une des glissières est pleine, le système s'arrête directement. Pour régler ce problème, nous allons trier les pièces dans les trois glissières et remplir les trois glissières avant l'arrêt du système. Chaque glissière peut contenir jusqu'à 6 pièces, donc le remplissage des glissières va se faire de cette façon :

- C1 (glissière N°01 en début de bande) : **6 pièces métalliques.**
- C2 (glissière N°02 en milieu de bande) : **6 pièces noires.**
- C3 (glissière N°03 en fin de bande) : **6 pièces rouges.**

Au départ, aucune pièce à usiner n'est présente au début de la bande, le stoppeur est sorti, les deux dérives sont rétractées et le moteur de la bande est mis à l'arrêt. Quand une pièce est identifiée par le capteur (Part-AV) et que l'on appuie sur la touche START, le moteur de la bande s'active. Lorsque la pièce arrive aux capteurs d'identification de couleur/matériau qui va être bloqué par le stoppeur pendant un certain temps afin d'être identifiés. Une fois l'identification est faite le stoppeur va laisser la pièce passer puis il revient à sa position initiale sortie.

- Le temps où la pièce bloquée par le stoppeur est : 1.695s

La durée entre le stoppeur et chaque dérivation ou l'extrémité de la bande est mesurée à l'aide d'un chronomètre afin de minimiser l'énergie dissipée du stoppeur et du convoyeur. Les durées avec lesquelles le stoppeur libère la pièce jusqu'à ce qu'il arrive aux glissières pour différentes positions :

- Le temps entre le stoppeur et la sortie de la dérivation 1 (la glissière C1) : 0.9275s
- Le temps entre le stoppeur et la sortie de la dérivation 2 (la glissière C2) : 1.5625s
- Le temps entre le stoppeur et la glissière C3 : 2.38s

Après que la pièce atteint sa propre glissière, elle sera évacuée par sa dérivation appropriée ou elle est automatiquement évacuée à la fin de la bande, tandis que la bande de transport s'arrête jusqu'à la détection d'une nouvelle pièce. La diffusion des pièces à usiner se fait de manière complètement aléatoire à partir de la sous-station source distribution et le nombre maximum des pièces au niveau du magasin est 19 pièces. Une fois la 19ème pièce arrive, une LED s'allume (glissières pleines) et la sous-station entière s'arrête pendant un temps suffisant pour décharger les glissières.

3.2.5 SADT des sous-stations

3.2.5.1 SADT de la sous-station de distribution

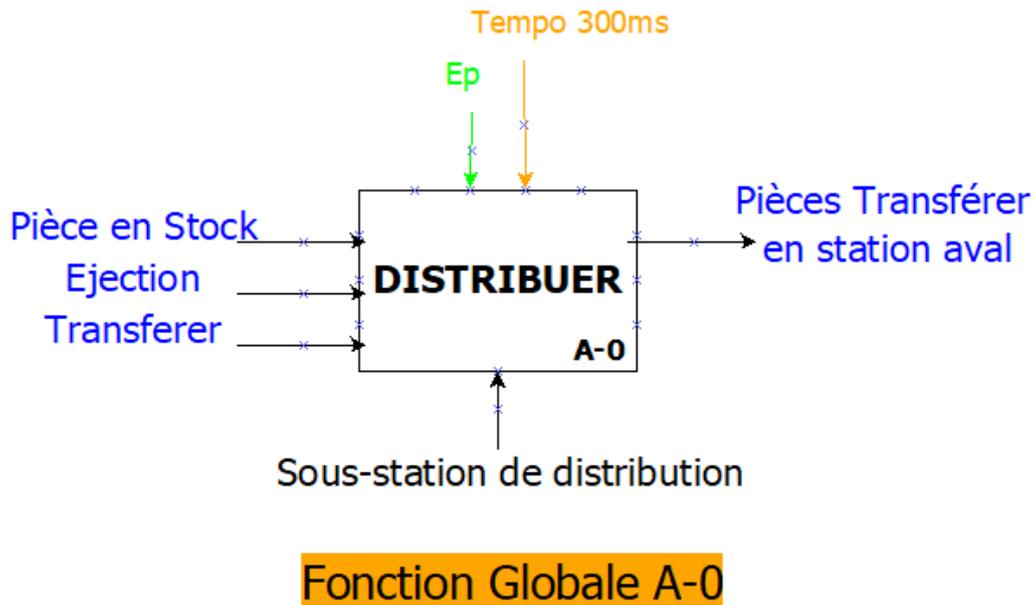


FIG. 3.9 : Fonction globale A-0 de la sous-station de distribution.

La figure 3.9 représente SADT A-0 de la sous station de distribution.

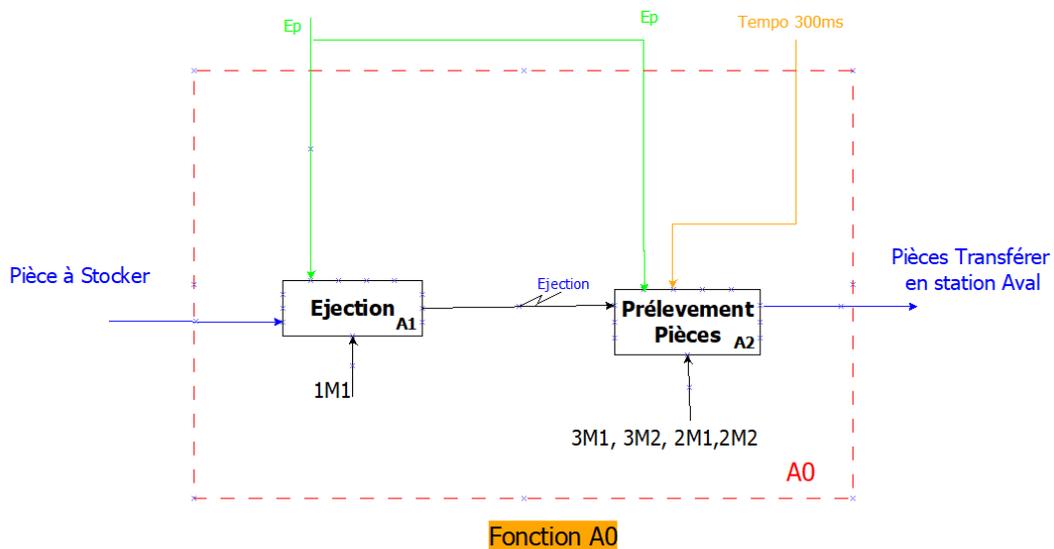


FIG. 3.10 : Fonction globale A0 de la sous-station de distribution.

La figure 3.10 représente SADT A0 de la sous station de distribution qui contient A1 d'éjection et A2 de prélèvement des pièces.

3.2.5.2 SADT de la sous-station d'usinage

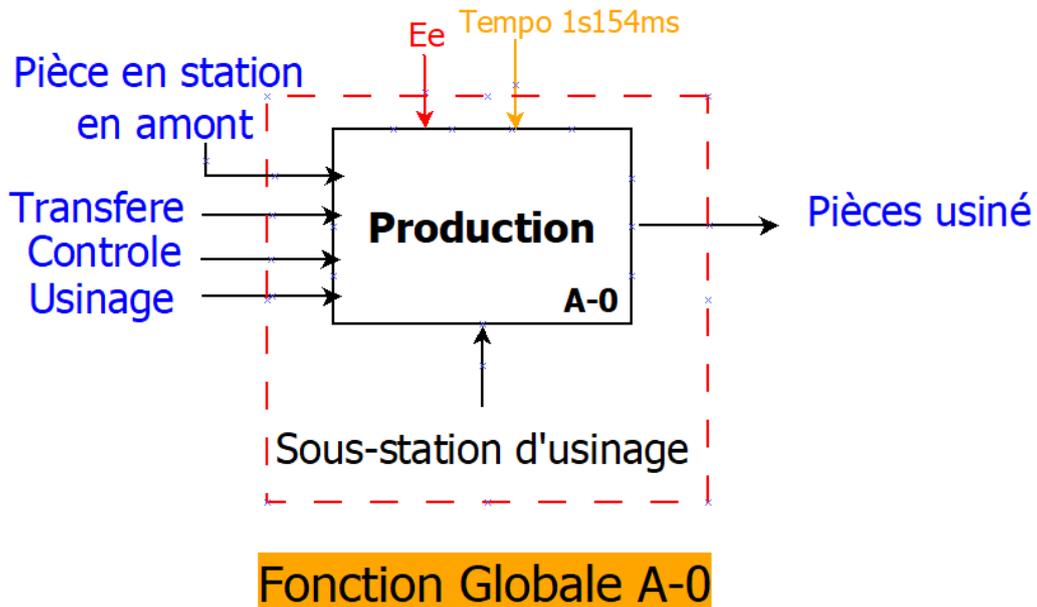


FIG. 3.11 : Fonction globale A-0 de la sous-station d'usinage.

La figure 3.11 représente SADT A-0 de la sous station d'usinage.

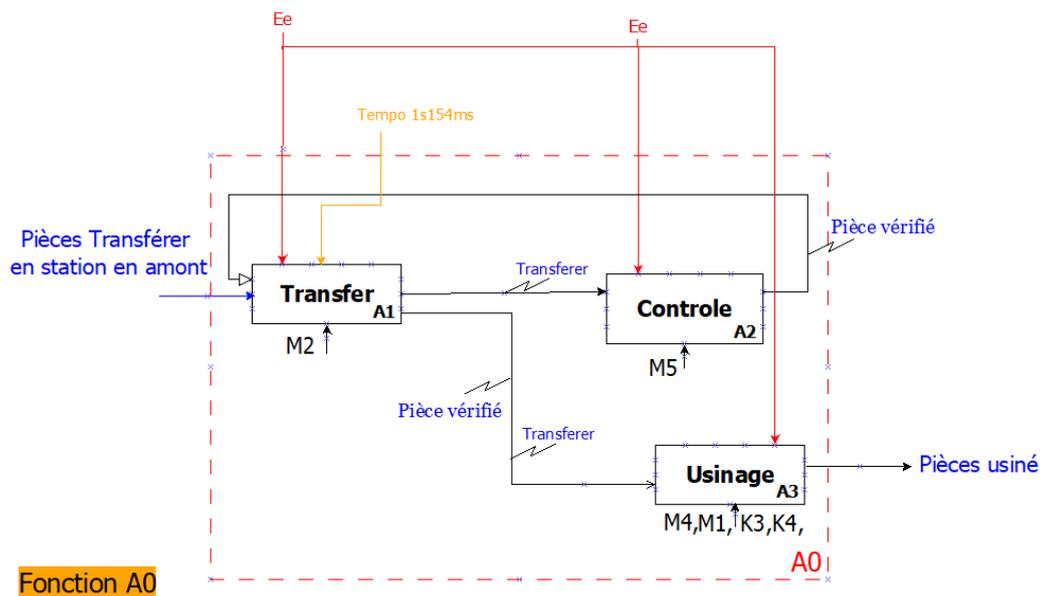
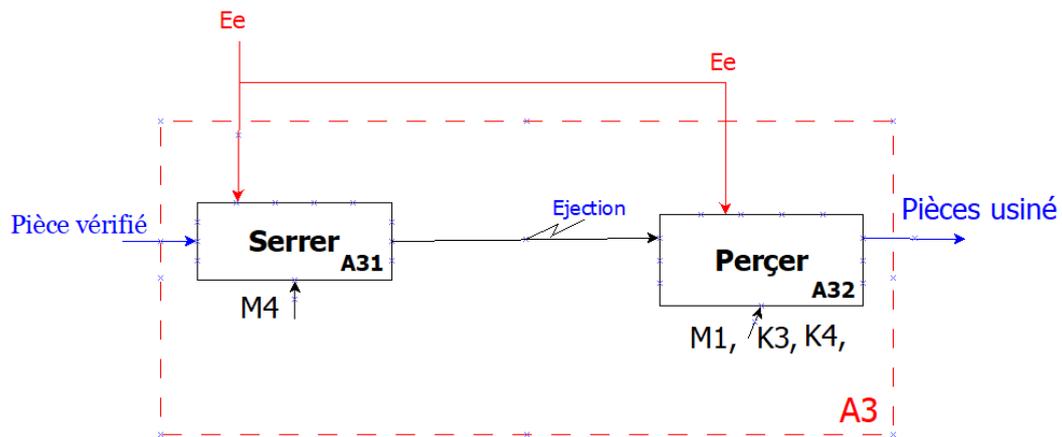


FIG. 3.12 : Fonction globale A0 de la sous-station d'usinage.

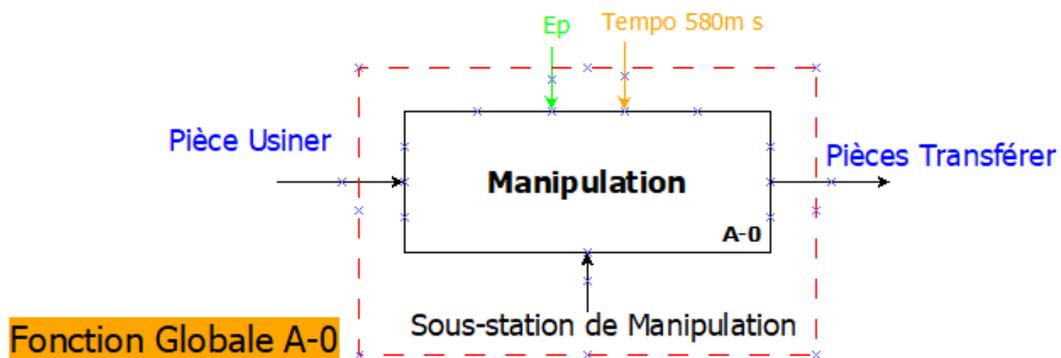
La figure 3.12 représente SADT A0 de la sous station d'usinage qui contient A1 de transfert et A2 de contrôle et A3 d'usinage. La figure 3.13 représente SADT A3 de la sous station d'usinage qui contient la partie A31 de serrage et A32 de perçage.



Fonction A3

FIG. 3.13 : Fonction globale A3 de la sous-station d'usinage.

3.2.5.3 SADT de la sous station de manipulation



Fonction Globale A-0

FIG. 3.14 : Fonction globale A-0 de la sous-station de manipulation.

La figure 3.14 représente SADT A-0 de la sous station de manipulation.

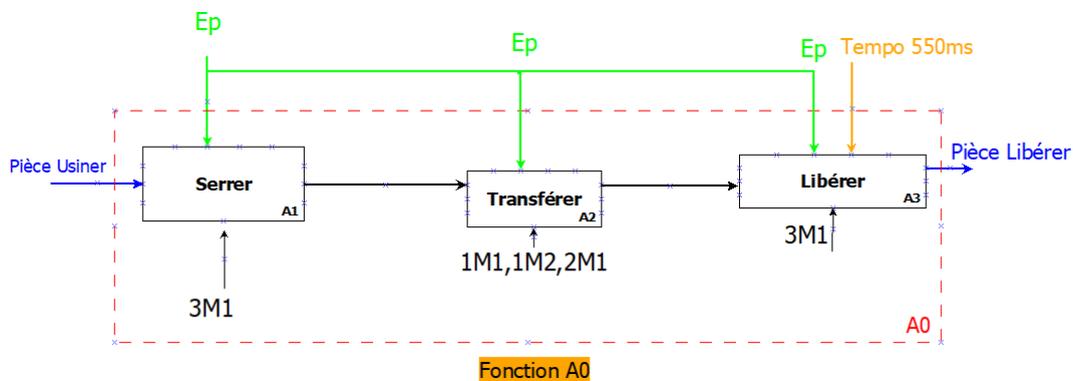


FIG. 3.15 : Fonction globale A0 de la sous-station de manipulation.

La figure 3.15 représente SADT A0 de la sous station de manipulation qui contient la partie A1 de serrage et A2 de transfert et la partie A3 qui libère la pièce.

3.2.5.4 SADT de la sous station de livraison

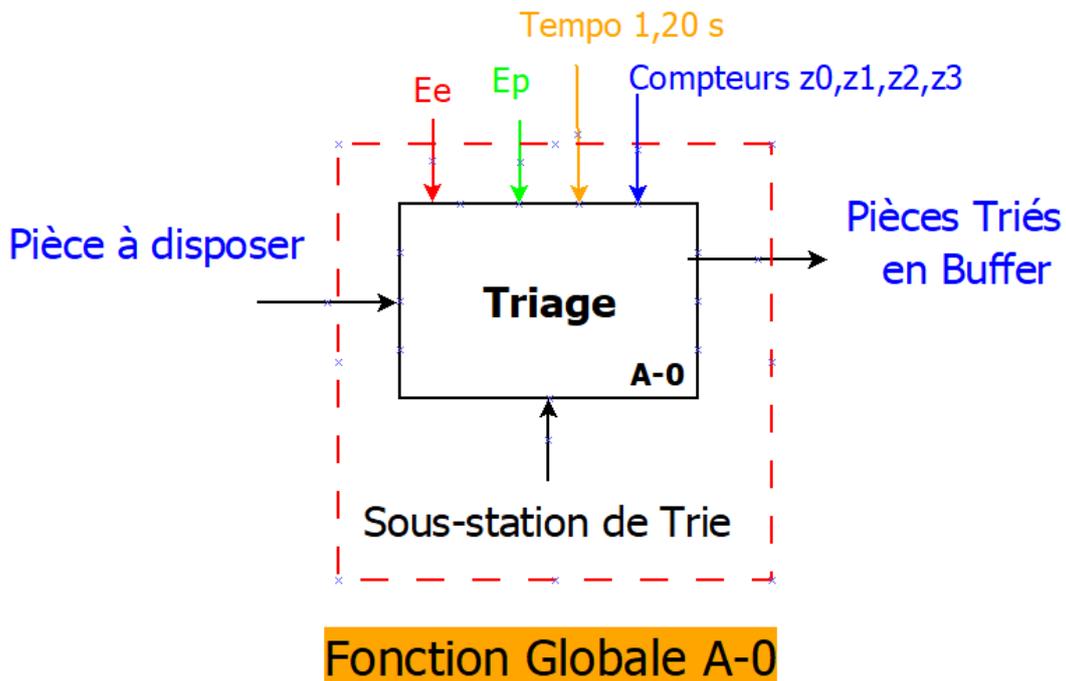


FIG. 3.16 : Fonction globale A-0 de la sous-station de livraison.

La figure 3.16 représente SADT A-0 de la sous station de livraison.

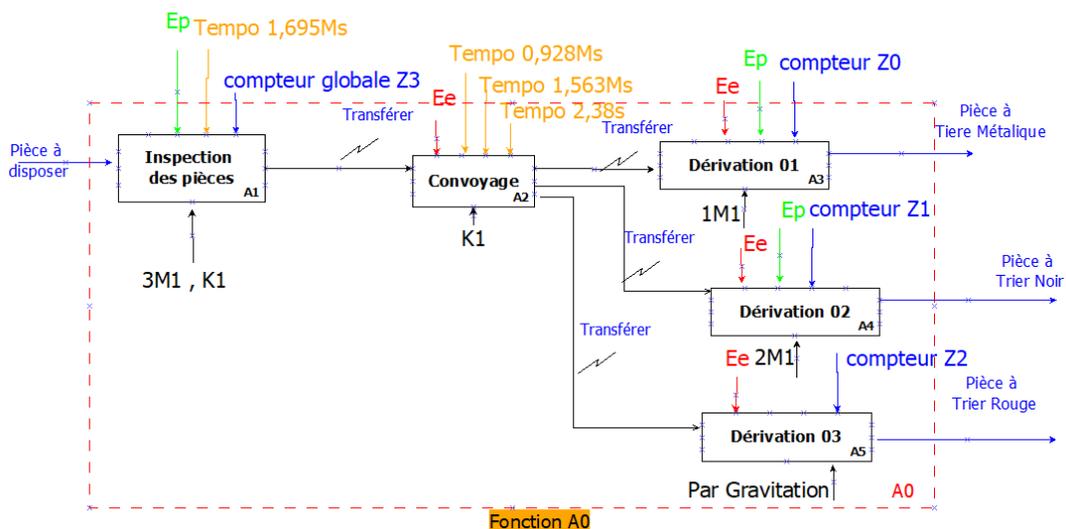


FIG. 3.17 : Fonction globale A0 de la sous-station de livraison.

La figure 3.17 représente SADT A0 de la sous station de livraison qui contient la partie A1 d'inspection des pièces et la partie A2 de convoyage et la partie A3 de dérivation 1 et la partie A4 de dérivation 02 ainsi la partie A5 de dérivation 3.

3.2.6 Pupitre de commande

Chaque station est composée de son propre pupitre de commande. Ce pupitre est utilisé pour commander la station à laquelle il est dédié, sa fonction principale et de permettre le lancement des actions par rapport à la station[69]. Le module du pupitre de commande se compose de trois boutons poussoirs avec voyant lumineux permettant à l'opérateur de contrôler (par exemple, Start (bouton vert), Stop (bouton rouge), Reset (bouton gris)) la station. Aussi une clé (Auto/Man) pour sélectionner le mode manuel ou mode automatique. Les voyants Q1 et Q2 sont dédiés à chaque sous-station par exemple, dans la sous-station distribution le voyant Q1 a pour rôle d'indiquer que le magasin est vide.

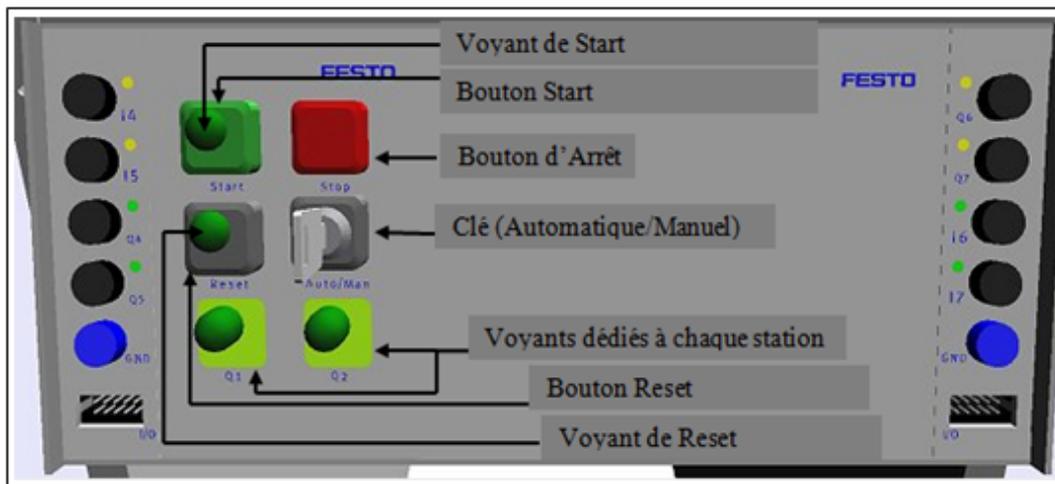


FIG. 3.18 : Module du pupitre de commande.

3.3 Aperçu de Step 7

STEP 7 est un logiciel pour la programmation d'automates programmables industriel (API) de type « Siemens », il permet de programmer un automate de manière individuelle (dans différents langages par exemple, « CONT, LOG, LIST, GRAPH ou SCL »), il permet aussi de tester, mettre en service et maintenir les systèmes d'automatisation SIMATIC. Il tient compte également du réseau, qui permet l'accès à n'importe quel API du réseau (pour le programmer). Le logiciel est nommé "SIMATIC Manager", en général accessible dans le menu de démarrage sous "Siemens".



FIG. 3.19 : L'icône du logiciel SIMATIC Manager.

3.3.1 Démarrage de Step 7

Pour lancer le logiciel STEP 7, il suffit de double-clic sur l'icône SIMATIC Manager sur le bureau Windows.

3.3.2 Création d'un projet

La création d'un nouveau projet nécessite la description de deux parties principales : la configuration du matériel et la configuration du fonctionnement.

Une fois SIMATIC MANAGER s'ouvre, une fenêtre d'assistant s'affiche pour créer rapidement un projet STEP 7. Dans notre cas, nous allons annuler la fenêtre de la figure 3.20 en appuyant sur « Annuler » afin de pouvoir choisir le CPU.

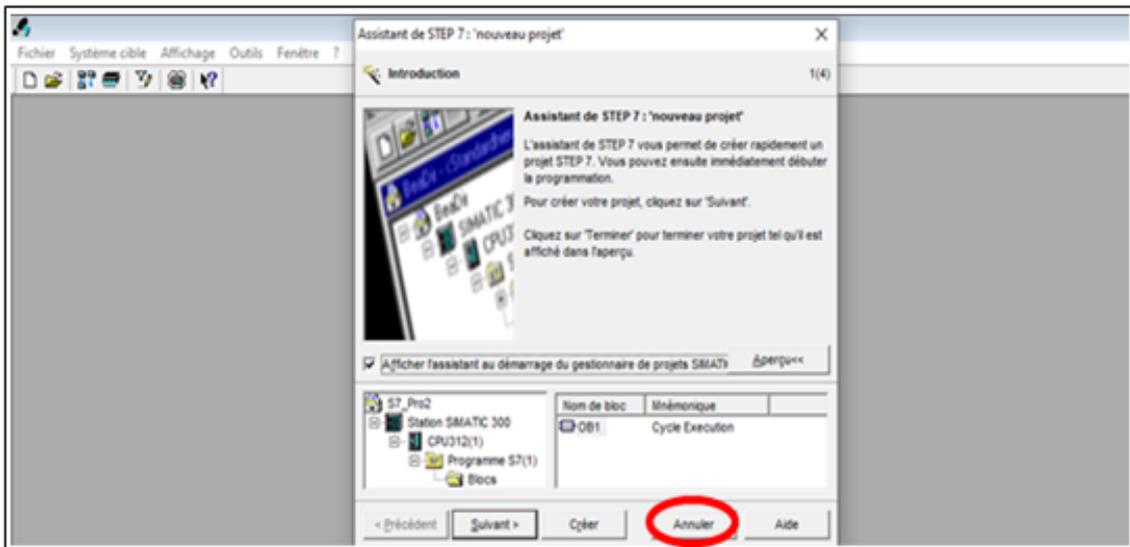


FIG. 3.20 : Fenêtre d'assistant pour créer rapidement un projet Step 7.

Pour commencer la programmation, nous devons d'abord créer un nouveau projet en cliquant sur l'icône "nouveau" de la barre d'outils pour ouvrir la fenêtre "nouveau projet". Dans cette fenêtre, nous saisissons un nom pour notre projet dans le champ "nom", le chemin d'accès à notre projet enregistré est indiqué dans le champ "emplacement de stockage", puis nous cliquons sur "OK". Mais nous pouvons utiliser le bouton "Parcourir" pour enregistrer le projet dans un autre emplacement que celui par défaut.

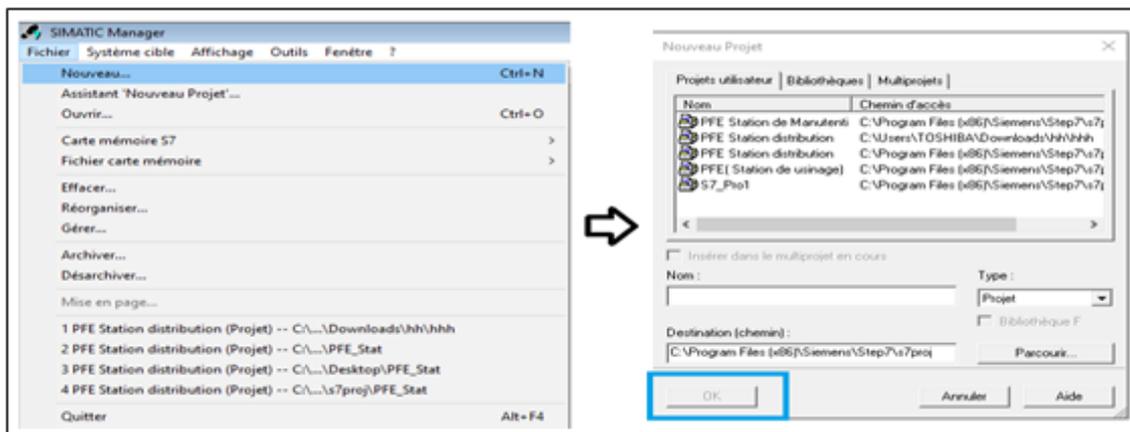


FIG. 3.21 : Création d'un nouveau projet.

3.3.3 Configuration matérielle

Dans un projet, la station représente la structure matérielle d'un automate programmable et contient les données permettant de configurer et d'affecter des paramètres aux différents modules. Après avoir créé le projet, nous allons sélectionner notre projet en cliquant sur son nom, puis dans le menu "Insertion", nous choisissons « Station » puis nous sélectionnons "Station SIMATIC 300".

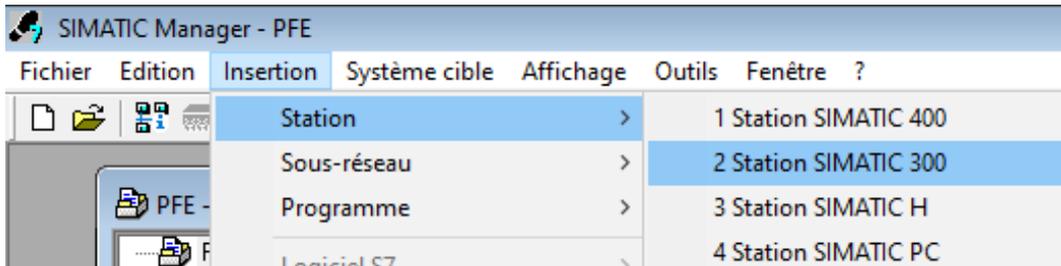


FIG. 3.22 : Fenêtre d'insertion d'une station.

Après avoir sélectionné "Station 300", nous voyons qu'un sous-groupe nommé "SIMATIC 300" a été ajouté à notre projet.

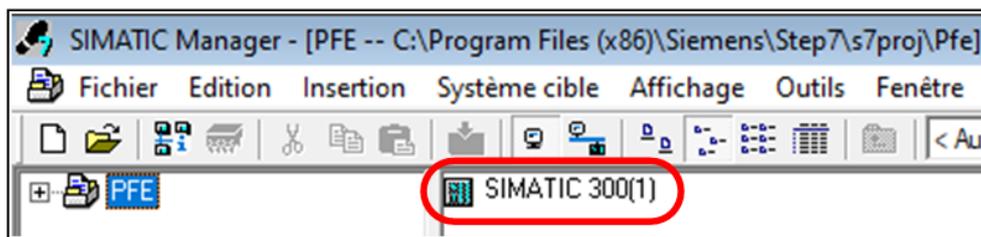


FIG. 3.23 : La station SIMATIC 300.

Quand nous faisons une double clique sur le SIMATIC 300, la fenêtre de la figure 3.24 s'ouvre, donc le "Matériel (Hardware)" a été ajouté au projet.

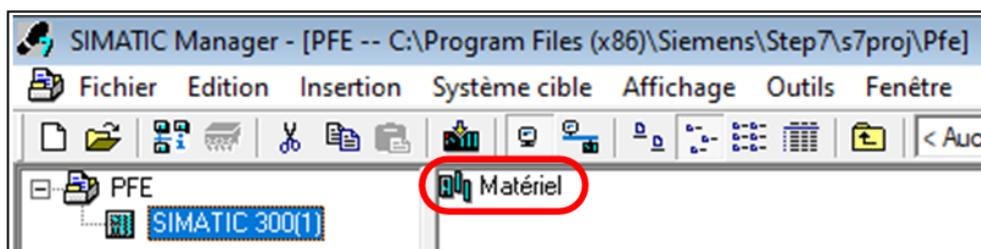


FIG. 3.24 : Partie matérielle.

Nous cliquons sur l'icône "matériel" qui nous amène au matériel, afin que nous configurions les modules dont nous avons besoin pour l'automate. Notre API est un "S7 300", donc nous devons sélectionner les modules appropriés à ce sous-groupe.

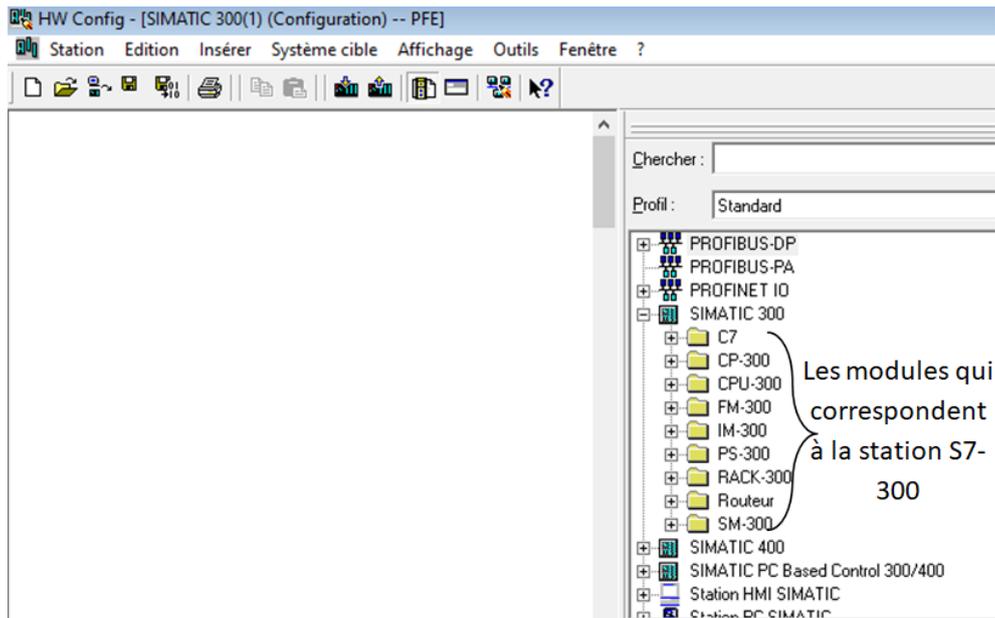


FIG. 3.25 : Configuration matérielle (HW Config).

Nous avons d'abord besoin d'un rack dans lequel nous allons intégrer les modules. Pour insérer le rack, nous ouvrons le "rack 300" dans le catalogue et nous glissons/déposons l'objet "Support Profile" dans la partie installation du matériel (à gauche de l'écran). Une petite fenêtre nommée "(0) UR" s'affiche pour insérer les modules.

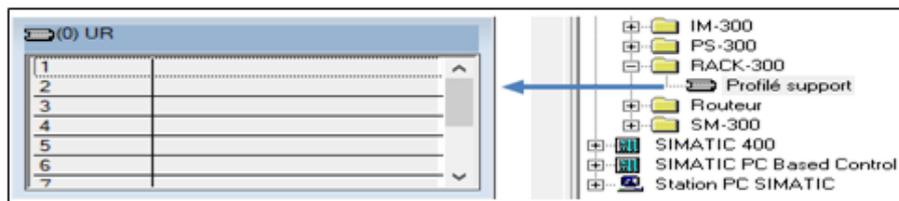


FIG. 3.26 : Insertion du Rack-300.

Nous devons installer une "alimentation" dans le premier emplacement pour alimenter les modules de l'automate. Pour cela, nous ouvrons le dossier "PS 300" et nous glissons/déposons l'alimentation "PS 307 5A" dans le premier slot du rack.

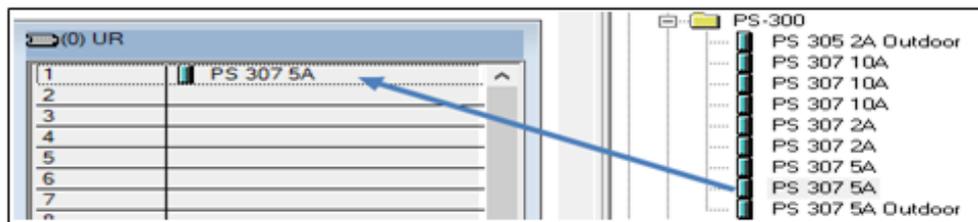


FIG. 3.27 : Insertion d'une alimentation PS 307 5A.

Dans le deuxième emplacement du rack, nous introduisons toujours le module "CPU". Nous allons sélectionner la "CPU 314", nous devons donc ouvrir le dossier "CPU 314"

Chapitre 3. La programmation, simulation et l'injection au niveau des sous-stations (MPS-variant)

et choisir la référence appropriée "CPU 314C-2 PN/DP" pour l'insérer dans le deuxième emplacement du rack.

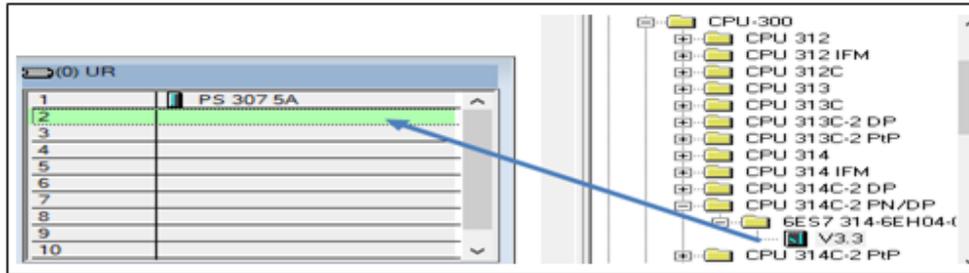


FIG. 3.28 : Insertion CPU 314C-2 PN/DP

3.3.4 Communication via câble Ethernet

Lorsque nous glissons/déposons le CPU approprié dans le rack, une fenêtre ETHERNET apparaît, donc nous cliquons sur "Nouveau", une autre fenêtre apparaît. Ensuite, nous cliquons sur OK, un câble ETHERNET sera ajouté. Et puis nous cliquons sur OK. Ces étapes sont illustrées dans la figure suivante.

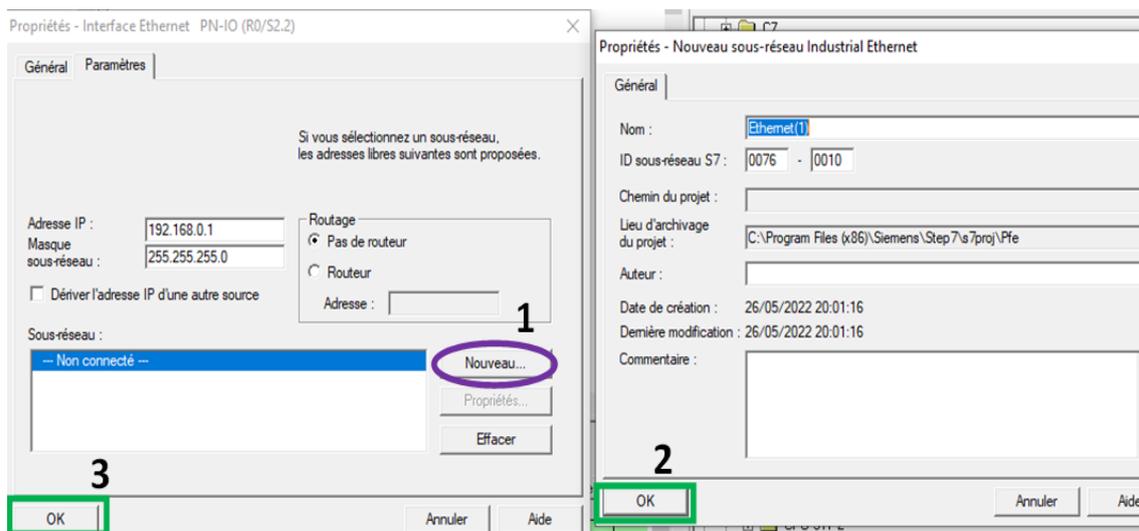


FIG. 3.29 : Choix de la communication via câble Ethernet.

3.3.5 Configuration du réseau

Pour configurer le réseau de notre projet, nous avons suivi les étapes ci-dessous :

1. Nous Cliquons sur "Simulation du module", la fenêtre de « PLCISIM » s'ouvre, cette étape est essentielle, car elle permet au logiciel d'ouvrir la fenêtre de configuration du réseau principal.
2. Nous Cliquons sur "Charger dans le module".

Chapitre 3. La programmation, simulation et l'injection au niveau des sous-stations (MPS-variant)

3. Nous passons vers la liaison réseau ou NETPRO, pour relier la station SIMATIC 300 et les deux réseaux MPI et ETHERNET.
4. Nous glissons "connexion au réseau de couleur rouge dans la station SIMATIC 300" dans « MPI » (ligne en rouge)". La ligne en vert de connexion ETHERNET est déjà connectée à la station SIMATIC 300.
5. À la fin, nous cliquons sur "enregistrer et compiler" pour valider les étapes précédentes.

Les étapes précédentes sont illustrées dans la figure 3.30 ci-dessous :

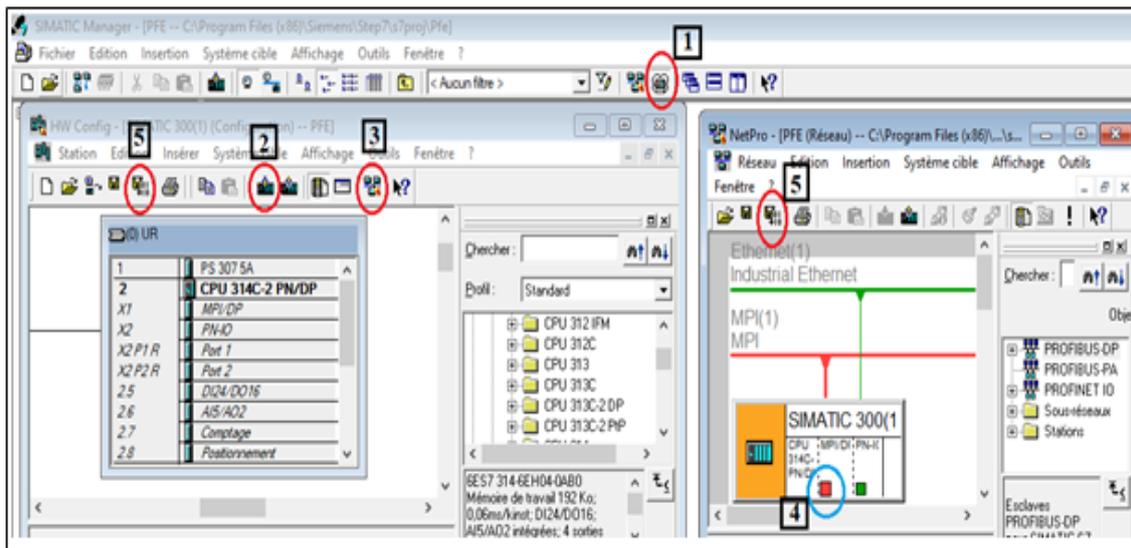


FIG. 3.30 : Fenêtre de configuration réseau.

Nous avons choisi de travailler avec un intervalle d'adresses d'entrées compris entre "E 0.0" et "E 1.7" et un intervalle d'adresses de sorties compris entre "A 0.0" jusqu'à "A 1.7". On a changé les adresses d'entrées/sorties par défaut du système.

Module	R...	A...	Adresse d'entrée	Adresse de sortie
PS 307 5A	6ES7			
CPU 314C-2 PN/DP	6ES7 2			
MPI/DP		2	2046 [°]	
PN-ID			2045 [°]	
Port 1			2044 [°]	
Port 2			2043 [°]	
DI24/DO16			136...138	136...137

Module	R...	A...	Adresse d'entrée	Adresse de sortie
PS 307 5A	6ES7			
CPU 314C-2 PN/DP	6ES7 2			
MPI/DP		2	2046 [°]	
PN-ID			2045 [°]	
Port 1			2044 [°]	
Port 2			2043 [°]	
DI24/DO16			0..2	0..1

FIG. 3.31 : Configuration des adresses d'entrées/sorties.

Après avoir terminé la configuration matérielle, nous pouvons consulter la structure du projet et voir qu'un sous-groupe portant le même nom que le CPU a été ajouté "CPU 314C-2 PN/DP", en cliquant dessus, nous trouvons le dossier « programme S7(4) » voir la figure 3.32 ci-dessous :

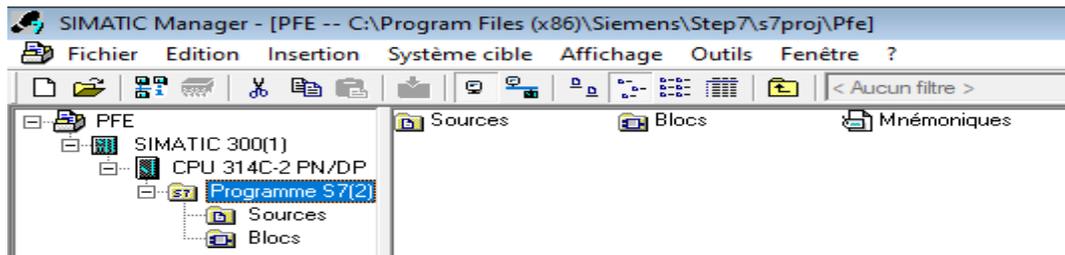


FIG. 3.32 : Hiérarchie du programme STEP 7.

Une fois la description du matériel terminée, il nous reste à entamer la partie programmation. Le dossier "Programme S7 (4)" contient le programme dans deux sous-dossiers distincts : les codes "Sources" et les "Blocs". Le sous-dossier Sources contient les codes sources de tous les blocs programmés en SCL ou STL. L'autre sous-dossier Blocs, on trouve le bloc principal OB1 et tous les autres Blocs appelés par ce dernier (FB, FC, DB, etc.). Il existe aussi un fichier "Mnémoniques" qui comprend une liste de variables. Les variables sont liées aux zones de mémoire suivantes :

- E : Zone Mémoire des entrées.
- A : Zone Mémoire des sorties.
- M : Zone Mémoire utilisateur (mémoire Internes), (Mémentos).
- T : Zone Mémoire des temporisations.
- Z : Zone Mémoire des compteurs.

3.3.6 Modélisation de notre système avec GRAFCET

Dans notre projet, le langage de programmation utilisé est le Grafcet. Pour cela nous devons créer un bloc fonctionnel (FB) en Graph. Une fenêtre de propriétés s'ouvre, où on choisit le nom et le langage de création (on choisit GRAPH). Il nous suffit de cliquer sur le Bloc fonctionnel (FB1) pour accéder à l'environnement de programmation.

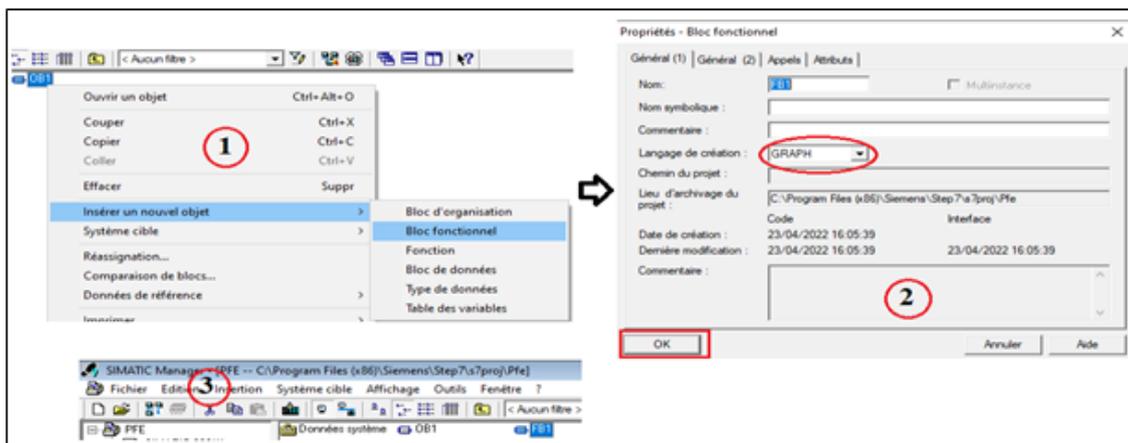


FIG. 3.33 : Création d'un GRAFCET.

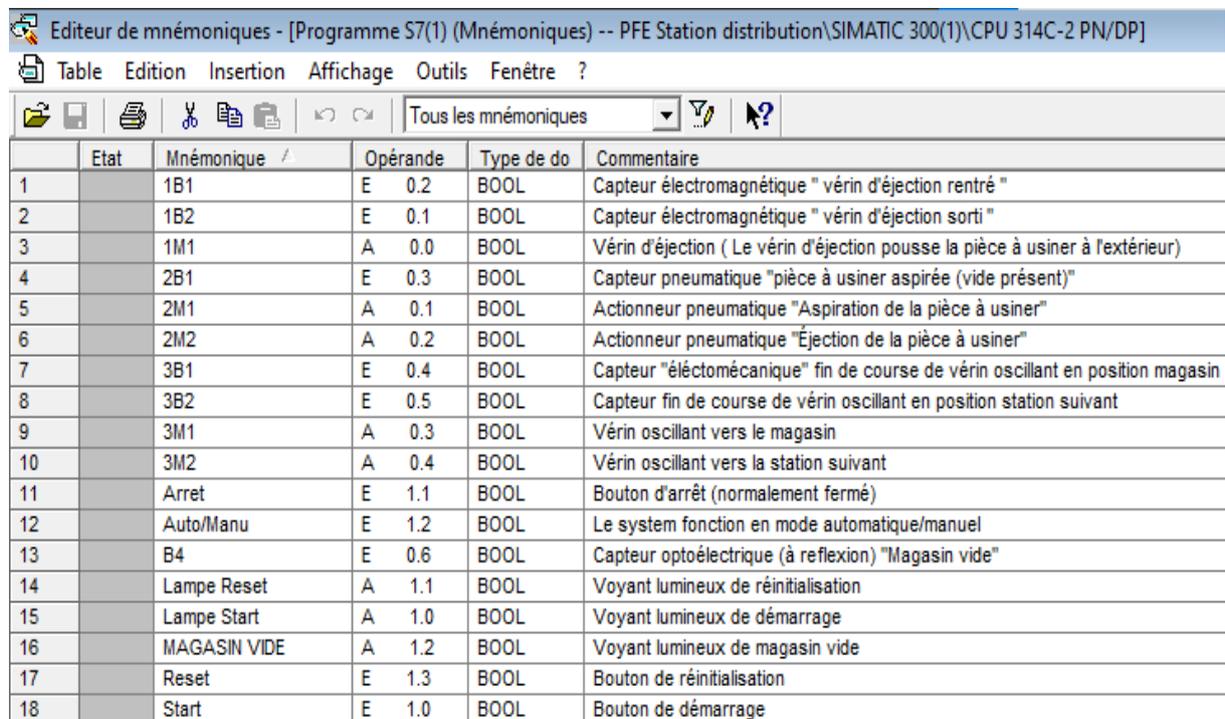
3.3.7 Appel du FB1 Depuis le bloc OB1

Pour que l'automate exécute le Grafcet, il faut que le bloc fonctionnel contenant le Grafcet [FB1] soit appelé depuis le bloc de programme principal Main [OB1]. L'ajout du bloc de programme se fait simplement en glissant le bloc de programme fonctionnel [FB1] dans le bloc de programme principal Main [OB1].

3.4 Modélisation du MPS Variant

3.4.1 Sous-station de distribution

3.4.1.1 Déclaration des variables de la sous-station de distribution



	Etat	Mnémonique /	Opérande	Type de do	Commentaire
1		1B1	E 0.2	BOOL	Capteur électromagnétique " vérin d'éjection rentré "
2		1B2	E 0.1	BOOL	Capteur électromagnétique " vérin d'éjection sorti "
3		1M1	A 0.0	BOOL	Vérin d'éjection (Le vérin d'éjection pousse la pièce à usiner à l'extérieur)
4		2B1	E 0.3	BOOL	Capteur pneumatique "pièce à usiner aspirée (vide présent)"
5		2M1	A 0.1	BOOL	Actionneur pneumatique "Aspiration de la pièce à usiner"
6		2M2	A 0.2	BOOL	Actionneur pneumatique "Éjection de la pièce à usiner"
7		3B1	E 0.4	BOOL	Capteur "électromécanique" fin de course de vérin oscillant en position magasin
8		3B2	E 0.5	BOOL	Capteur fin de course de vérin oscillant en position station suivant
9		3M1	A 0.3	BOOL	Vérin oscillant vers le magasin
10		3M2	A 0.4	BOOL	Vérin oscillant vers la station suivant
11		Arret	E 1.1	BOOL	Bouton d'arrêt (normalement fermé)
12		Auto/Manu	E 1.2	BOOL	Le system fonction en mode automatique/manuel
13		B4	E 0.6	BOOL	Capteur optoélectrique (à reflexion) "Magasin vide"
14		Lampe Reset	A 1.1	BOOL	Voyant lumineux de réinitialisation
15		Lampe Start	A 1.0	BOOL	Voyant lumineux de démarrage
16		MAGASIN VIDE	A 1.2	BOOL	Voyant lumineux de magasin vide
17		Reset	E 1.3	BOOL	Bouton de réinitialisation
18		Start	E 1.0	BOOL	Bouton de démarrage

FIG. 3.34 : Table des variables la sous-station de distribution.

3.4.1.2 Création du Grafcet de la sous-station de distribution

Notre Grafcet est présenté sous forme de deux graphes, la première est de commande (marche, arrête, reset) et le deuxième représente le processus de distribution des pièces. Dans la première étape dans la figure 3.35 nous désactivons toutes les étapes (S1 OFF|S-ALL) et le voyant du Reset s'allume dès le début. La figure 3.35 montre une partie du Grafcet GS.

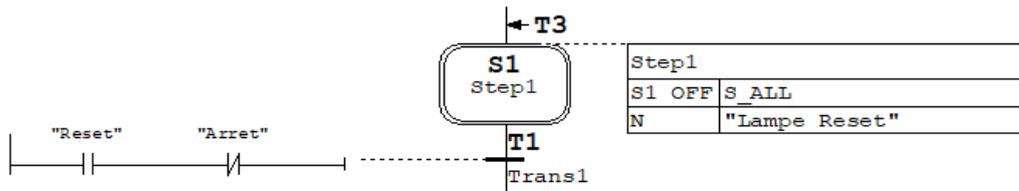


FIG. 3.35 : Une partie GS de la sous-station de distribution.

La figure 3.36 montre une partie du grafcet du processus de distribution des pièces, nous sommes dans la partie automatique, si le magasin est rempli de pièces, le vérin d'éjection pousse la pièce à l'extérieur.

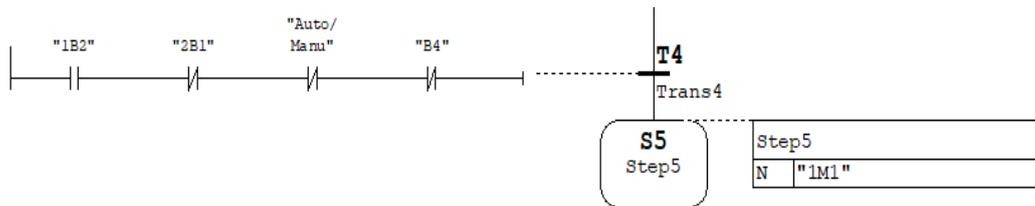


FIG. 3.36 : Une partie GPN de la sous-station distribution.

3.4.2 Sous-station d'usage

3.4.2.1 Déclaration des variables de la sous-station d'usage

Editeur de mnémoniques - [Programme S7(1) (Mnémoniques) -- PFE(usage VCompt)\SIMATIC 300(1)\CPU 314C-2 PN/D

Table Edition Insertion Affichage Outils Fenêtre ?

Tous les mnémoniques

	Etat	Mnémonique /	Opérande	Type de do	Commentaire
1		1B1	E 0.3	BOOL	Capteur électromécanique "Unité de perçage en position haute"
2		1B2	E 0.4	BOOL	Capteur électromécanique "Unité de perçage en position basse"
3		Arret	E 1.1	BOOL	Bouton d'arrêt (normalement fermé)
4		Auto/Manu	E 1.2	BOOL	Le system fonction soit en mode automatique/manuel
5		B1	E 0.2	BOOL	Capteur Capacitif "Pièce dans l'unité de contrôle"
6		B2	E 0.1	BOOL	Capteur Capacitif "Pièce à usiner à l'unité de perçage"
7		B3	E 0.5	BOOL	Capteur Inductif "Table d'indexation positionnée"
8		B4	E 0.6	BOOL	Capteur Inductif "Vérification du trou de perçage"
9		compteur	Z 0	COUNTER	Nombre de tour
10		G7_STD_3	FC 72	FC 72	
11		K3	A 0.2	BOOL	Unité de perçage basse
12		K4	A 0.3	BOOL	Unité de perçage haute
13		Lampe Reset	A 1.1	BOOL	Voyant de réinitialisation
14		Lampe Start	A 1.0	BOOL	Voyant lumineux de démarrage
15		M1	A 0.0	BOOL	Moteur de l'unité de perçage en marche
16		M2	A 0.1	BOOL	Moteur de la table d'indexation en marche
17		M4	A 0.4	BOOL	Pièce à fixer
18		M5	A 0.5	BOOL	Contrôler la pièce à usiner
19		Part_AV	E 0.0	BOOL	Capteur Capacitif "Pièce à usiner disponible"
20		Reset	E 1.3	BOOL	Bouton de réinitialisation
21		Start	E 1.0	BOOL	Bouton de démarrage
22		TIME_TCK	SFC 64	SFC 64	Read the System Time

FIG. 3.37 : Table des variables la sous-station d'usage

3.4.2.2 Création du Grafcet de la sous-station d'usinage

Le Grafcet que nous proposons se présente sous la forme de deux graphes, le premier est celui de la commande (marche, arrête, reset) et le deuxième représente le traitement des pièces. Dans la première étape figure 3.38, nous désactivons toutes les étapes (S1 OFF|S-ALL), le voyant Reset s'allume et nous faisons une déclaration du compteur. La figure 3.38 montre une partie du Grafcet GS.

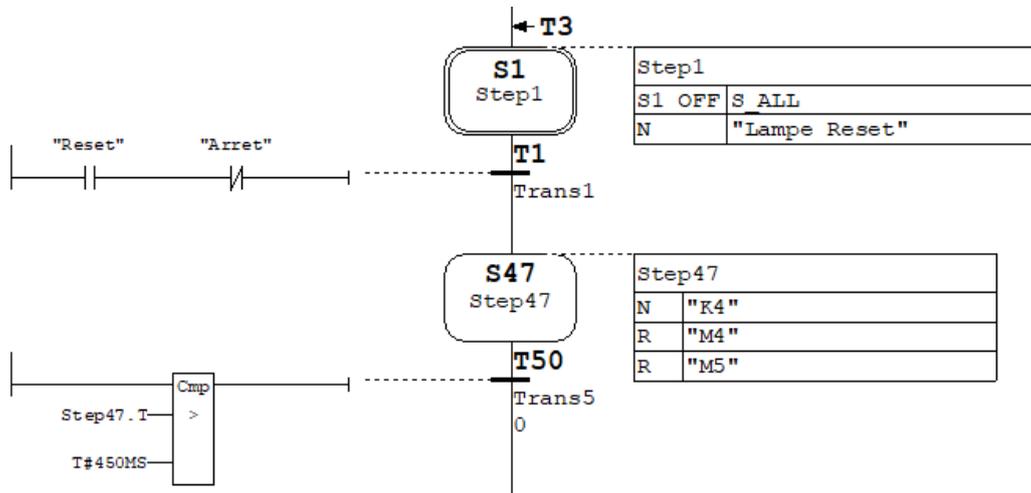


FIG. 3.38 : Une partie GS de la sous-station d'usinage.

Lorsque la pièce à usiner est présente dans l'unité de contrôle, le plateau d'indexation s'arrête, l'induit de la bobine de contrôle descend "M5" et vérifie si la pièce est posée avec ouverture vers le haut par le capteur "B4" pendant un temps spécifié (2.300 sec). Le plateau à indexation rotative "M2" tourne de 60° si le résultat du contrôle est bon.

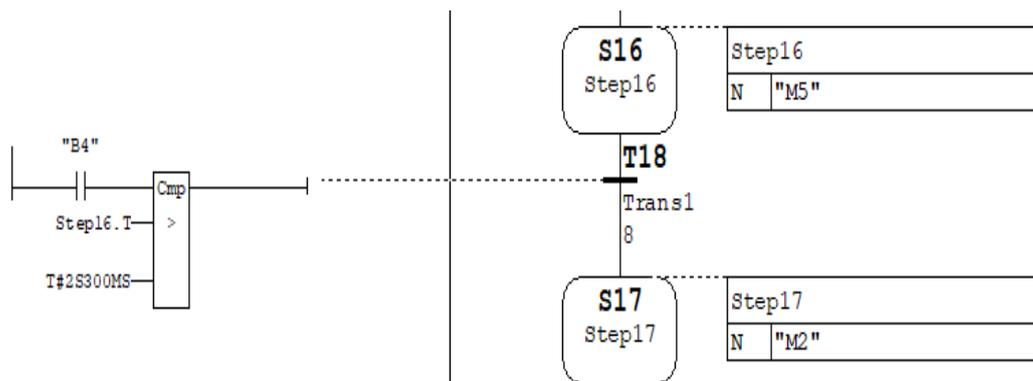


FIG. 3.39 : Une partie GPN de la sous-station d'usinage.

3.4.3 Sous-station de manipulation

3.4.3.1 Déclaration des variables de la sous-station de manipulation

Chapitre 3. La programmation, simulation et l'injection au niveau des sous-stations (MPS-variant)

	Etat	Mnémonique /	Opérande	Type de do	Commentaire
1		1B1	E 0.1	BOOL	Capteur électromagnétique "Manipulation à la station en amont"
2		1B2	E 0.2	BOOL	Capteur électromagnétique "Manipulation à la station aval"
3		1M1	A 0.0	BOOL	Actionneur pneumatique "Manipulation vers la station en amont"
4		1M2	A 0.1	BOOL	Actionneur pneumatique "Manipulation vers la station en aval"
5		2B1	E 0.4	BOOL	Capteur électromagnétique "Pince sortie"
6		2B2	E 0.5	BOOL	Capteur électromagnétique "Pince rentrée"
7		2M1	A 0.2	BOOL	Actionneur pneumatique "Vérin de levage descend"
8		3B1	E 0.6	BOOL	Capteur Optique de proximité "Détecter la pièce au niveau de la pince"
9		3M1	A 0.3	BOOL	Actionneur pneumatique "La pince s'ouvre"
10		Arret	E 1.1	BOOL	Bouton d'arrêt (normalement fermé)
11		Auto/Manu	E 1.2	BOOL	Le system fonction soit en mode automatique/manuel
12		G7_STD_3	FC 72	FC 72	
13		Lampe Reset	A 1.1	BOOL	Voyant lumineux de réinitialisation
14		Lampe Start	A 1.0	BOOL	Voyant lumineux de démarrage
15		Reset	E 1.3	BOOL	Bouton de réinitialisation
16		Start	E 1.0	BOOL	Bouton de démarrage
17		TIME_TCK	SFC 64	SFC 64	Read the System Time

FIG. 3.40 : Table des variables la sous-station de manipulation

3.4.3.2 Création du Grafcet de la sous-station de manipulation

Le Grafcet que nous avons proposé se présente sous forme de deux graphes, le premier est celui de la commande (marche, arrête, reset) et le deuxième représente le transport des pièces entre deux stations. Dans l'étape montrée à la figure 3.41, le voyant Start s'allume, si nous appuyons sur le bouton Start, le système va démarrer et le fonctionnement sera soit en mode automatique (Auto) ou bien Manuel (Man) tout dépend du choix du mode. La figure 3.41 montre une partie du Grafcet GS.

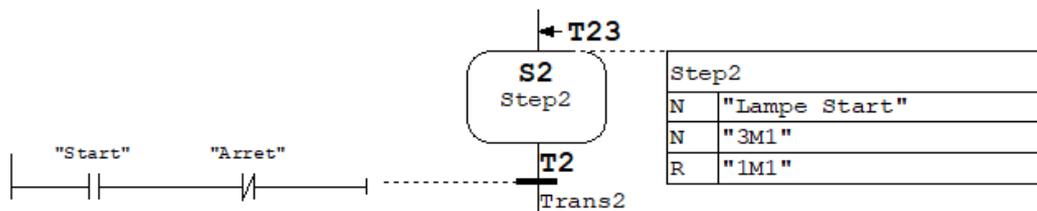


FIG. 3.41 : Une partie GS de la sous-station de manipulation.

Lorsque la pince s'ouvre, elle libère la pièce dans la station aval (livraison), le robot cartésien retourne à la station d'usinage (amont) "1M1" et le cycle se répète.

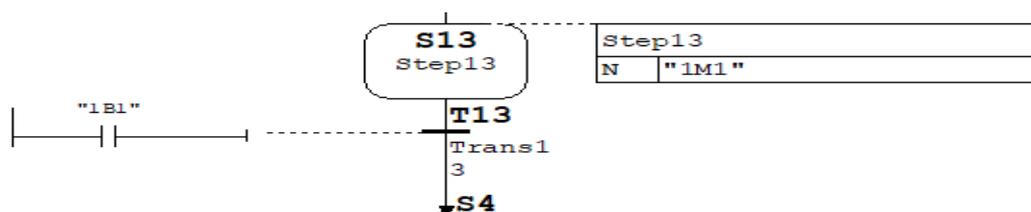
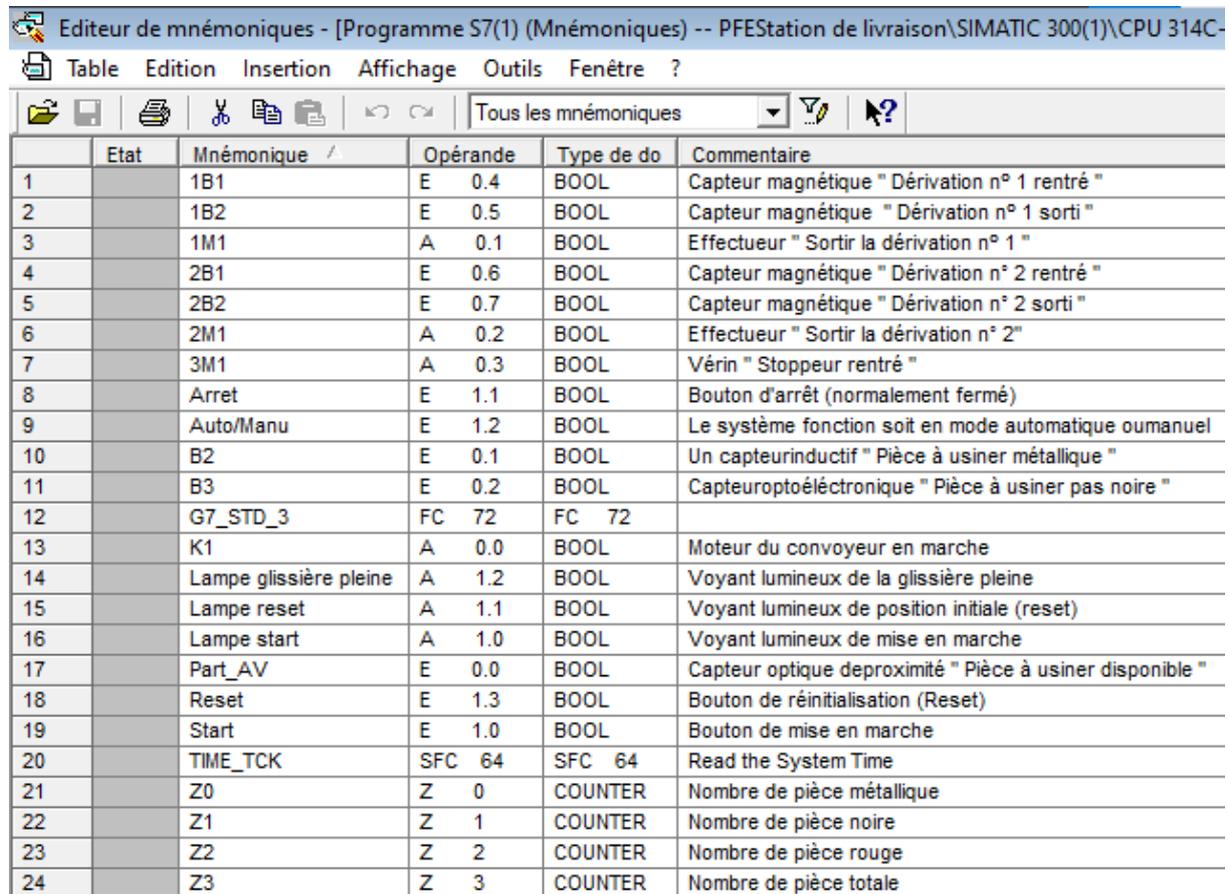


FIG. 3.42 : Une partie GPN de la sous-station Manipulation.

3.4.4 Sous-station de livraison (tri)

3.4.4.1 Déclaration des variables de la sous-station de livraison



	Etat	Mnémonique /	Opérande	Type de do	Commentaire
1		1B1	E 0.4	BOOL	Capteur magnétique " Dérivation n° 1 rentré "
2		1B2	E 0.5	BOOL	Capteur magnétique " Dérivation n° 1 sorti "
3		1M1	A 0.1	BOOL	Effecteur " Sortir la dérivation n° 1 "
4		2B1	E 0.6	BOOL	Capteur magnétique " Dérivation n° 2 rentré "
5		2B2	E 0.7	BOOL	Capteur magnétique " Dérivation n° 2 sorti "
6		2M1	A 0.2	BOOL	Effecteur " Sortir la dérivation n° 2 "
7		3M1	A 0.3	BOOL	Vérin " Stoppeur rentré "
8		Arret	E 1.1	BOOL	Bouton d'arrêt (normalement fermé)
9		Auto/Manu	E 1.2	BOOL	Le système fonction soit en mode automatique ou manuel
10		B2	E 0.1	BOOL	Un capteur inductif " Pièce à usiner métallique "
11		B3	E 0.2	BOOL	Capteur optoélectronique " Pièce à usiner pas noire "
12		G7_STD_3	FC 72	FC 72	
13		K1	A 0.0	BOOL	Moteur du convoyeur en marche
14		Lampe glissière pleine	A 1.2	BOOL	Voyant lumineux de la glissière pleine
15		Lampe reset	A 1.1	BOOL	Voyant lumineux de position initiale (reset)
16		Lampe start	A 1.0	BOOL	Voyant lumineux de mise en marche
17		Part_AV	E 0.0	BOOL	Capteur optique de proximité " Pièce à usiner disponible "
18		Reset	E 1.3	BOOL	Bouton de réinitialisation (Reset)
19		Start	E 1.0	BOOL	Bouton de mise en marche
20		TIME_TCK	SFC 64	SFC 64	Read the System Time
21		Z0	Z 0	COUNTER	Nombre de pièce métallique
22		Z1	Z 1	COUNTER	Nombre de pièce noire
23		Z2	Z 2	COUNTER	Nombre de pièce rouge
24		Z3	Z 3	COUNTER	Nombre de pièce totale

FIG. 3.43 : Table des variables la sous-station de livraison.

3.4.4.2 Création du Grafcet de la sous-station de livraison

Le grafcet que nous avons proposé est composé de deux graphes comme les sous-stations précédents, le premier est dédié à la commande (start, stop, reset) et le second représente le fonctionnement normal du tri des pièces. La figure 3.44 montre une partie GS de la station de tri, dans la première étape, nous désactivons toutes les étapes (S1 OFF|S-ALL), nous faisons une déclaration du voyant Reset avec une réinitialisation des différents compteurs (Z0, Z1, Z2, Z3) utilisés dans ce système.

Si nous cliquons sur le bouton Reset, la transition 1 (T1) est franchi, donc l'étape 1 est désactivée et la deuxième étape est active.

Dans la deuxième étape, nous remettons à zéro les trois actions (1M1, 2M2, K1).

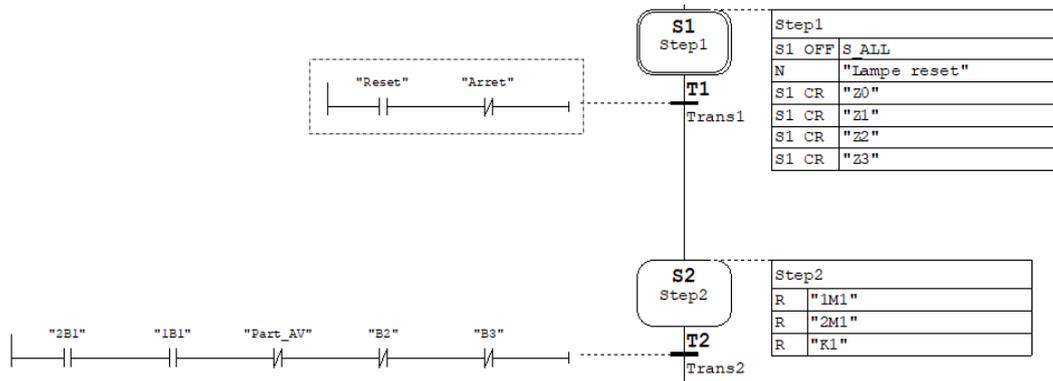


FIG. 3.44 : Une partie GS de la sous-station de tri.

Dans l'étape 5, le compteur Z3 s'incrémente, après cette étape, nous avons un OU divergent qui se compose de quatre séquences une séquence pour le passage des pièces noir, un deuxième pour les pièces rouges un troisième pour les pièces métalliques et un quatrième au cas où toutes les glissières seront pleines (18 pièces). Dans la figure 3.45, lorsque nous sommes en mode automatique, le compteur va compter 18 pièces et une fois que la 19ème pièce est détectée par le capteur Part-AV, les capteurs 1B1 et 2B1 se mettent à 1, la transition T20 est franchi, l'étape 15 s'active et la lampe glissière pleine s'allume, alors toute la station s'arrête pendant 1min et 2sec pour effectuer le déchargement des glissières.

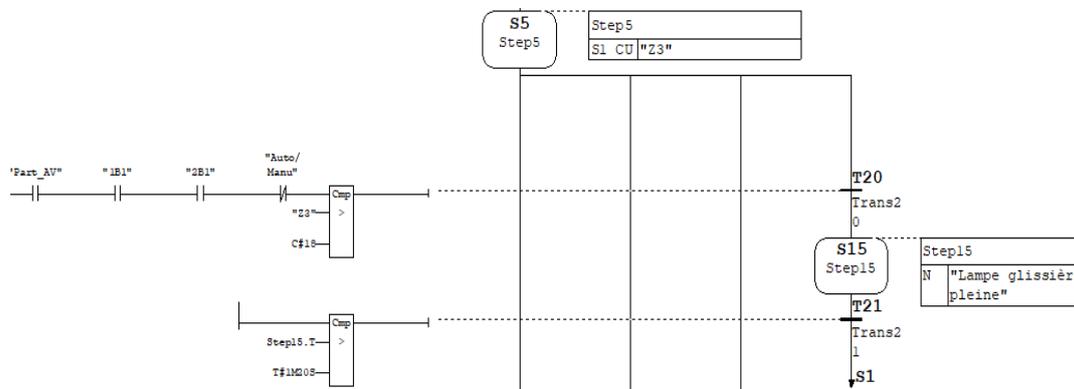


FIG. 3.45 : Une partie GPN de la sous-station de tri des pièces (a).

Dans la figure 3.46 lorsque nous serons en mode automatique et que le compteur compte moins de 18 pièces, si le capteur Part-AV détecter une pièce, 1B1 et 2B1 se mettent à 1, le moteur du convoyeur se mis en marche.

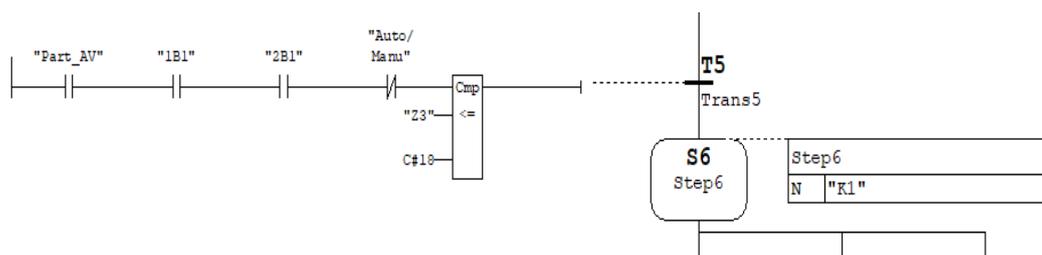


FIG. 3.46 : Une partie GPN de la sous-station tri des pièces (b).

La figure 3.47 montre le cas du tri d'une pièce métallique, d'abord le compteur Z0 s'incrémente, la pièce métallique sera détectée à l'aide des deux capteurs "B2" et "B3", cette pièce est arrêtée par le stoppeur 3M1 qui laisse suffisamment de temps aux deux capteurs pour détecter la pièce. Selon notre cahier des charges la pièce sera glissée dans la glissière C1 (glissière au début de la bande N°01). Par conséquent, quand le stoppeur laisse passer la pièce, celle-ci est transférée sur le convoyeur pendant un temps déterminé jusqu'à ce que la dérivation 1 "1M1" fasse glisser la pièce vers C1.

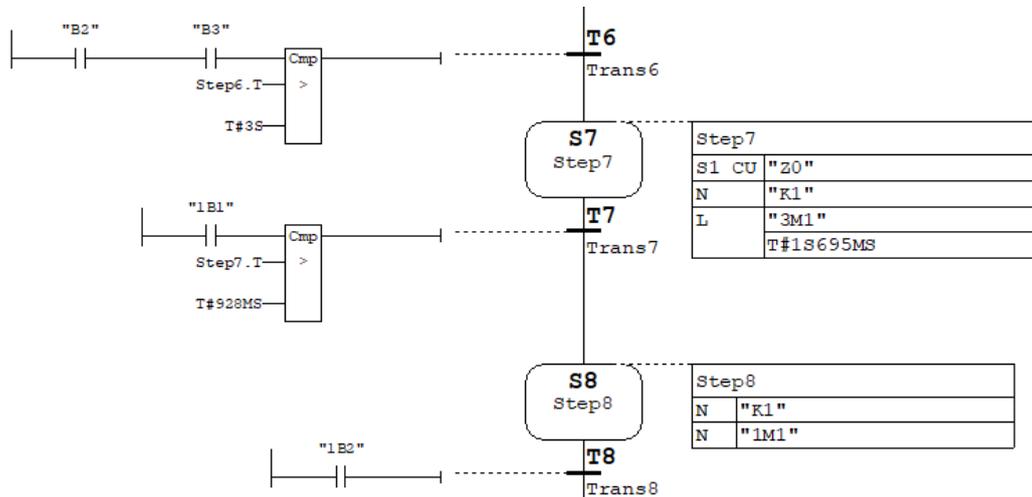


FIG. 3.47 : Une partie GPN de la sous-station tri des pièces (c).

3.5 Simulation de notre MPS variant avec S7-PLCSIM

3.5.1 Présentation du S7-PLCSIM

Le module de simulation S7-PLCSIM est une application qui permet d'exécuter et de tester le programme dans un automate programmable virtuel (API virtuel)[70]. La simulation s'effectuant entièrement dans le logiciel STEP7. Il offre une interface simple pour visualiser et modifier divers paramètres tels que les variables d'entrée et de sorties. La barre d'outils du SIMATIC Projet Manager comporte un bouton permettant d'activer ou de désactiver la simulation. Quand nous cliquons sur ce bouton , le logiciel S7-PLCSIM apparaît avec son CPU de simulation.

- En sélectionnant la case placée devant RUN-P , vous pouvez lancer l'API simulé dans PLCSIM.
- Les entrées ou les sorties qui s'affichent peuvent être mise à 1 en sélectionnant la case sinon elles sont mises à 0.

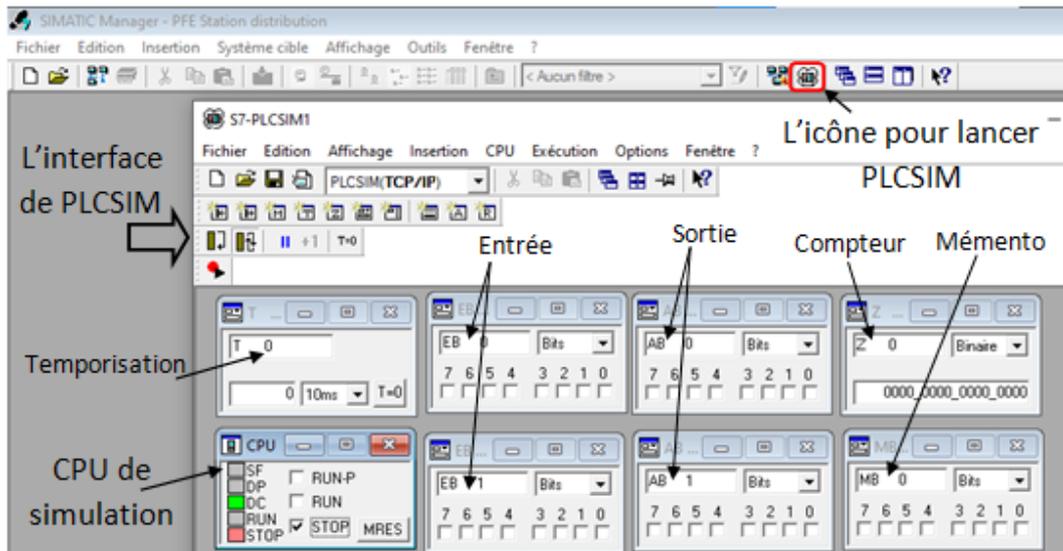


FIG. 3.48 : L'interface du simulateur S7-PLCSIM.

1. La partie programmation doit d'abord être chargée par la sélection de tous les blocs et de toutes les données du système qui se trouvent dans le programme.
2. Le chargement de tous les Blocs se fait en cliquant sur l'icône " charger " qui se trouve dans la barre d'outils.
3. Nous passons alors au chargement de la partie fonctionnelle et à la visualisation du programme en cliquant sur le bloc FB1. On charge notre programme.
4. Enfin, nous cliquons sur le bouton de visualisation (icône sous forme de lunettes) pour visualiser la simulation du programme.

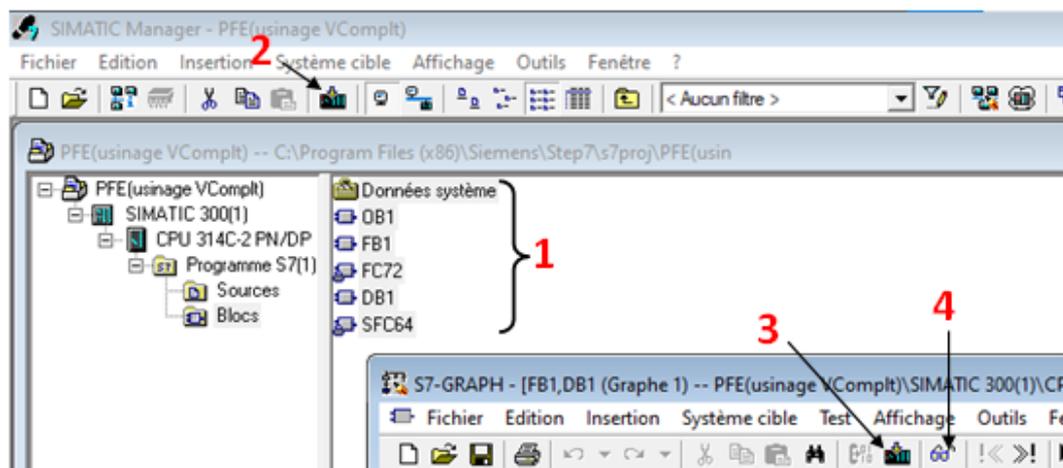


FIG. 3.49 : Chargement du programme et lancement de la visualisation.

3.5.2 Simulation de la sous-station de distribution

La figure 3.50 illustre une partie du graph 1 GS de la commande de la sous-station de distribution.

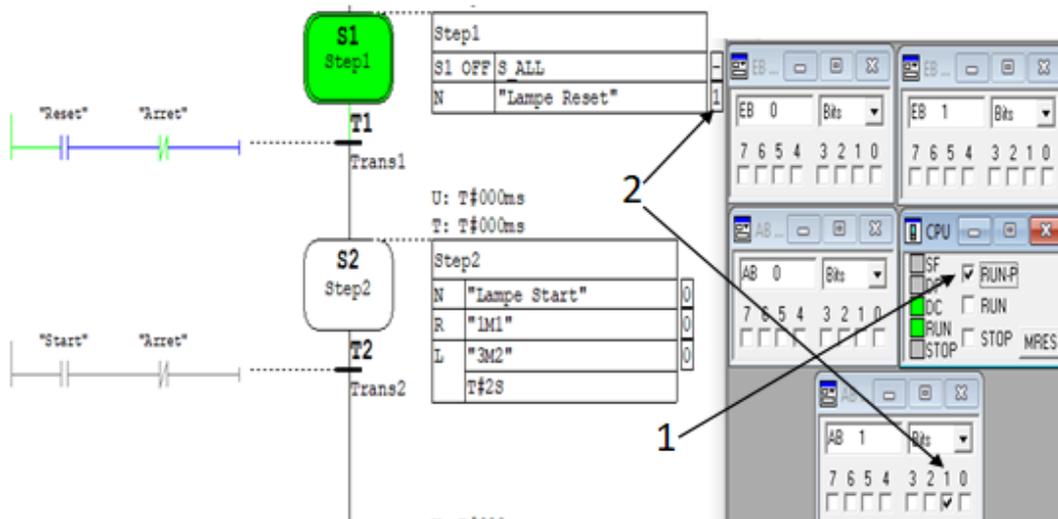


FIG. 3.50 : Simulation de la sous-station de distribution partie GS (a).

1. Lorsque nous passons en mode RUN-P, la première étape (S1 ou Step1) est activée (colorée en vert) automatiquement et nous remarquons que toutes les autres étapes sont désactivées.
2. Les actions de la première étape sont exécutées (voyant Reset s'allume "A 1.1") voir la figure 3.50.

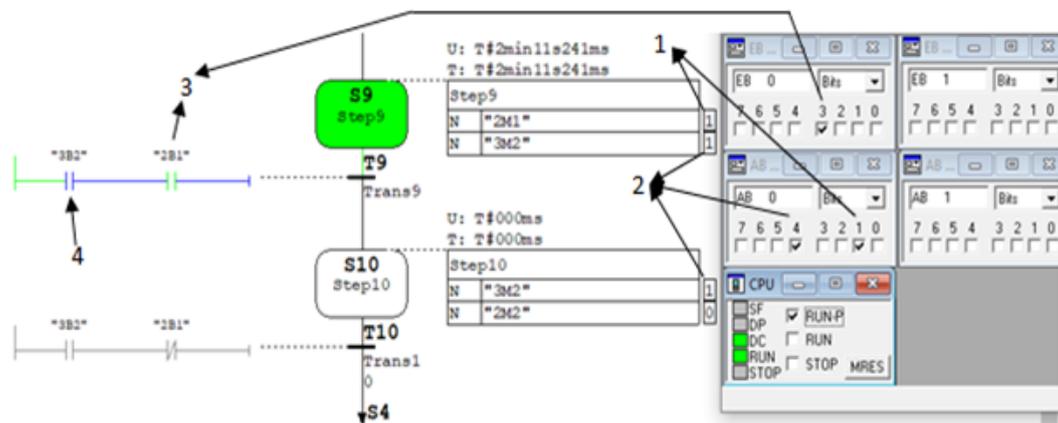


FIG. 3.51 : Simulation de la sous-station de distribution partie GPN.

La figure 3.51 illustre une partie du graph 2 GPN de la distribution des pièces, nous remarquons que l'étape 9 s'active et que ses actions s'exécutent.

1. Le vérin (2M1 "A 0.1"), permet l'aspiration de la pièce.
2. Le vérin (3M2 "A 0.4") déplace la pièce à usiner vers la station suivante "d'usinage".
3. Le capteur (2B1 "E 0.3") indique que la pièce à usiner est bien aspiré.
4. Si le capteur (3B2 "E 0.5") détecte que la pièce est en position station suivante "usinage", il permettra le passage à l'étape suivante qui est l'éjection de la pièce par (2M2 "A 0.2").

Chapitre 3. La programmation, simulation et l'injection au niveau des sous-stations (MPS-variant)

Dans le graphe 1 GS, nous insérons une étape pour arrêter directement le système en cas de problème ou de défaut en cliquant sur le bouton d'arrêt (E 1.1).

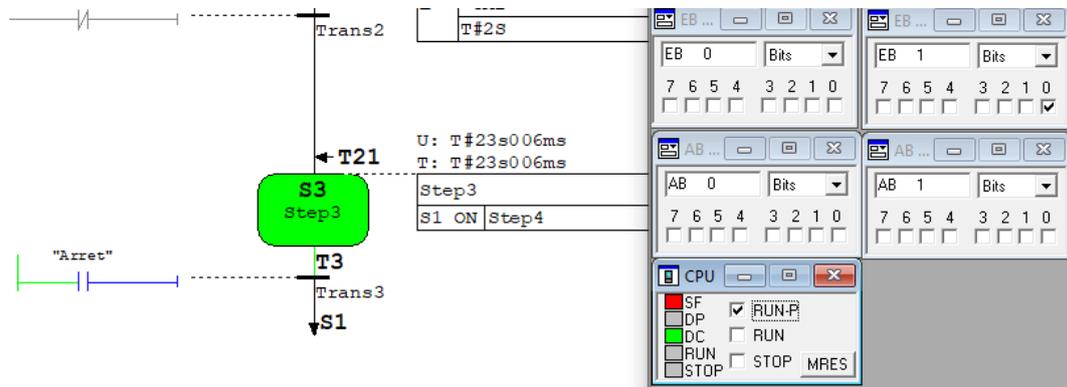


FIG. 3.52 : Simulation de la sous-station de distribution partie GS (b).

3.5.3 Simulation de la sous-station d'usinage

La figure 3.53 illustre une partie du graph 1 GS de la commande de la station d'usinage.

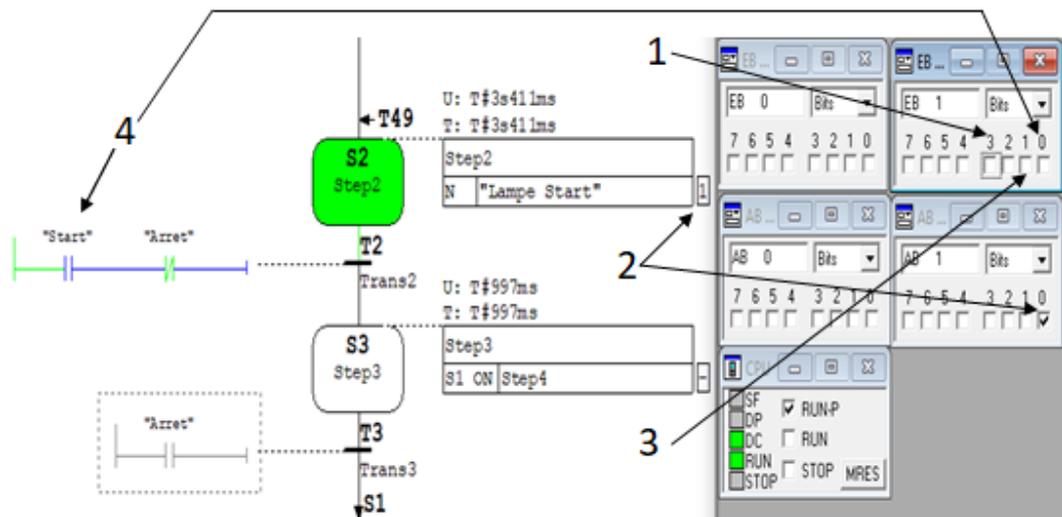


FIG. 3.53 : Simulation de la sous-station d'usinage partie GS.

1. Quand on clique sur le bouton de Reset "E 1.3", la première transition est franchie et l'étape S2 est activée, comme le montre la figure 3.53.
2. Les actions de S2 sont exécutées (voyant Start s'allume "A 1.1") comme l'indique la figure 3.53.
3. Si nous cliquons sur le bouton d'arrêt "E 1.1", le fonctionnement du système s'arrête.
4. Si nous cliquons sur le bouton Start "E 1.0", T2 devient franchi et S2 est désactivée, S3 est activée et ses actions sont exécutées, c'est-à-dire que la simulation passera au deuxième graph (Le traitement des pièces).

Chapitre 3. La programmation, simulation et l'injection au niveau des sous-stations (MPS-variant)

La figure 3.54 montre une partie du graph 2 GPN du processus de traitement des pièces. Après avoir cliqué sur le bouton Start, l'étape S4 sera activée.

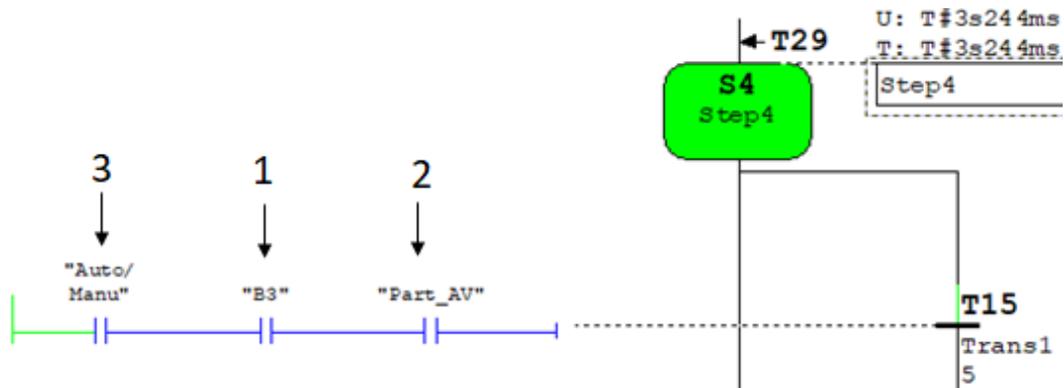


FIG. 3.54 : Simulation de la sous-station d'usinage partie GPN (a).

1. Si la table d'indexation est positionnée (B3 "E 0.5") c'est-à-dire que la condition est vraie.
2. Si la pièce est disponible (Part-AV "E 0.0") c'est-à-dire que la condition est vraie.
3. Si nous sélectionnons l'adresse de la touche (Auto/Manu "E 1.2"), c'est-à-dire que le système travaille en mode manuel. (La non-sélection de l'adresse de la variable Auto/Manu "E 1.2", veut dire que le système fonctionne en mode automatique).

Si les trois conditions précédentes sont vraies, la transition T15 est franchie et l'étape suivante S14 est activée et ses actions exécutées, ce qui signifie que le moteur de la table d'indexation (M2 "A 0.1") est mis en marche et que la table sera tournée pendant un temps déterminé (1.154 sec) jusqu'à ce qu'elle atteigne la position de contrôle. Ceci est illustré à la figure 3.55.

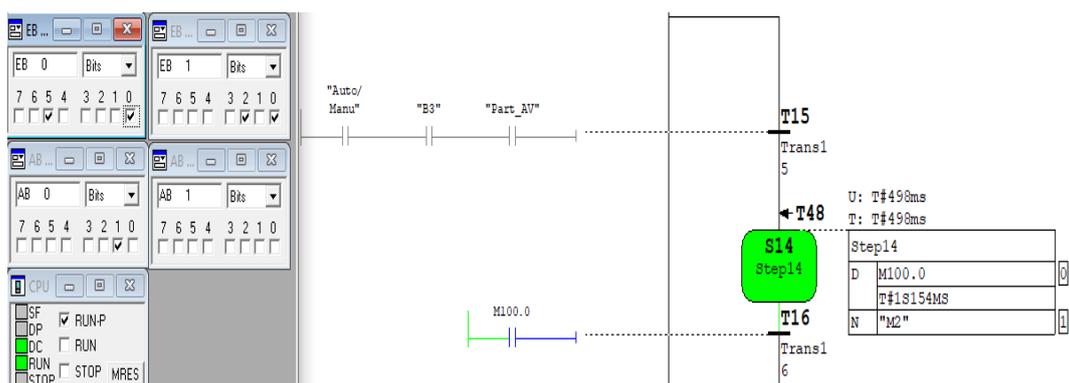


FIG. 3.55 : Simulation de la sous-station d'usinage partie GPN (b).

3.5.4 Simulation de la sous-station de manipulation

La figure 3.56 montre une partie du graph 2 GPN de la station de manipulation. Si nous ne sélectionnons pas l'adresse du mode (Auto/Manu "E 1.2"), le système fonctionnera

Chapitre 3. La programmation, simulation et l'injection au niveau des sous-stations (MPS-variant)

en mode automatique. Lorsque le manipulateur est en position amont indiquée par son capteur (1B1 "E 0.1"), nous voyons également que la pince est en position rentrée indiquée par son capteur (2B2 "E 0.5"), l'étape S5 s'active et ses actions s'exécutent, d'abord la pince s'ouvre (3M1 "A 0.3"), puis la pince sortira "vers le bas" (2M1 "A 0.2") jusqu'à ce que le capteur électromagnétique (2B1 "E 0.4") détecte que la pince est en position basse.

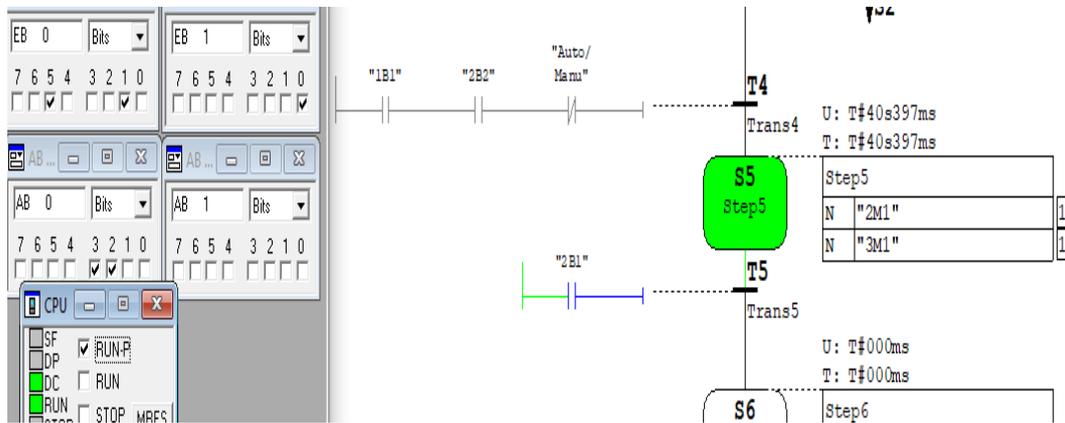


FIG. 3.56 : Simulation de la sous-station de manipulation partie GPN (a).

Lorsque l'étape S12 est activée, la pince s'ouvre (3M1 "A 0.3") et libère la pièce dans la station aval, si la transition T12 est franchie (c'est-à-dire que si les conditions (1B2 "E 0.2") et (3B1 "E 0.6") sont vraies), l'étape S13 est activée et le manipulateur retourne à la station amont (1M1 "A 0.0") et le cycle se répète.

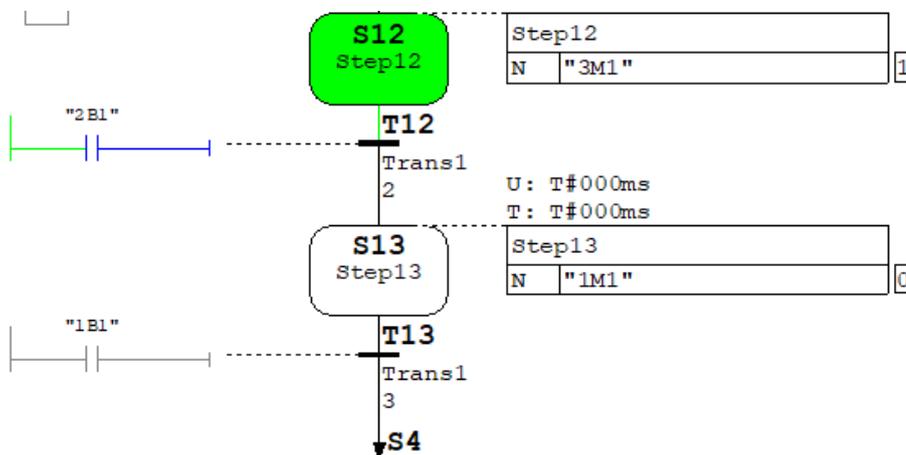


FIG. 3.57 : Simulation de la sous-station de manipulation partie GPN (b).

3.5.5 Simulation de la sous-station de livraison (tri)

La figure 3.58 illustre une partie du graph 2 GPN de la station de tri. En appuyant sur le bouton Start "E 1.0", l'étape 5 du graph 2 est activée, le compteur Z3 commence à compter les pièces.

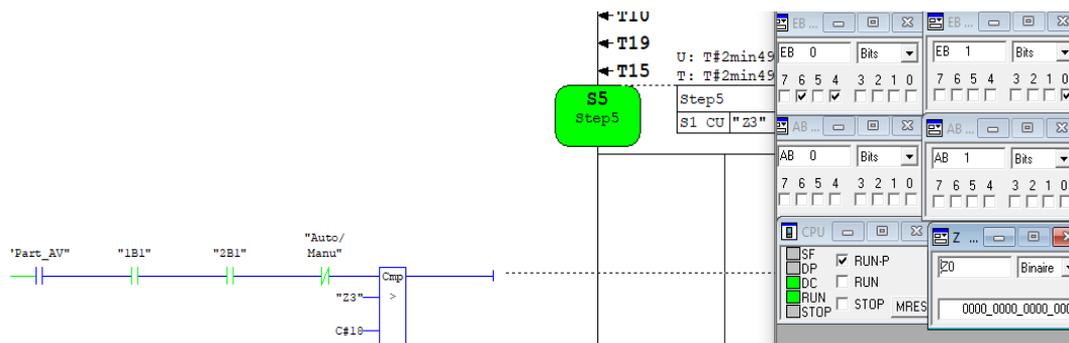


FIG. 3.58 : Simulation de la sous-station de tri partie GPN (a).

Si le compteur atteint 18 pièces alors les trois glissières sont pleines et une fois que la 19ème pièce est détectée par le capteur Part-AV, la lampe glissière pleine s'allume pendant un temps pour effectuer le déchargement des glissières. Si le compteur est inférieur à 18, les autres pièces sont triées selon leur type.

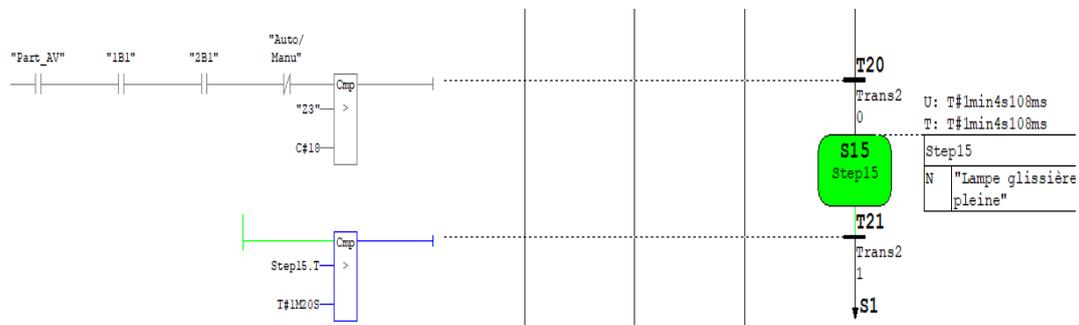


FIG. 3.59 : Simulation de la sous-station de tri partie GPN (b).

La figure 3.60 montre une partie du graphe 2 GPN de la station de tri, nous expliquons dans ce qui suit le cas où une pièce métallique est détectée. Lorsque la pièce est arrêtée par le stoppeur pendant un temps d'attente, le capteur B2 (capteur de métal) détecte que la pièce bloquée par le stoppeur est une pièce métallique, alors la pièce sera transférée par le convoyeur (la durée du transfert est bien déterminée), jusqu'à ce que la dérivation 1 "1M1" fasse glisser la pièce vers la glissière C1.

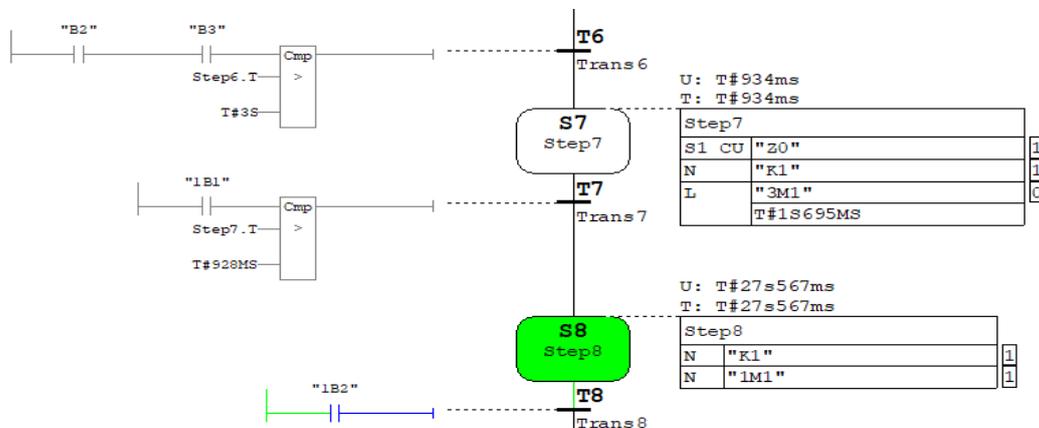


FIG. 3.60 : Simulation de la sous-station de tri partie GPN (c).

3.6 Test des sous-stations réelles via l'outil L'EASY-PORT USB

3.6.1 EASYPORT USB

L'interface de processus EasyPort USB est utilisée pour la transmission bidirectionnelle de signaux de processus entre un processus de contrôle réel qui utilise une technologie basse tension (24 V DC) et un PC. Le fonctionnement d'un EasyPort USB est simple : l'interface USB est connectée au PC. La connexion à l'équipement d'automatisation se fait via des connecteurs standards SysLink. Les signaux d'entrée et de sortie sont ainsi accessibles à partir d'un PC[71].

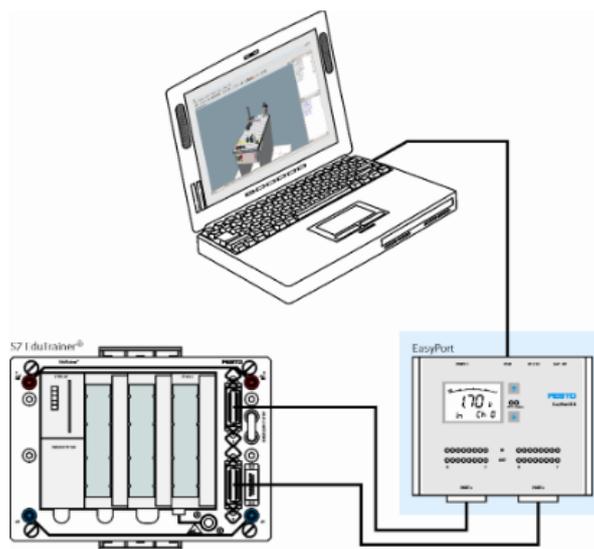


FIG. 3.61 : Câblage d'un processus réel et un PC à l'aide d'un EasyPort USB[71].

- 1/ Connecteur Sub D femelle.
- 2/ Port USB.
- 3/ Port RS 232.
- 4/ Bornes à vis isolées pour l'alimentation.
- 5/ $\uparrow\downarrow$ Boutons.
- 6/ LED de statut (vert).
- 7/ LED d'erreur (rouge).
- 8/ Interfaces SysLink (ports 1 et 2).
- 9/ LEDs d'indication de statut des sorties numériques (orange).
- 10/ LEDs d'indication de statut des entrées numériques (vert).
- 11/ Panneau LCD pour les signaux analogiques et les adresses EasyPort.

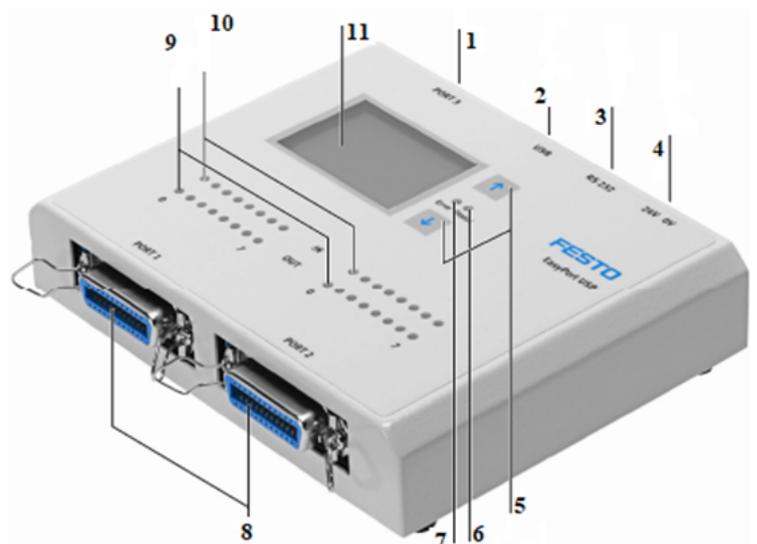


FIG. 3.62 : Identification d'EasyPort USB[71].

3.6.2 EzOPC

Il est facile d'établir la communication entre EasyPort USB et autres équipements Festo Didactic. Il suffit pour cela d'utiliser le serveur EzOPC qui est un programme doté d'une interface utilisateur graphique. Pour configurer EzOPC, il faut sélectionner les partenaires de communication. Ensuite, spécifier la gamme d'E/S dont nous souhaitons échanger les signaux.[71].

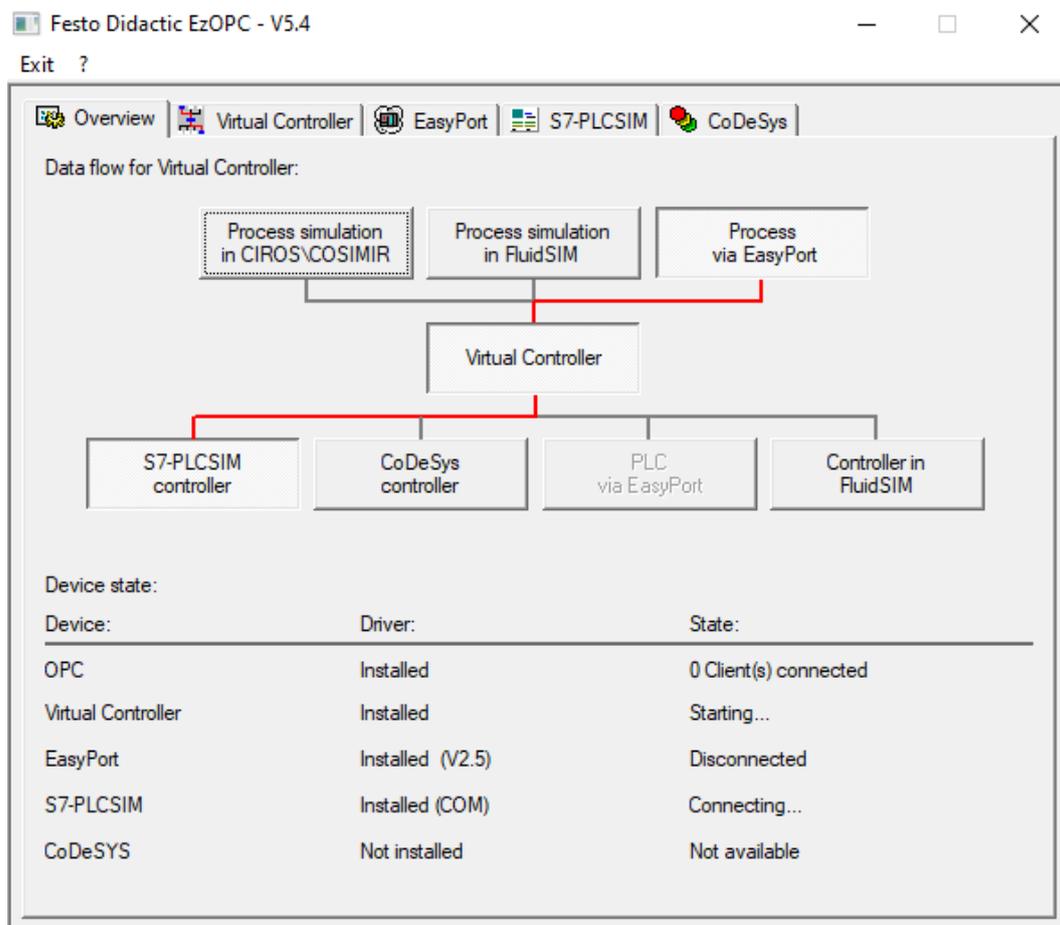


FIG. 3.63 : Mode de connexion de l'interface EzOPC.

3.6.3 Branchement de la Sous-Station réelle avec EASYPORT USB

Le fonctionnement de ce dispositif est assez simple : il suffit de connecter l'interface USB au PC. Le raccordement à l'équipement d'automatisation se fait par des connecteurs SysLink pour isoler l'automate programmable réel. Les signaux d'entrée et de sortie sont donc lisibles à partir du PC (Simulateur S7-PLCSIM) et de l'EasyPort. Tout d'abord, nous séparons les câbles SysLink (processus et pupitre de commande) de l'automate et nous les introduisons respectivement dans les ports 1 et 2 de l'EasyPort, ensuite, nous connectons l'interface USB au PC via un câble USB. Enfin, nous mettons l'EasyPort sous tension (24V DC). Le branchement d'EasyPort entre la station et notre PC est illustré dans la figure 3.64.

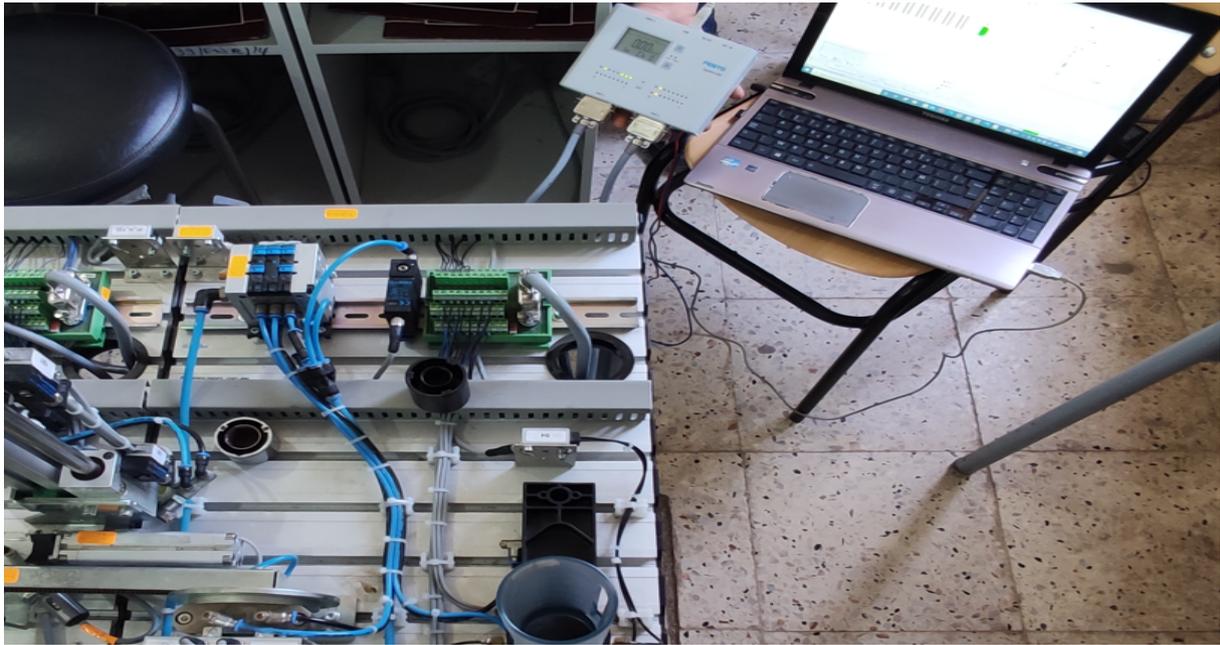


FIG. 3.64 : Branchement de la sous-station de distribution via EasyPort USB.

Après le branchement de l'EasyPort, nous ouvrons la fenêtre d'EzOPC pour sélectionner les partenaires de communication, afin de lier la connexion entre S7-PLCSIM, l'interface EasyPort USB et le processus. Ensuite, nous spécifions la gamme d'E/S dont nous souhaitons échanger les signaux. Nous constatons à partir de la figure 3.65 que les quatre dispositifs (OPC, Virtual Controller, EasyPort, S7-PLCSIM) sont connectés, et que la liaison entre S7-PLCSIM, l'interface USB EasyPort et le processus a été bien établi.

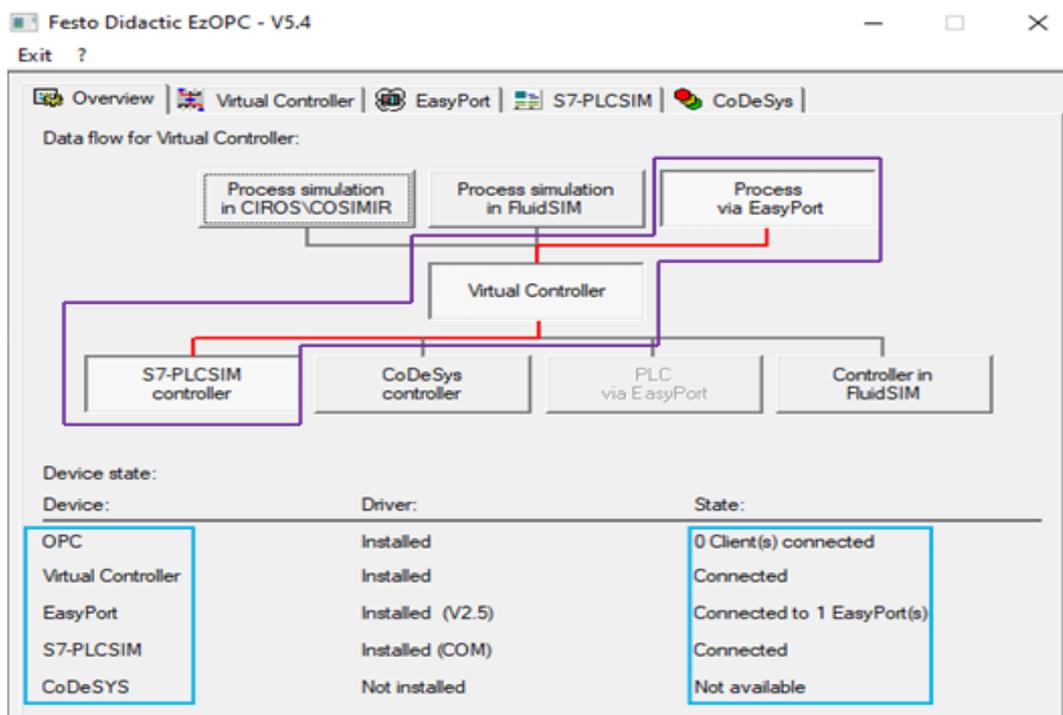


FIG. 3.65 : Choix des partenaires de communication dans l'interface EzOPC.

Chapitre 3. La programmation, simulation et l'injection au niveau des sous-stations (MPS-variant)

Nous allons maintenant passer à la visualisation en temps réel d'une des stations, qui est la station de distribution, nous allons voir le processus de synchronisation entre le Grafcet, S7-PLCSIM, les LEDs de EasyPort USB et la station réelle.

Le principe d'exécution du programme utilisant EasyPort dans chaque sous-station est le même que la sous-station de distribution.

Dans l'EasyPort, les LEDs vertes représentent les entrées et les LEDs orange les sorties, dans l'interface EzOPC, les LEDs vertes représentent les entrées et les LEDs jaunes les sorties.

Quand nous lançons la visualisation, l'étape initiale s'allume, nous verrons que la LED Reset s'allume "A 1.1" dans chaque dispositif que ce soit dans la simulation, visualisation et sous-station réelle.

Selon les figures suivantes, nous remarquons que les entrées/sorties présentent une excellente synchronisation, entre les divers dispositifs de visualisation, de simulation et le processus réel.

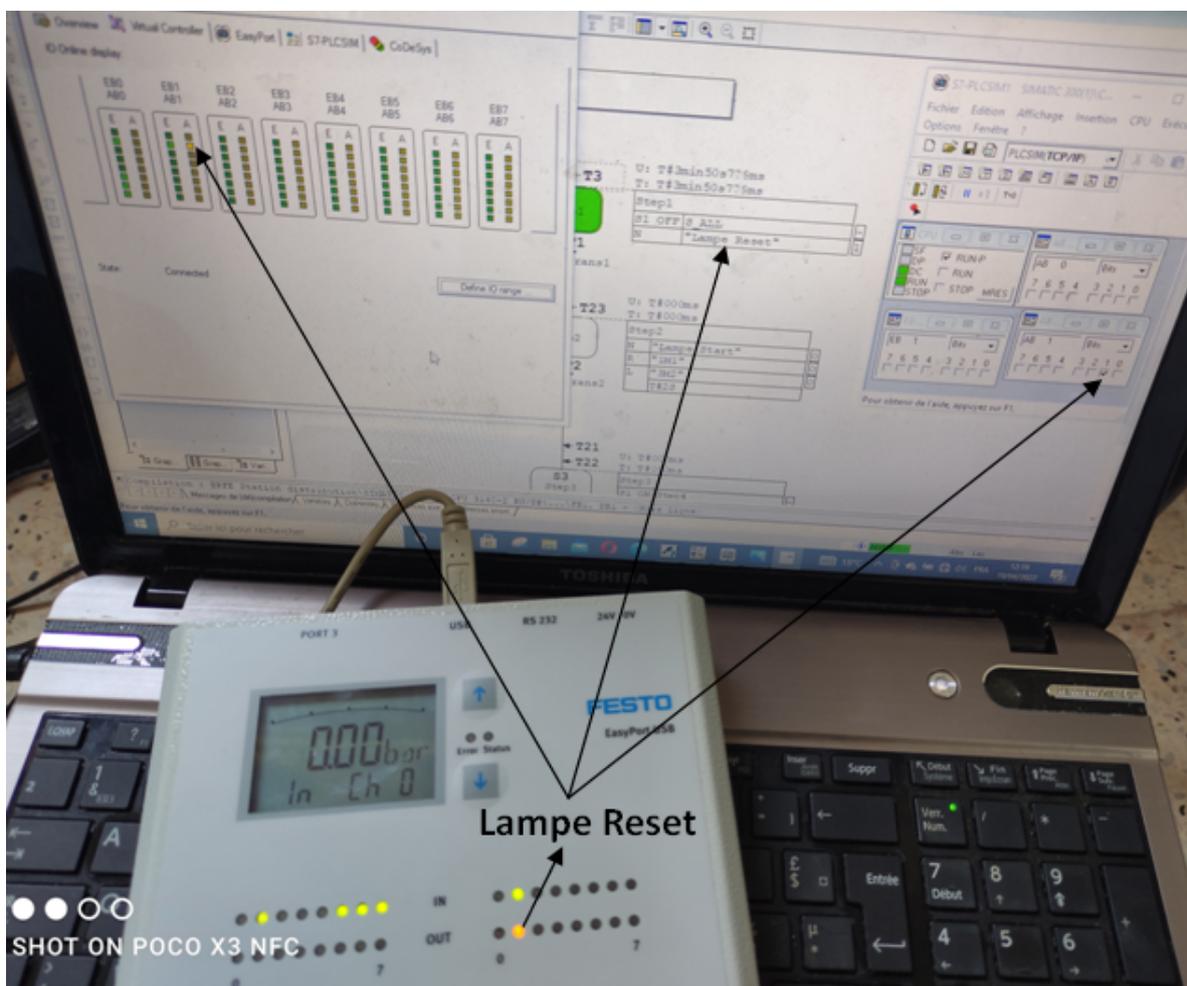


FIG. 3.66 : La synchronisation entre le Grafcet partie GS, l'interface EzOPC, PLCSIM et l'EasyPort.

Si nous mettons le bouton Reset "E 1.3" à 1, la transition T1 est franchie, donc l'étape 1 est désactivée et l'étape 2 s'active avec l'exécution de ses actions.

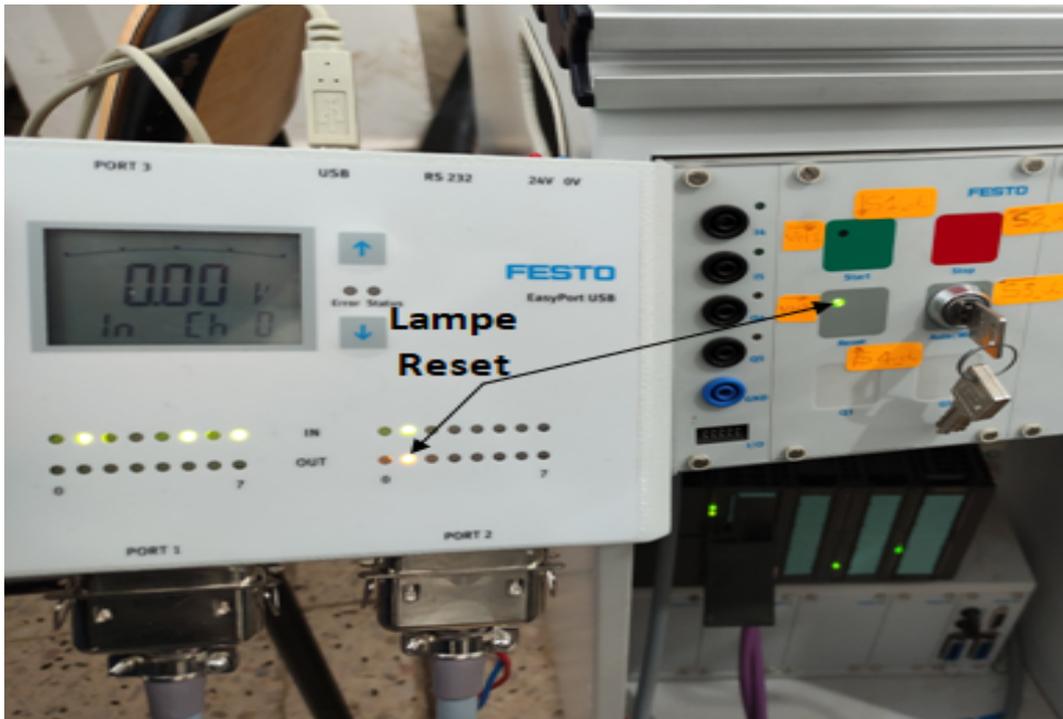


FIG. 3.67 : La synchronisation entre l'EasyPort et le pupitre de commande (a).

Nous verrons que la LED Start s'allume "A 1.0" dans chaque dispositif que ce soit dans la simulation, visualisation et notre processus réel. La position du vérin d'éjection de la pièce (1M1 "A 0.0") est réinitialisée, nous observons cette action dans le processus réel. Nous activons le vérin oscillant (3M2 "A 0.4 ") pendant un certain temps pour qu'il revienne à son état initial.

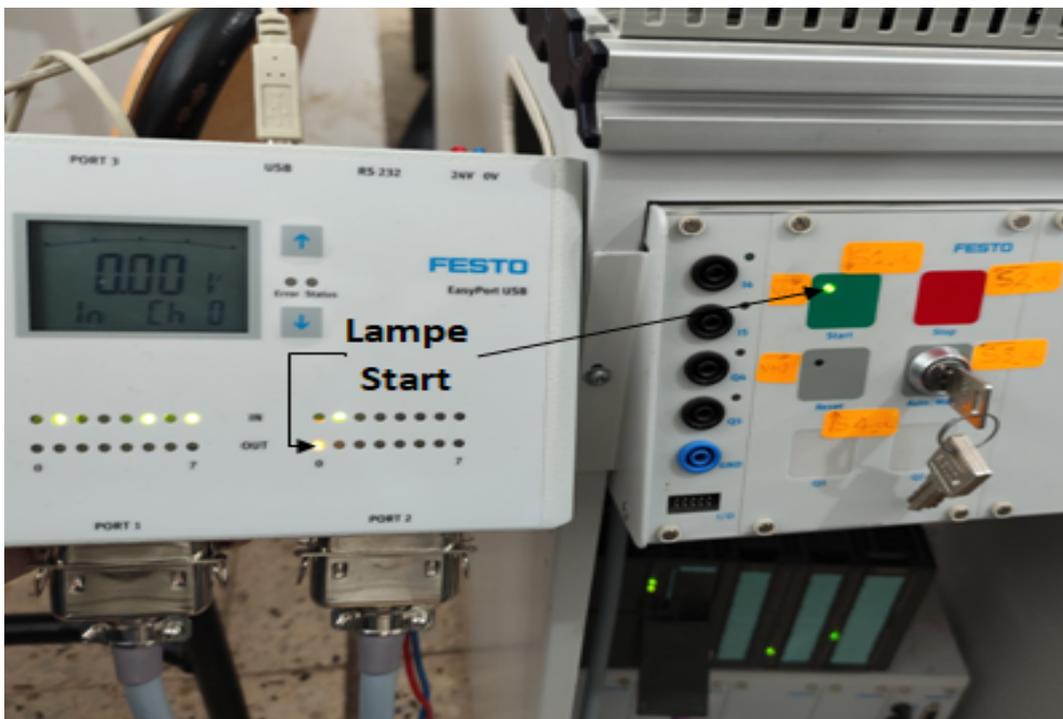


FIG. 3.68 : La synchronisation entre l'EasyPort et le pupitre de commande (b).

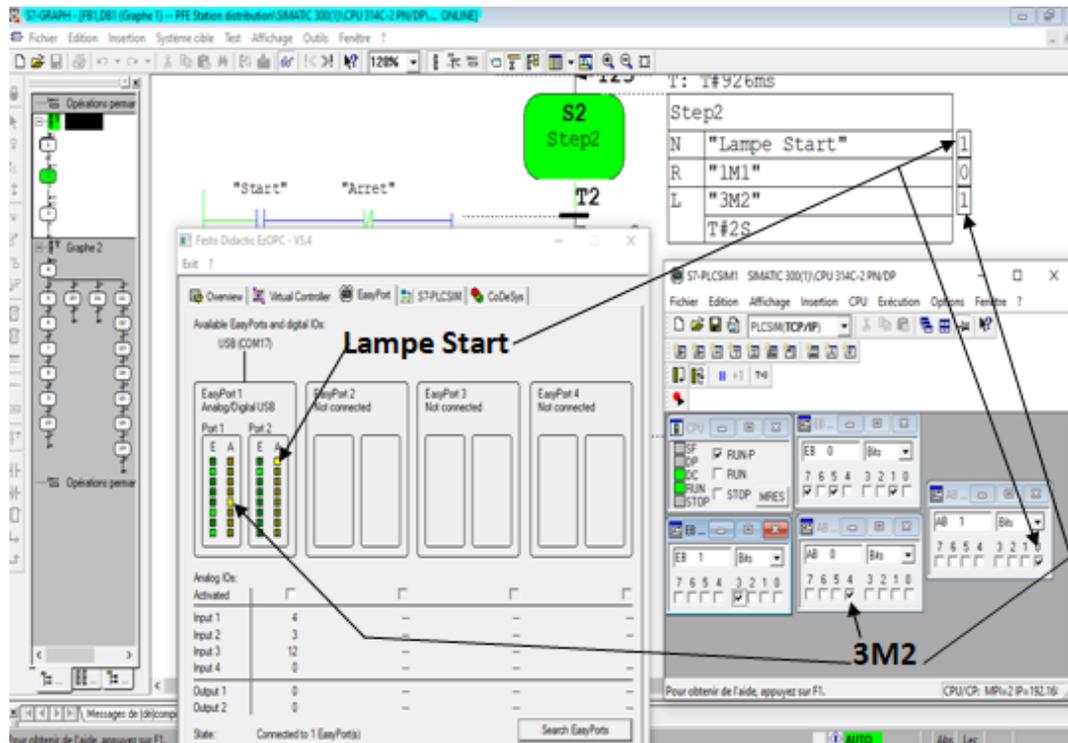


FIG. 3.69 : La synchronisation entre le Grafset partie GS, le simulateur PLCSIM et l'interface EzOPC.

D'après les figures précédentes, nous constatons que les entrées/sorties présentent une excellente synchronisation, entre les divers dispositifs de visualisation, de simulation et notre processus réel.

Si le magasin est vide, La LED correspondante s'allume (MAGASIN VIDE " A 1.2 ").



FIG. 3.70 : La synchronisation entre l'EasyPort et le pupitre de commande (c).

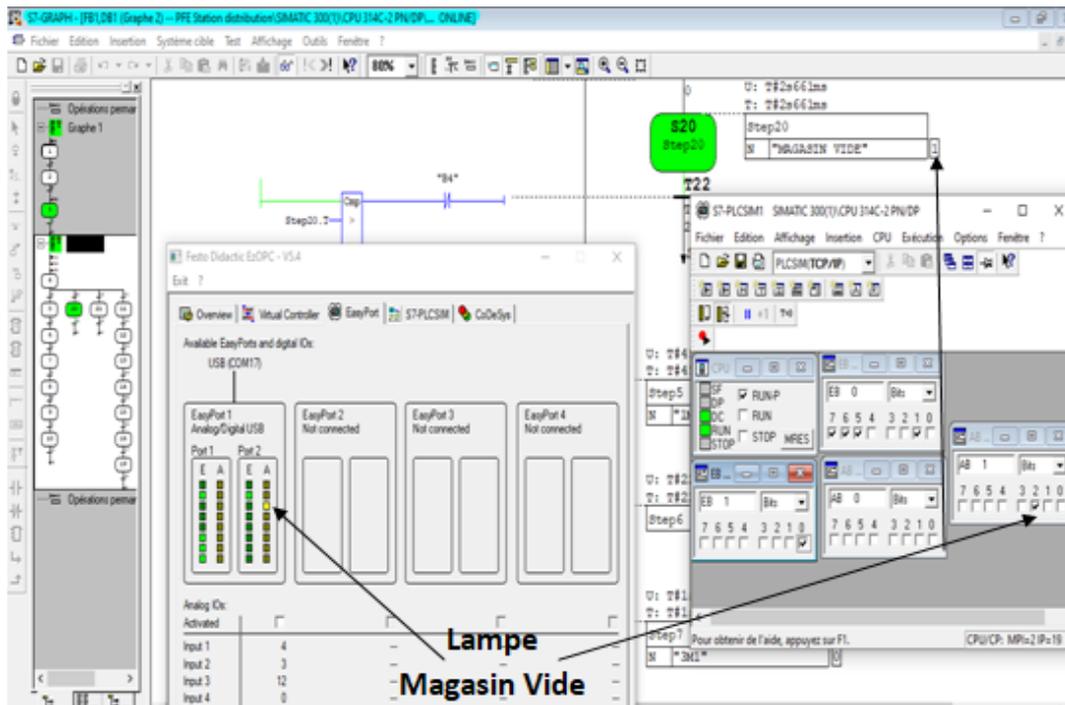


FIG. 3.71 : La synchronisation entre le Grafset partie GPN, le simulateur PLCSIM et l'interface EzOPC.

3.7 Injection des programmes dans l'API

Dans l'injection des programmes, nous allons utiliser une carte mémoire MMC de type SIEMENS, chaque station contient un API, donc nous allons insérer pour chaque API sa carte mémoire.

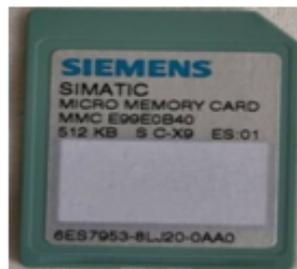


FIG. 3.72 : Carte mémoire MMC du type SIEMENS.

Pour injecter un programme dans un automate réel, nous devons suivre certaines procédures.

a. Étape N°01 : Établir une connexion entre le PC et l'automate.

La première chose il faut savoir comment établir une connexion entre le PC et l'auto-
mate, pour cela, il faut utiliser l'un des moyens de communications suivants :

- Communication MPI " Multi Point Interface "

- Communication Profinet " Ethernet ".
- Communication Profibus.

Dans notre travail, nous avons choisi d'utiliser deux types de communication pour l'injection de programmes : la communication MPI et la communication Ethernet.

- La connexion entre le PC et l'automate via la communication MPI

L'utilisation de la communication MPI "Multi Point Interface" c'est-à-dire que nous devons utiliser un adaptateur USB-MPI (tous les automates S7300-400 contiennent un port MPI).

✓ Le PC adapter ou adaptateur PC

Il s'agit d'un câble spécial destiné utilisé par les appareils Siemens. Pour qu'un PC puisse communiquer avec un automate, il est nécessaire d'utiliser un adaptateur. L'adaptateur PC permet de se connecter à l'automate, de le configurer, de le programmer, de transférer les programmes et de vérifier l'état des entrées et des sorties via le logiciel SIMATIC STEP 7. Les anciens modèles d'adaptateurs PC utilisaient l'interface RS232 pour la connexion aux PC. Aujourd'hui de nombreux PC sont équipés de ports USB et non de ports série, les nouveaux adaptateurs PC se connectent donc aux PC via USB.[72].



FIG. 3.73 : L'adaptateur USB MPI[72].

Pour que l'adaptateur soit détecté par le PC, nous devons d'abord installer son driver. Une fois que le branchement est terminé et que les deux LEDs de l'adaptateur sont verts, nous allons entamer la partie logicielle (STEP 7).



FIG. 3.74 : Branchement de l'automate avec le PC via l'adaptateur USB-MPI.

Au niveau de la partie logicielle, il s'agit d'abord de configurer les différents pilotes pour que la communication soit efficace. Pour cela nous passons à la configuration d'une interface PG/PC avec STEP 7 Siemens.

Dans la fenêtre du SIMATIC Manager, nous constatons que la communication de notre projet n'est pas compatible avec l'adaptateur PC MPI, nous devons donc la modifier. Cette communication est modifiée dans la fenêtre "Paramètres de l'interface PG/PC" du menu "Outils" de la barre des tâches du SIMATIC Manager.

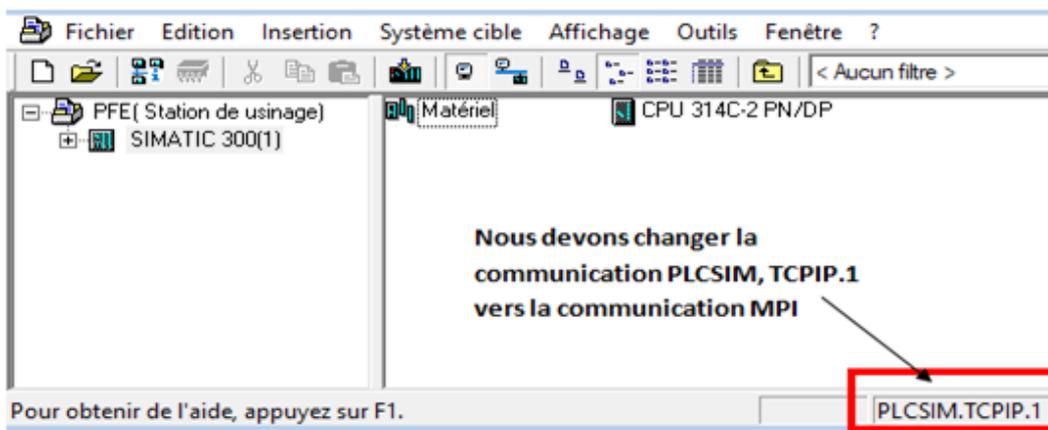


FIG. 3.75 : Changement du type de communication à l'interface MPI.

1. Dans SIMATIC Manager, nous ouvrons la fenêtre de configuration "Paramètres de l'interface PG/PC" via le menu "Outils".
2. Après l'apparition d'une liste, nous cliquons sur " Paramètres de l'interface PG/PC".

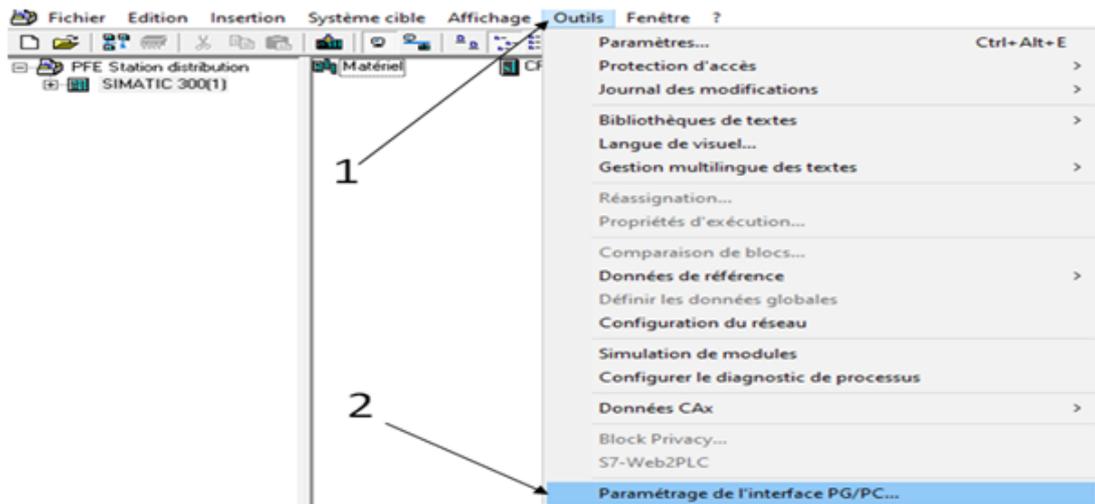


FIG. 3.76 : Menu "Paramètres de l'interface PG/PC".

La fenêtre des "Paramètres de l'interface PG/PC" apparaît et nous choisissons l'interface souhaitée dans la liste "Jeu de paramètres utilisés". Nous choisissons PC Adapter(MPI) comme indiqué sur la figure 3.77.

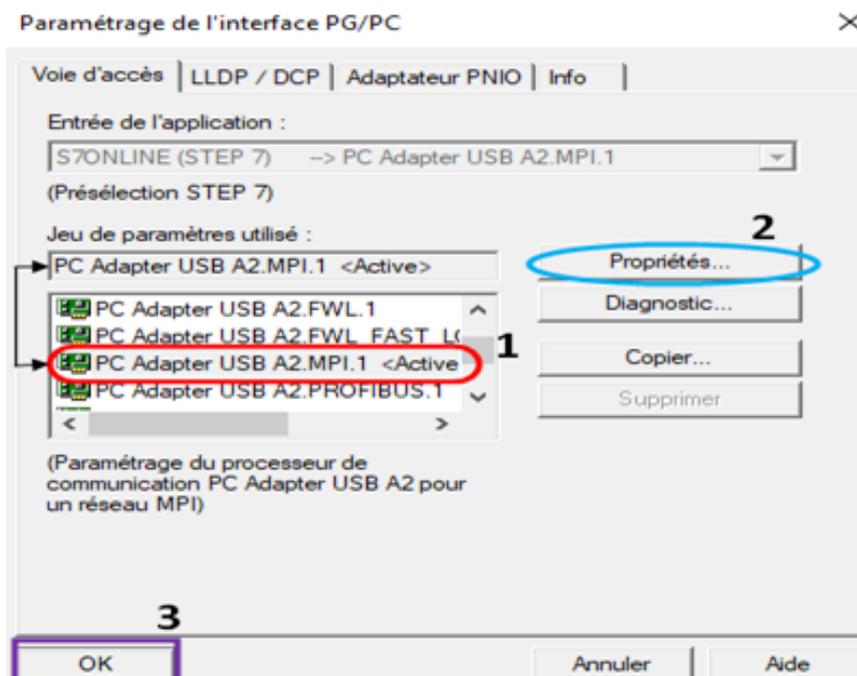


FIG. 3.77 : Le choix de la communication MPI dans la fenêtre "PG/PC interface settings".

Après avoir cliqué sur propriétés, une fenêtre apparaît. Dans la connexion locale, nous sélectionnons Connexion à USB et on clique sur OK. Si la connexion est déjà à USB, laissons-la par défaut.

Nous constatons dans la fenêtre du SIMATIC Manager que la communication entre notre projet et l'adaptateur PC MPI est établie.

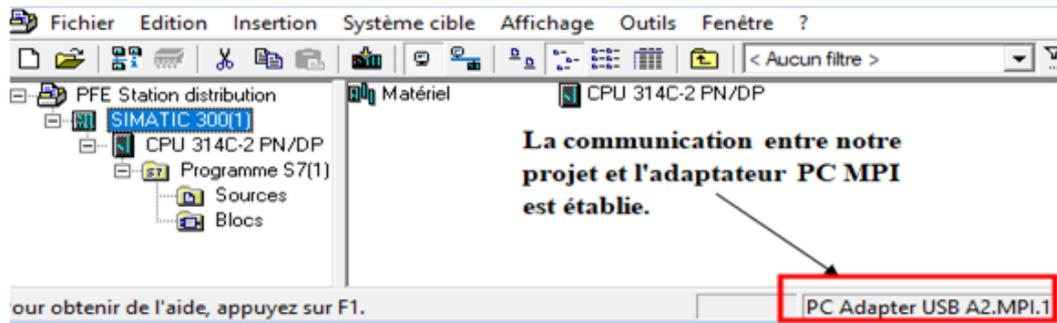


FIG. 3.78 : La communication entre notre projet et l'adaptateur PC MPI est établie.

- La connexion entre le PC et l'automate via la communication Ethernet

La deuxième méthode que nous avons utilisé pour connecter un PC avec l'automate, c'est la communication via Ethernet, c'est-à-dire que nous avons utilisé un câble Ethernet (RJ45). Au début, nous vérifions la configuration du réseau de notre projet par la vérification de l'adresse IP et du masque de sous-réseau de nos APIs de CPU 314C-2 PN/DP. En cliquant sur l'icône de configuration du réseau dans la barre des tâches de SIMATIC Manager, une fenêtre apparaîtra.

1. En cliquant sur le bouton vert, une fenêtre de propriétés - interface Ethernet s'ouvre.
2. Nous vérifions l'adresse IP et le masque de sous-réseau de nos CPU 314C-2 PN/DP.
3. Nous voyons que la communication Ethernet a été établie.
4. Nous terminons la vérification en appuyant sur OK.

Toutes ces étapes sont illustrées dans la figure ci-dessous :

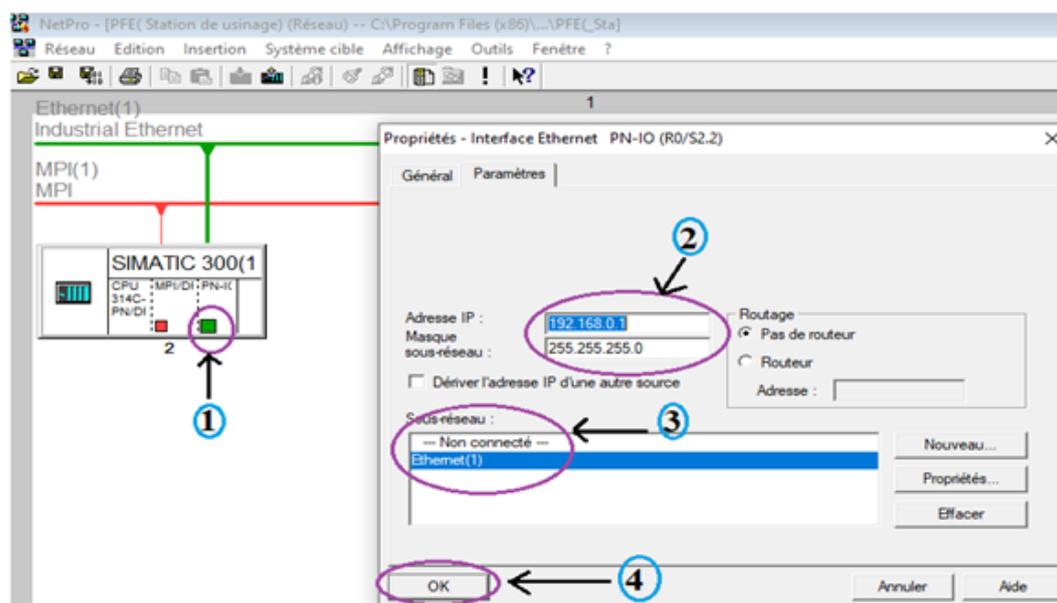


FIG. 3.79 : Vérification de la configuration du réseau de notre projet.

Chapitre 3. La programmation, simulation et l'injection au niveau des sous-stations (MPS-variant)

Après l'introduction des adresses IPs, nous allons connecter le câble Ethernet, nous allons vérifier le nom de la carte réseau compatible avec le réseau Ethernet dans la fenêtre des connexions réseau dans le PC. Nous cliquons sur la connexion Ethernet, la fenêtre d'état Ethernet s'ouvre.

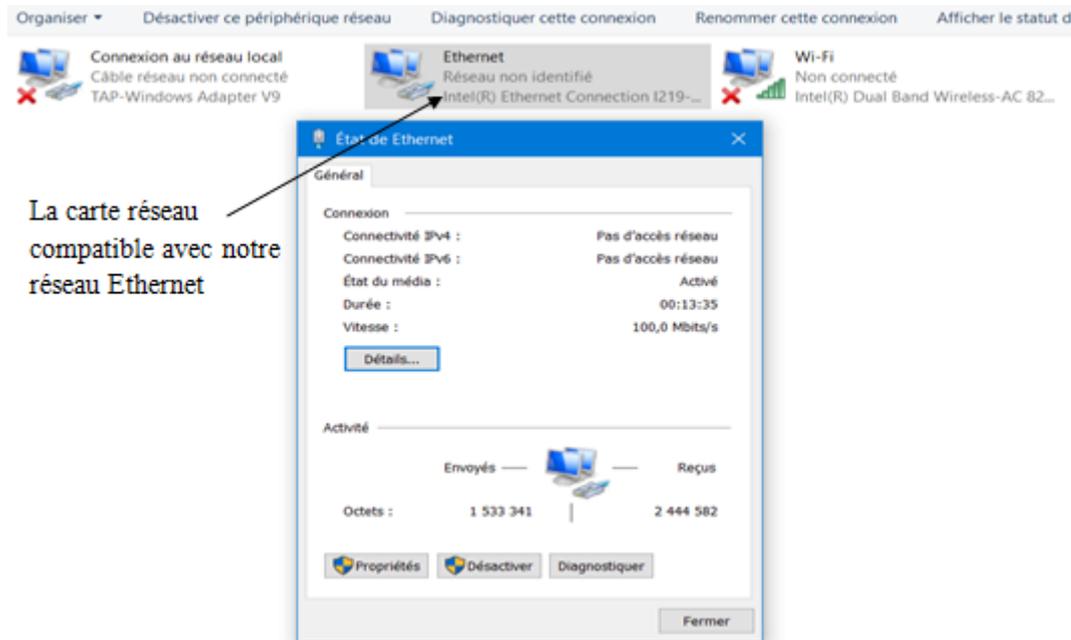


FIG. 3.80 : La fenêtre d'état d'Ethernet.

Nous cliquons sur propriétés pour vérifier/modifier l'adresse IP, une fenêtre de propriété Ethernet apparaît. Nous recherchons le protocole (TCP/IPv4).

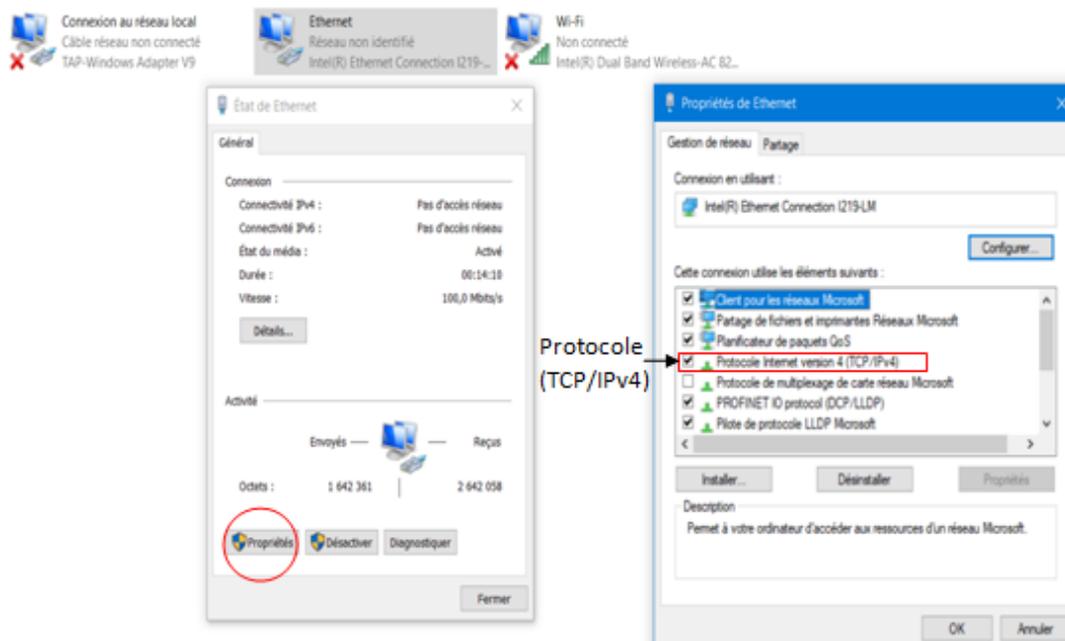


FIG. 3.81 : Recherche de la fenêtre propriétés de protocole Internet version 4 (TCP/IPv4).

Après avoir cliqué sur ce protocole une fenêtre de propriété TCP/IPv4 s'ouvre pour

Chapitre 3. La programmation, simulation et l'injection au niveau des sous-stations (MPS-variant)

modifier l'adresse IP et le masque de sous-réseau de cette connexion Ethernet. Nous devons choisir une adresse IP pour la connexion Ethernet différente de l'adresse de notre automate réel.

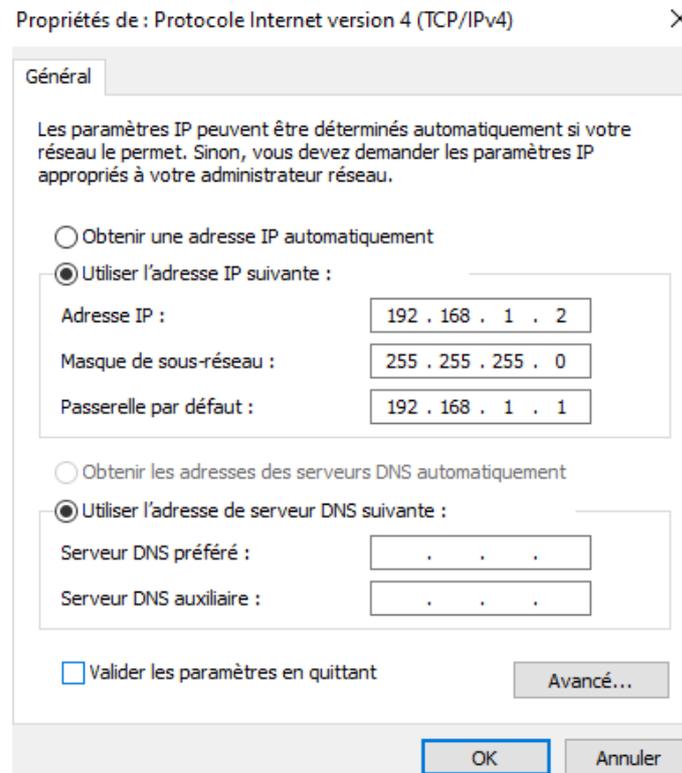


FIG. 3.82 : Fenêtre de propriétés de protocole TCP/IPv4

Nous passons ensuite à la modification de la communication de notre projet. Dans la fenêtre du SIMATIC Manager, nous constatons que la communication de notre projet n'est pas compatible avec la communication Ethernet, nous devons donc la modifier. Cette communication est modifiée dans la fenêtre "Paramètres de l'interface PG/PC" du menu "Outils" de la barre des tâches du SIMATIC Manager.

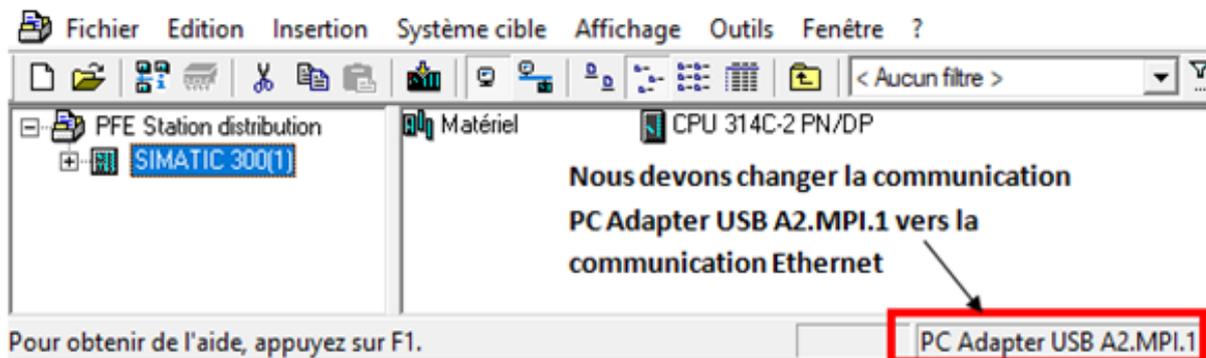


FIG. 3.83 : Changement de type de communication à l'interface Ethernet.

1. Dans SIMATIC Manager, nous ouvrons la fenêtre de configuration "Paramètres de l'interface PG/PC" via le menu "Outils".

Chapitre 3. La programmation, simulation et l'injection au niveau des sous-stations (MPS-variant)

2. Après l'apparition d'une liste, nous cliquons sur " Paramètres de l'interface PG/PC".

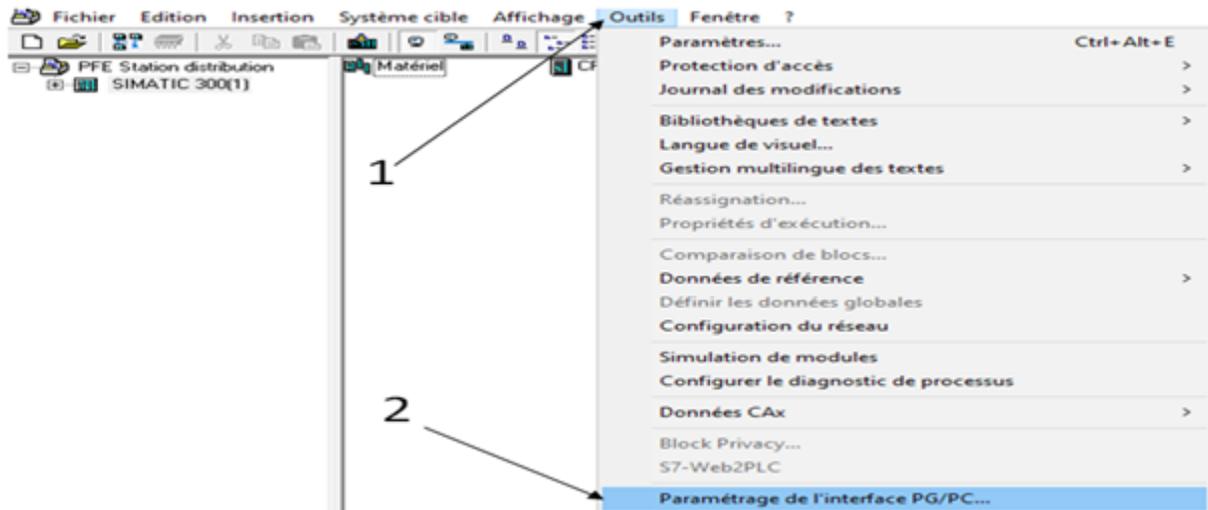


FIG. 3.84 : "Paramètres de l'interface PG/PC".

Une fenêtre des "Paramètres de l'interface PG/PC" apparaît et nous choisissons l'interface souhaitée dans la liste " Jeu de paramètres utilisés ". Nous sélectionnons le nom de la carte réseau compatible avec le réseau Ethernet, puis nous cliquons sur "Diagnostic", une fenêtre s'ouvre, nous cliquons sur "Test" puis sur OK pour confirmer la communication.

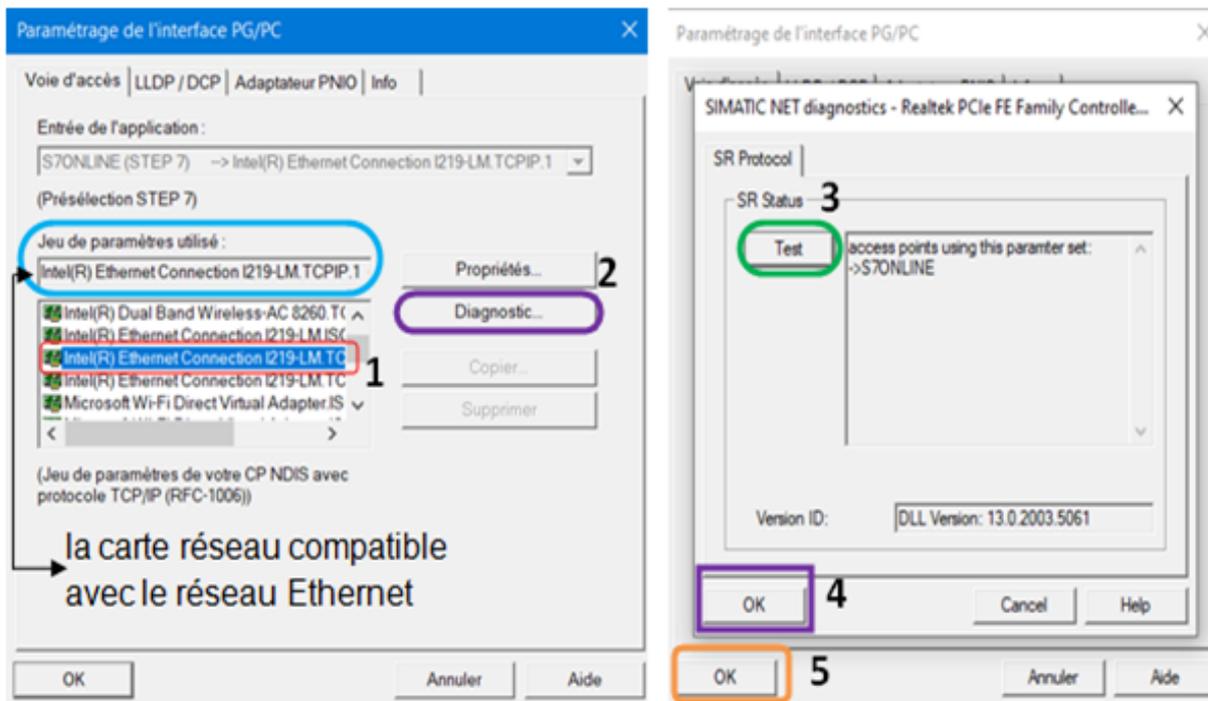


FIG. 3.85 : Sélection de la carte réseau du PC.

Nous constatons dans la fenêtre du SIMATIC Manager que la communication entre notre projet au niveau du PC et l'automate est faite via le réseau Ethernet est établie.

Chapitre 3. La programmation, simulation et l'injection au niveau des sous-stations (MPS-variant)

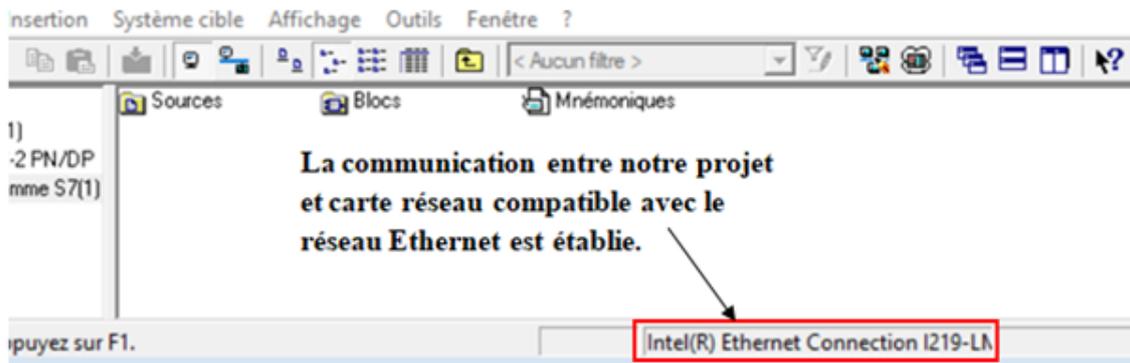


FIG. 3.86 : Communication entre PC et l'automate via Ethernet.

Après le changement de communication, nous passons à l'édition de la Station Ethernet, pour cela nous cliquons sur le menu "Système cible", une liste apparaît, nous cliquons sur "Éditer Station Ethernet".

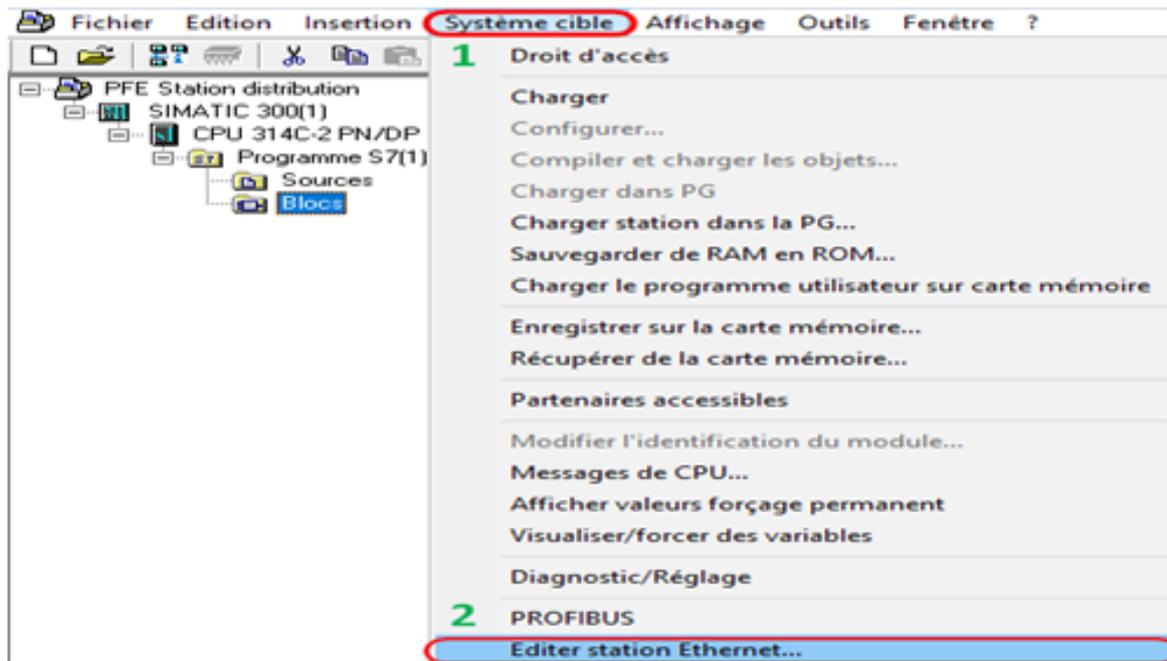


FIG. 3.87 : Menu pour ouvrir l'Éditer Station Ethernet.

Après avoir cliqué sur "Éditer Station Ethernet", une fenêtre s'ouvre.

Nous cliquons sur "Parcourir", une fenêtre s'ouvre, pour chercher l'adresse IP qui se fait automatiquement. Après avoir trouvé l'adresse IP correspondante, nous appuyons sur OK.

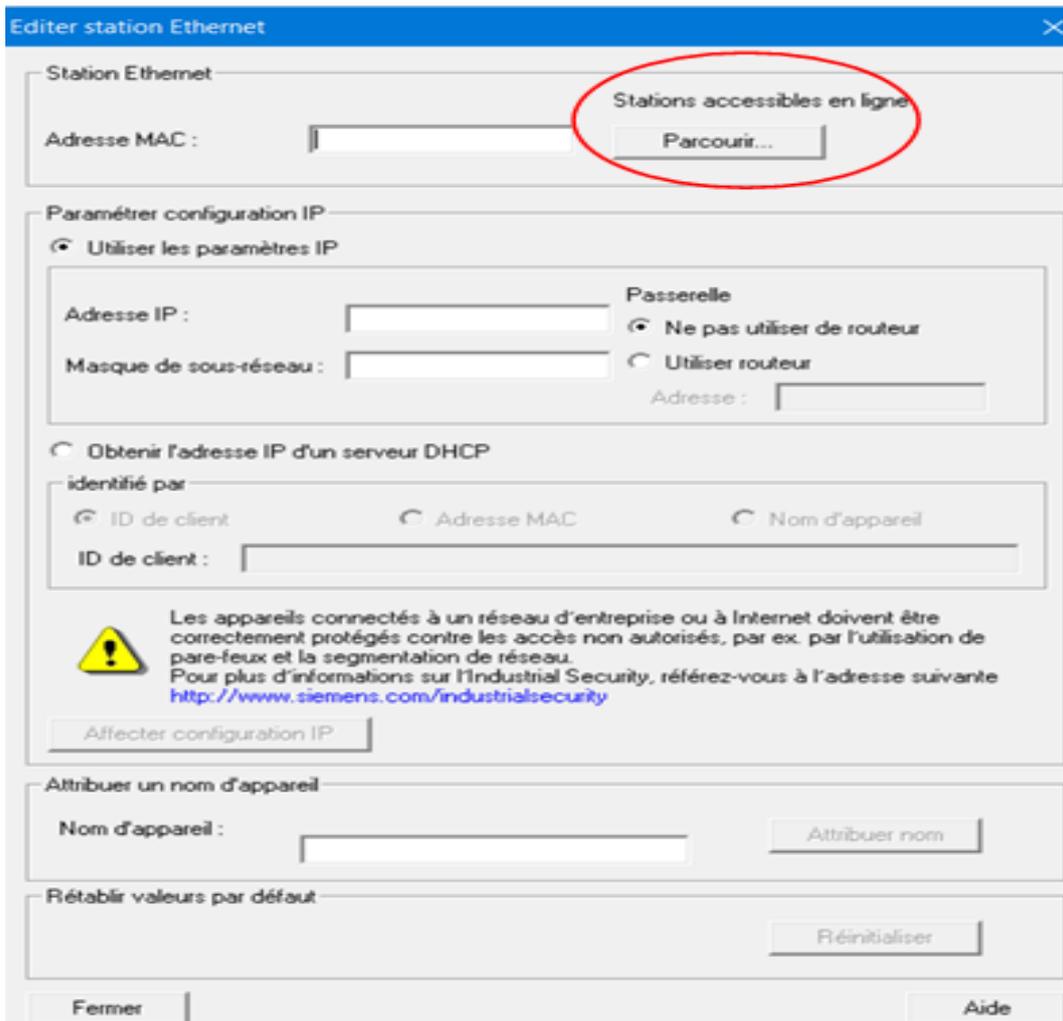


FIG. 3.88 : Fenêtre "Édit Station Ethernet".

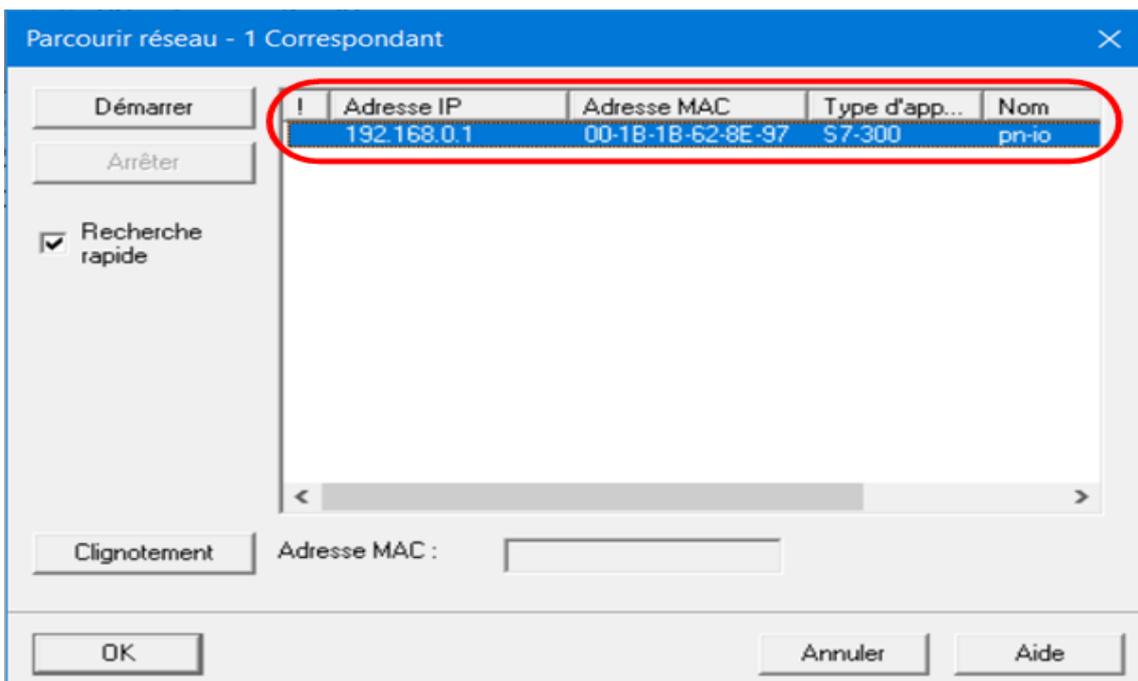


FIG. 3.89 : Recherche automatique de l'adresse IP.

Chapitre 3. La programmation, simulation et l'injection au niveau des sous-stations (MPS-variant)

Après avoir cliqué sur OK, nous observons que la station Ethernet est correctement éditée.

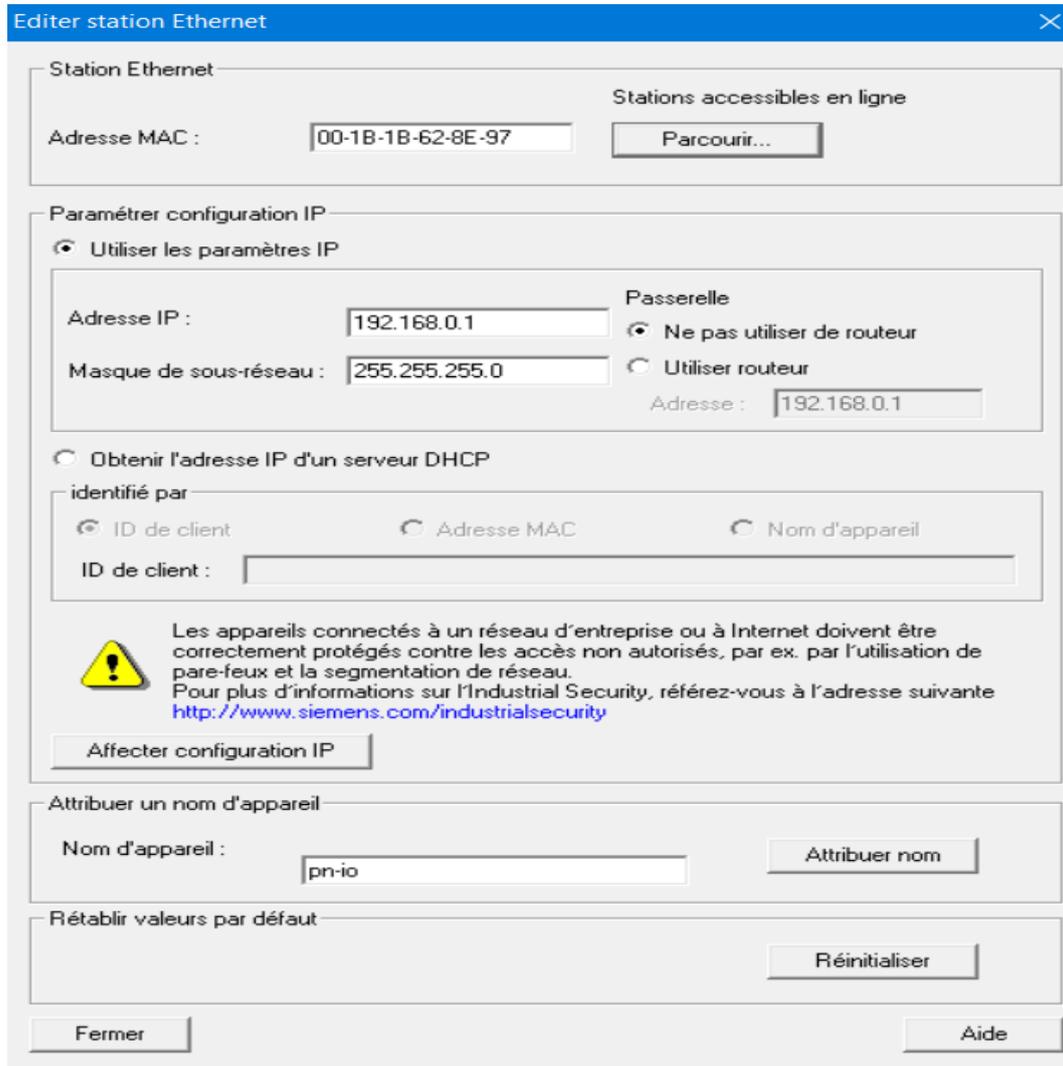


FIG. 3.90 : Fenêtre "Édit Ethernet Station" après la recherche de l'adresse IP.

Le chargement du programme avec la communication Profibus se fait de la même manière que la communication MPI, sauf que nous allons utiliser un câble Profibus, changer la configuration du réseau de notre projet et choisir l'interface désirée dans les "paramètres de l'interface PG/PC". Pour changer la configuration du réseau de notre Projet, nous devons suivre les étapes illustrées dans les figures ci-dessous. Nous devons d'abord ouvrir la fenêtre de configuration du réseau en cliquant sur l'icône  dans la barre des tâches de SIMATIC Manager. Puisque l'automate que nous utilisons ne contient qu'une seule interface RS485 qui est utilisée par les câbles de communication MPI et Profibus donc on ne peut pas utiliser les deux communications en même temps. Pour établir la communication Profibus, nous devons d'abord changer la communication MPI vers Profibus.

Chapitre 3. La programmation, simulation et l'injection au niveau des sous-stations (MPS-variant)

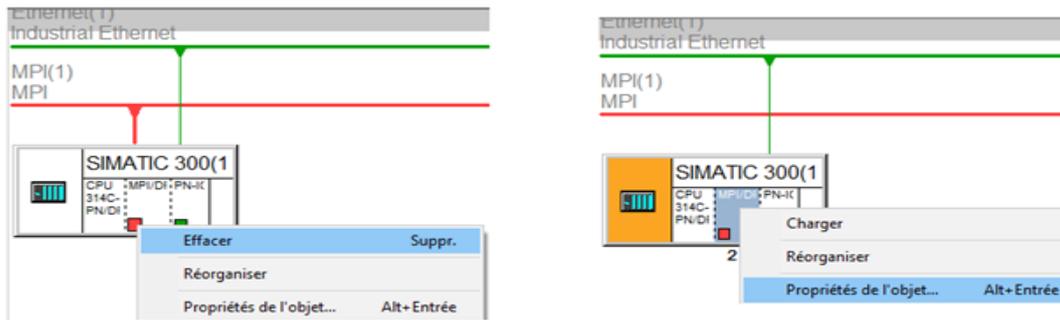


FIG. 3.91 : Changement de la communication MPI vers Profibus.

Quand nous cliquons sur "Propriétés de l'objet" une fenêtre apparaît, nous choisissons le type d'interface et son nom, après avoir cliqué sur OK une autre fenêtre s'ouvre, nous cliquons sur "Nouveau".

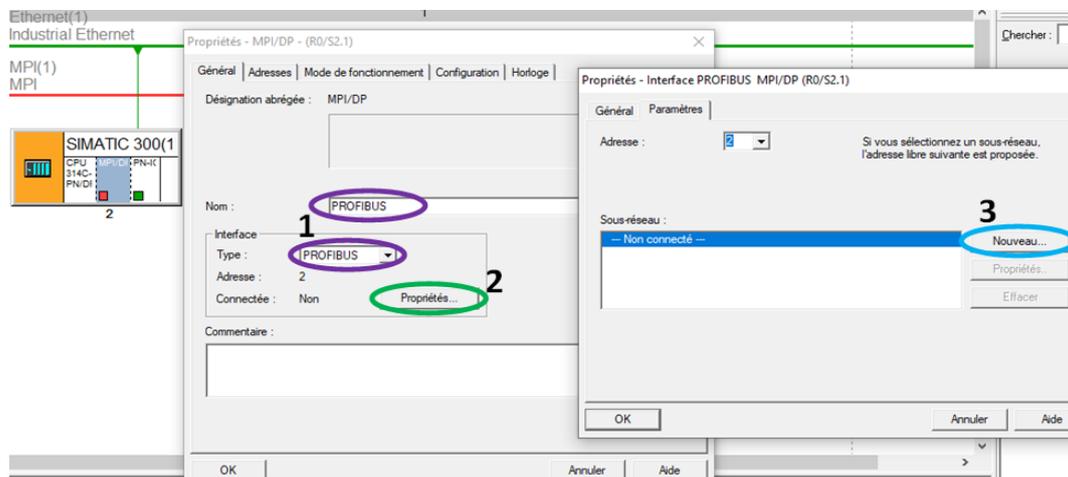


FIG. 3.92 : Le choix de communication Profibus (a).

Après avoir cliqué sur "Nouveau", une autre fenêtre s'ouvre, nous cliquons sur OK.

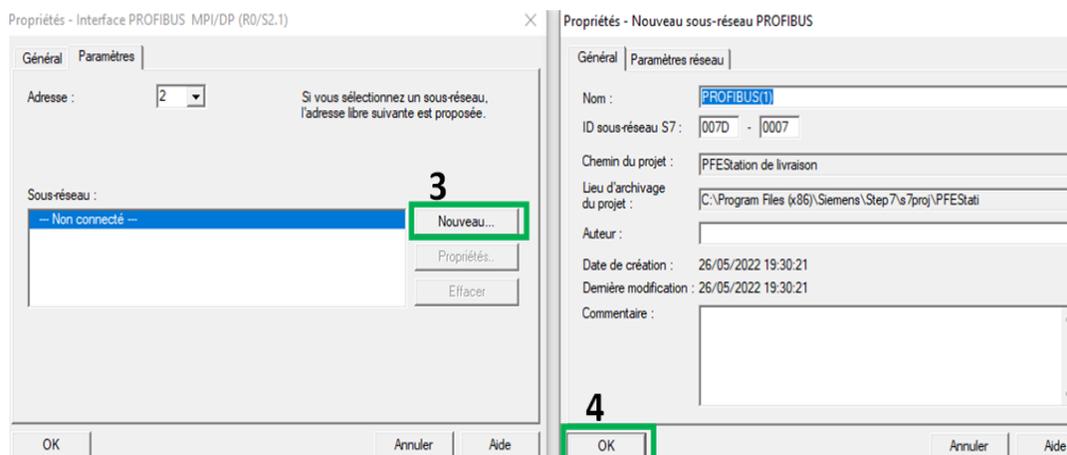


FIG. 3.93 : Le choix de la communication Profibus (b).

D'après la figure 3.94, nous voyons que la communication Profibus est établie dans la configuration du réseau de notre projet.

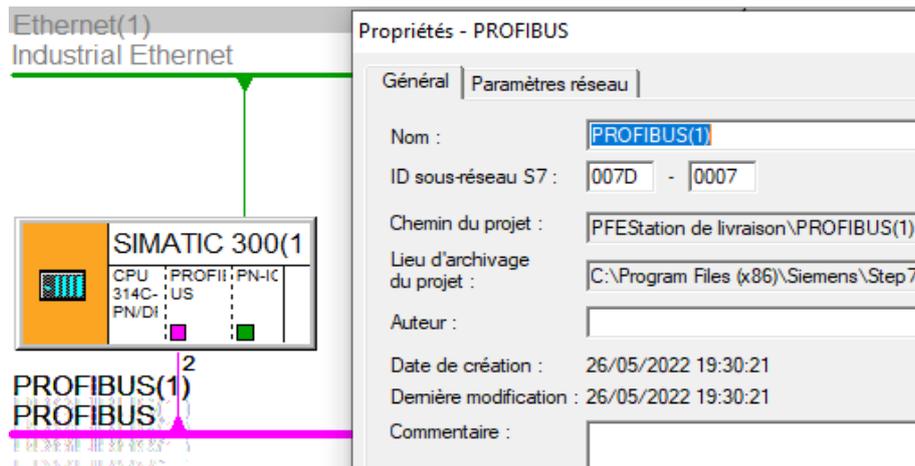


FIG. 3.94 : Communication Profibus établie.

Le chargement de la partie matérielle et de la partie programme se fait de la même manière que ce soit avec une communication MPI, Ethernet ou Profibus. Il est important de noter que l'automate doit être en mode stop pendant le chargement soit de la partie matérielle ou programme.

b. Étape N°02 : Chargement de la partie matérielle (Hardware).

Nous cliquons sur matériel pour entrer dans la zone de configuration du matériel.

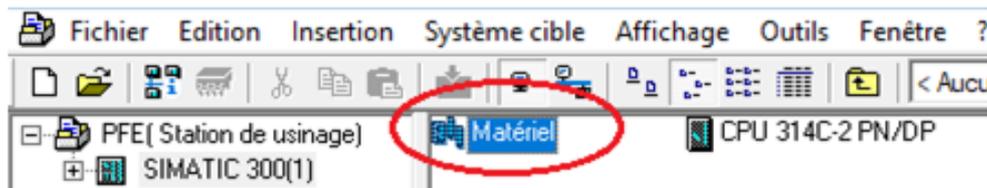


FIG. 3.95 : La fenêtre du SIMATIC Manager.

Après avoir cliqué sur matériel la fenêtre HW Config (Configuration matérielle) s'ouvre.

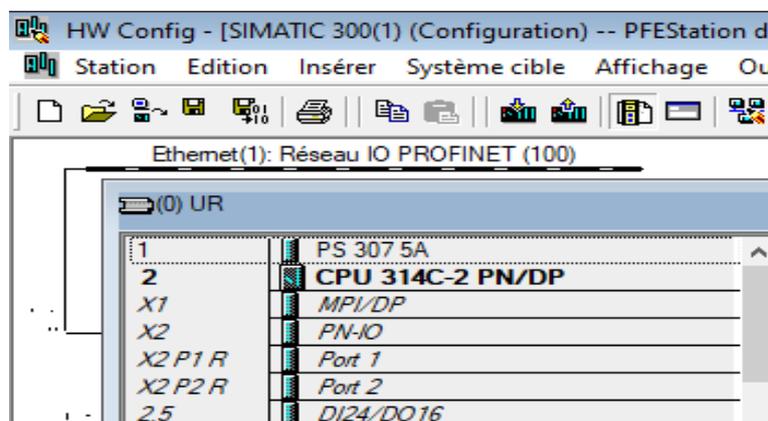


FIG. 3.96 : La fenêtre de configuration matérielle (HW Config).

Chapitre 3. La programmation, simulation et l'injection au niveau des sous-stations (MPS-variant)

Nous cliquons sur l'icône  de chargement dans le module, une fenêtre s'ouvre qui indique où se trouve le CPU 314C 2PN/DP dans le châssis N°0 et l'emplacement N°2 et on clique sur le bouton OK.

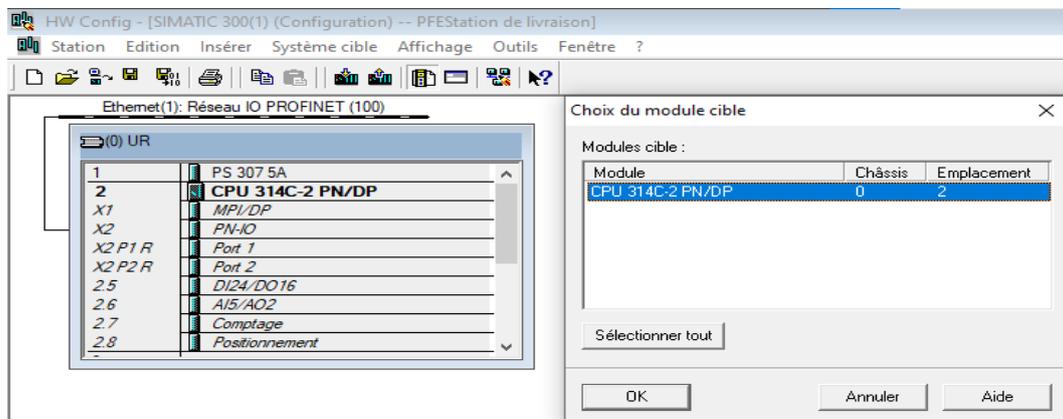


FIG. 3.97 : Le choix du module cible.

Lorsque nous appuyons sur le bouton OK, une autre fenêtre s'ouvre, nous cliquons sur "Affichage", notre CPU (314C-2PN/DP) apparaît, puis nous cliquons sur OK.

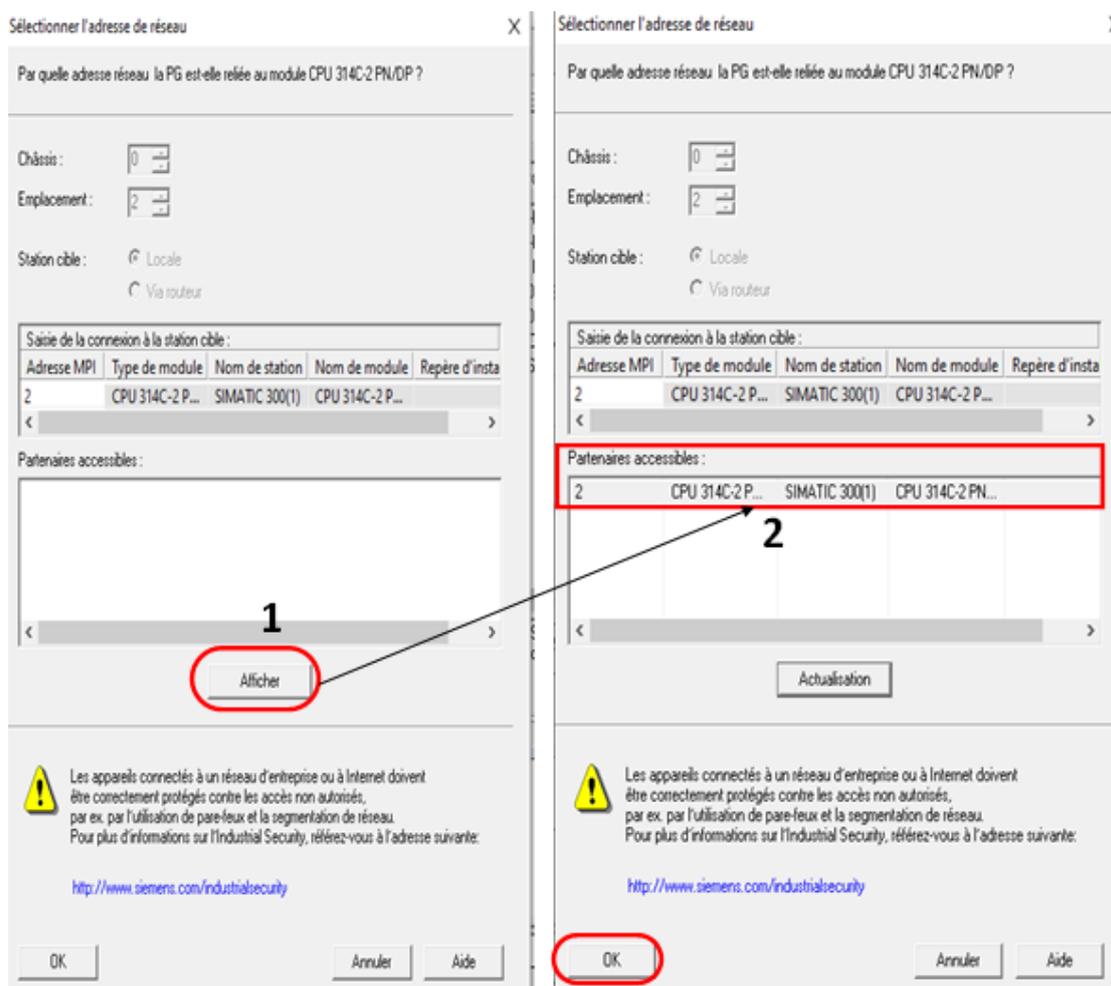


FIG. 3.98 : Chargement avec la communication MPI.

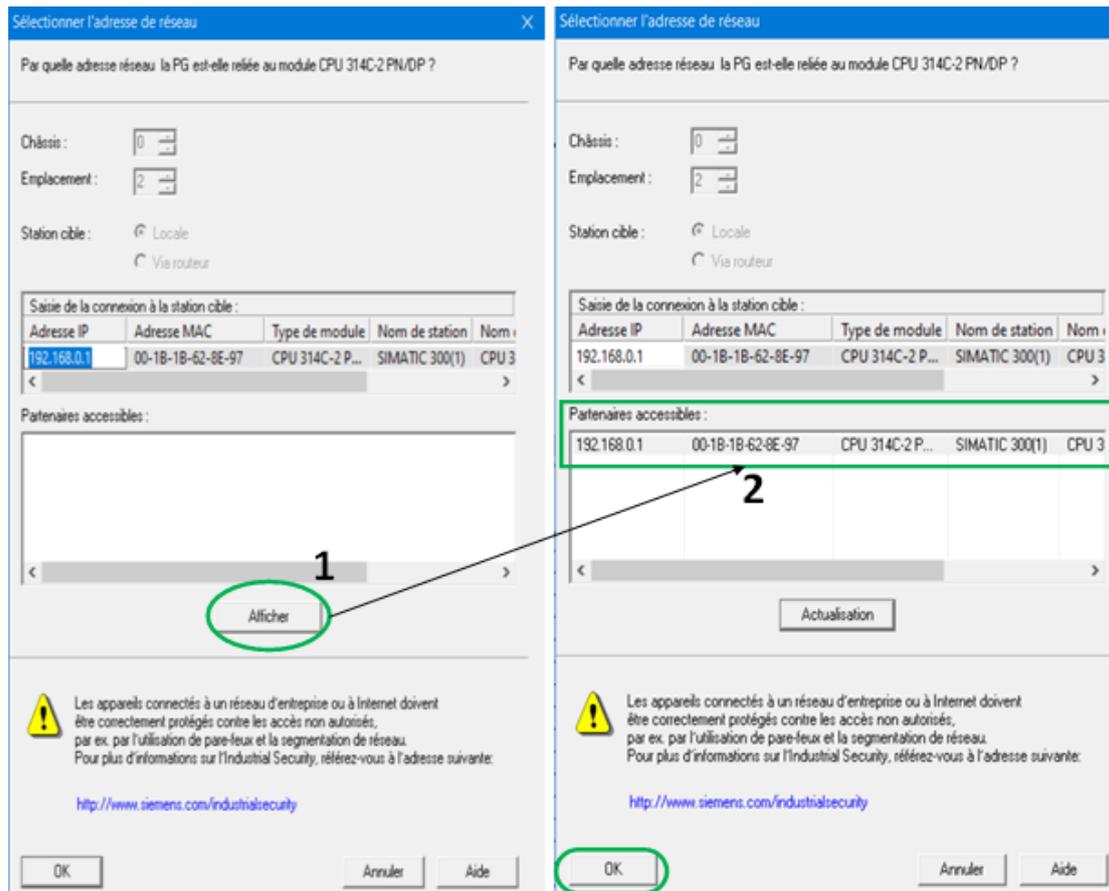


FIG. 3.99 : Chargement avec la communication Ethernet.

Après avoir cliqué sur OK, la partie matérielle se chargera dans notre automate réel, comme le montre la figure 3.100.

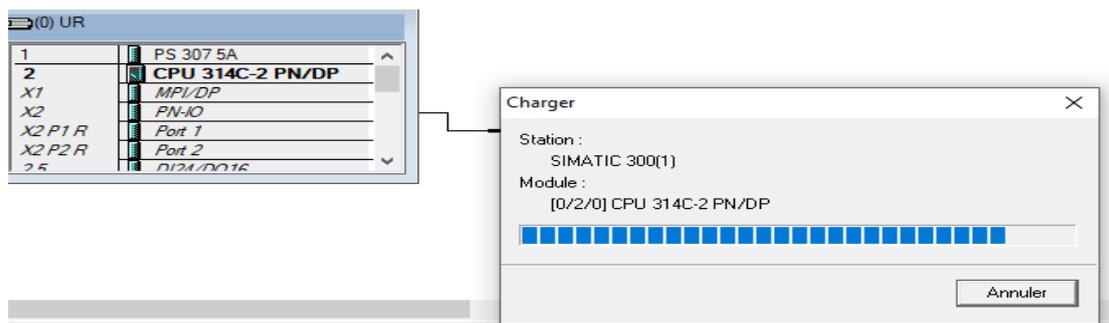


FIG. 3.100 : Chargement de la configuration matérielle.

c. Étape N°03 : Chargement de la partie programme (Software).

Après avoir chargé la partie matérielle, nous passons au chargement de la partie programmation. Nous devons d'abord accéder à la partie programmation, puis nous entrons au niveau « Blocs ». On doit sélectionner tous les blocs et toutes les données du système qui se trouvent dans le programme. Ensuite en cliquant sur l'icône "charger" dans la barre d'outils, nous pouvons charger tous les blocs.

Chapitre 3. La programmation, simulation et l'injection au niveau des sous-stations (MPS-variant)

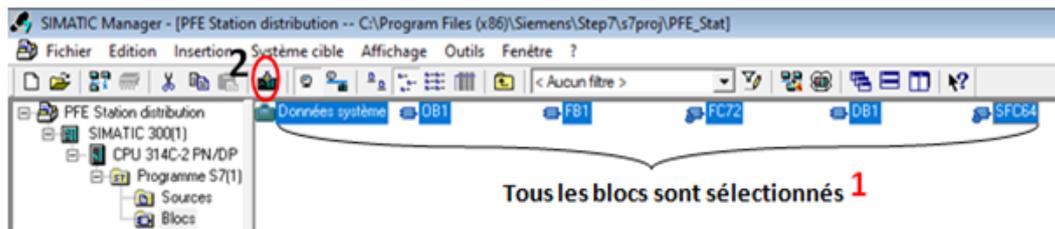


FIG. 3.101 : Sélection de tous les blocs.

Après avoir cliqué sur charger, nous voyons que notre programme est en train de se charger dans l'automate réel.

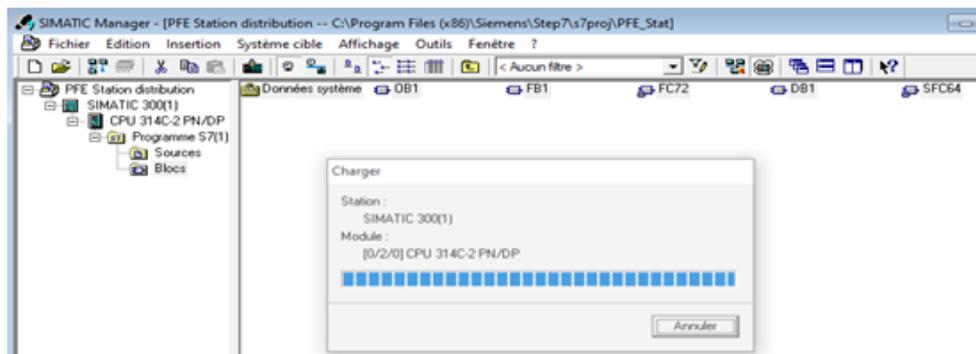


FIG. 3.102 : Chargement des blocs.

Pour confirmer le chargement de la partie fonctionnelle (bloc FB1), nous nous rendons sur cette partie et nous cliquons sur l'icône "charger" dans la barre d'outils, une fenêtre s'ouvre nous sélectionnons les deux rubriques puis nous cliquons sur OK (figure 3.103), nous remarquons que le bloc fonctionnel va se charger, cela signifie que notre projet est bien injecté dans notre automate réel. Enfin, pour voir la visualisation, nous cliquons sur le bouton visualisation "lunette" qui se trouve dans la barre d'outils.



FIG. 3.103 : Chargement du bloc FB1 (partie fonctionnelle).

D'après la figure 3.104, nous apercevons que l'injection des programmes dans notre automate réel est bien faite via la communication MPI.

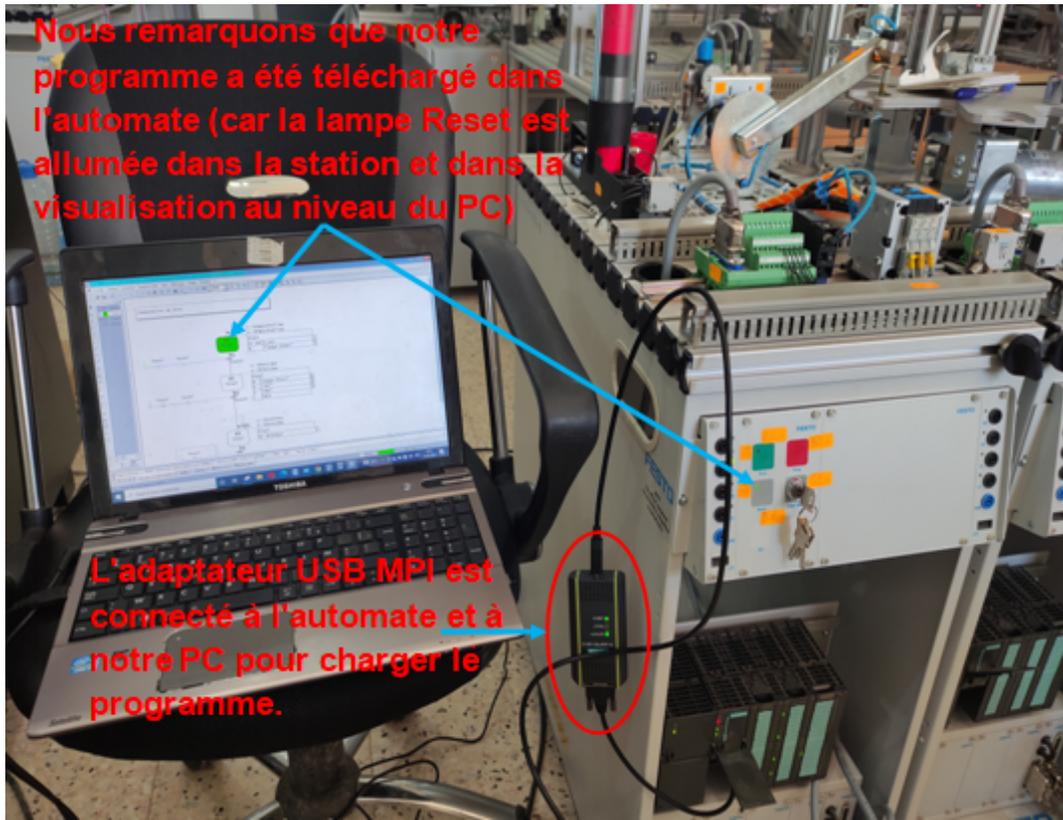


FIG. 3.104 : Le chargement du programme via la communication MPI.

3.8 Résultats obtenus après l'injection de chaque sous-station par son programme

Après l'injection des programmes au niveau des API, nous avons synchronisé les quatre stations en fonction du temps. Nous éliminons l'intercommunication entre les stations (capteurs et actionneurs) pour gagner un peu du temps. Aussi, nous calibrons certains actionneurs et capteurs afin que notre variante du système MPS fonctionne en mode optimal.

La figure 3.105 illustre notre nouveau système de production automatisé, MPS variant (distribution, usinage, manipulation "manutention", livraison "tri").



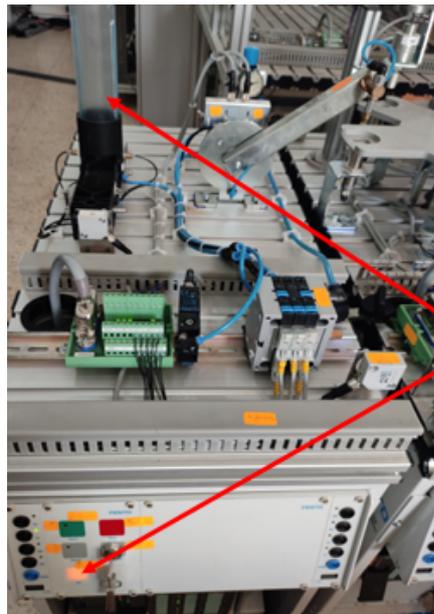
FIG. 3.105 : Notre nouveau système de production automatisé MPS variant

3.8.1 Sous-station de distribution

La figure 3.106 montre une prise de vue successive de cycle du fonctionnement de la station de distribution, le vérin d'éjection éjecte une pièce à partir du magasin, puis le vérin oscillant est déplacé vers la position "magasin". Une fois la pièce aspirée, le vérin oscillant tourne jusqu'à la position "station d'usinage", lorsqu'il arrive à cette station, il pose la pièce.



FIG. 3.106 : Résultats obtenus pour la sous-station du distribution.

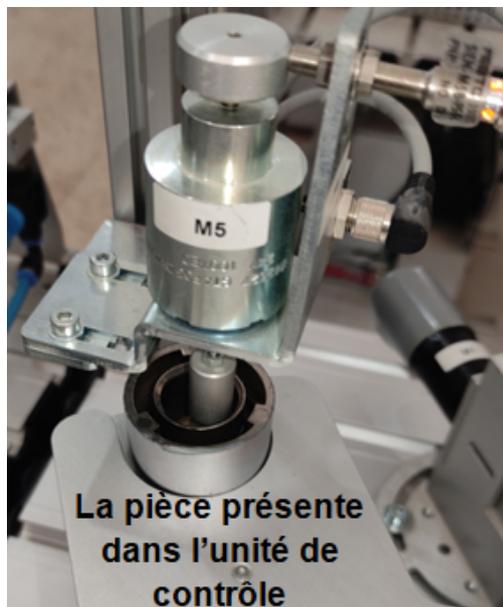


Le voyant "Magasin vide" est allumé car il n'y a pas de pièces dans le magasin.

FIG. 3.107 : Le voyant "magasin vide" de la sous-station de distribution.

3.8.2 Sous-station d'usinage

La figure 3.108 montre les deux opérations de contrôle et perçage effectuées sur le plateau à indexation rotatif. Le traitement d'une seule pièce s'effectue en deux cycles.



La pièce présente dans l'unité de contrôle



Le dispositif de serrage fixe la pièce et la perceuse descend

FIG. 3.108 : Les résultats obtenus pour la sous-station d'usinage.

3.8.3 Sous-station de manipulation

La figure 3.109 montre le déplacement de la pièce à partir de la station d'usinage jusqu'à la station de tri.

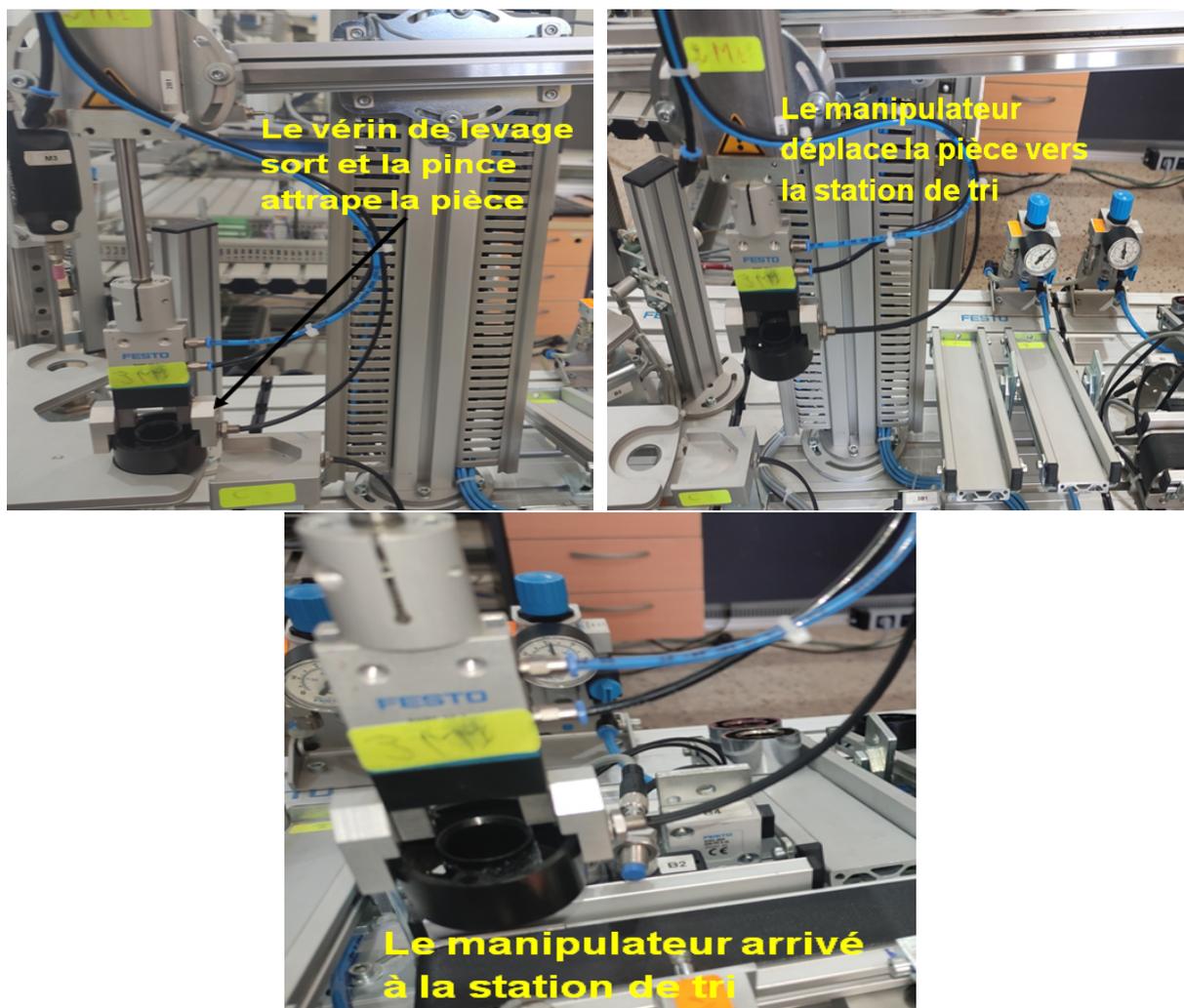


FIG. 3.109 : Les résultats obtenus pour la sous-station de manipulation.

3.8.4 Sous-station de livraison (tri)

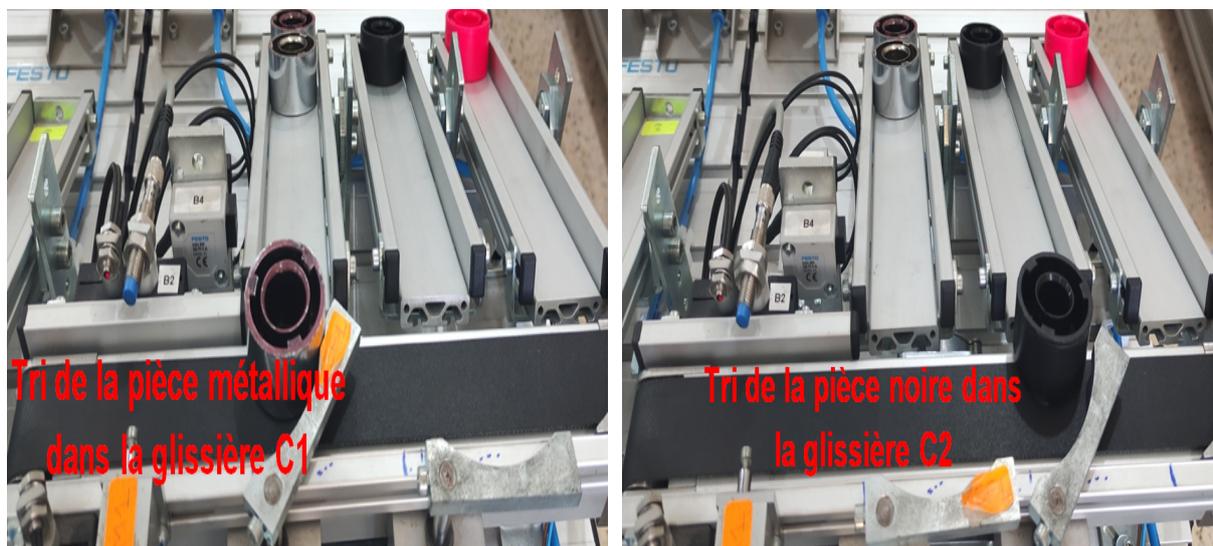


FIG. 3.110 : Les résultats obtenus pour la sous-station de livraison (tri).

Chapitre 3. La programmation, simulation et l'injection au niveau des sous-stations (MPS-variant)

La figure 3.110 illustre le tri des pièces dans les trois glissières en fonction de leurs couleurs et type (noir, rouge ou métallique).

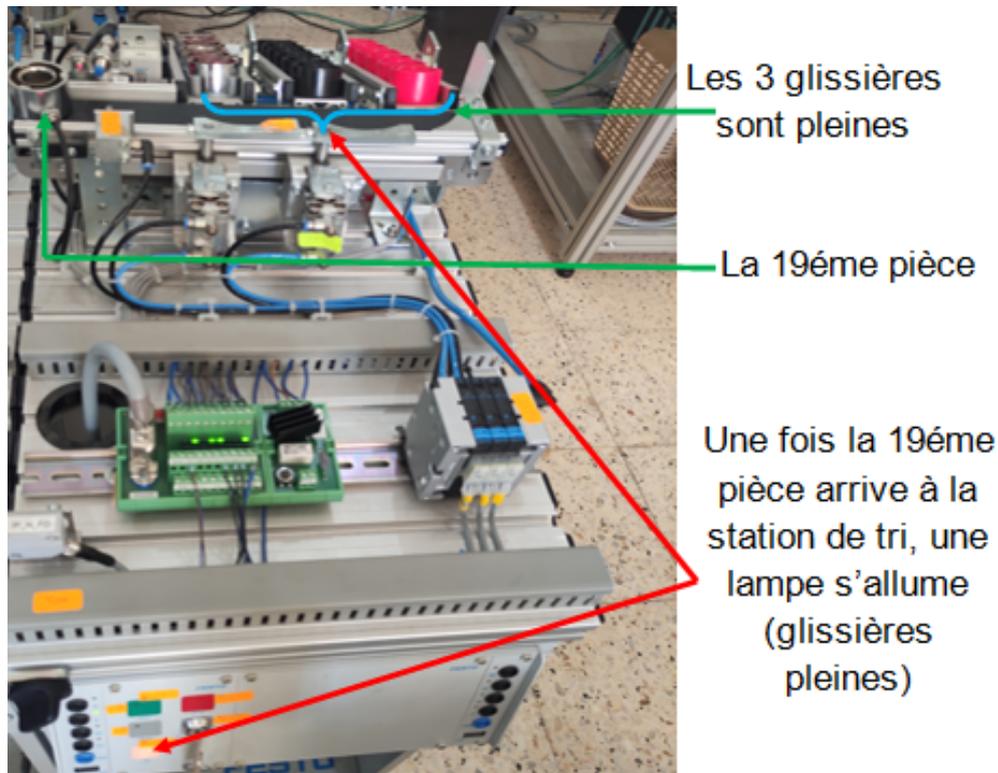


FIG. 3.111 : Le voyant "glissières pleines" de la sous-station de livraison (tri).

3.9 Conclusion

Dans ce chapitre, nous avons exposé les différentes étapes à suivre pour la configuration et la programmation du système étudié. Nous avons commencé par les étapes de création du projet et la configuration du logiciel et du matériel dans le logiciel STEP 7 de Siemens. Ensuite, nous avons expliqué les différentes variables (Entrées et sorties) qu'on trouve dans chaque sous-station, où nous avons créé pour chaque sous-station son propre Grafset. Par la suite, nous avons simulé les projets sous le simulateur S7-PLCSIM, après nous avons vérifié le fonctionnement des programmes via la simulation des projets. L'étape suivante était de faire la simulation et le test du fonctionnement des programmes via l'outil EasyPort USB de FESTO afin de confirmer le bon fonctionnement sur les sous-stations réelles. Dans cette partie, nous avons injecté les programmes dans les sous-stations et lancer le système. Le fonctionnement était parfait et sans problème.

CHAPITRE

4

LA CONCEPTION, SIMULATION ET
SUPERVISION DE NOTRE MPS
VARIANT

4.1 Introduction

Les outils de supervision sont utilisés afin de créer des interfaces homme-machine sous des panneaux de supervision qui sont plus ou moins complexes et divers. La majorité des fabricants des IHMs disposent de leurs propres logiciels de supervision, dont on peut citer :

- WinCC de la société Siemens.
- Vijeo Look de la société Schneider Electric.
- FactoryTalk de la société Rockwell Automation.
- PcVue de la société Arc Informatique.
- InTouch de la société Wonderware.

Dans notre projet, nous avons travaillé sur un système utilisant des automates Siemens, pour cela nous allons utiliser l'outil de supervision WinCC flexible de Siemens qui est dédié aux automates de Siemens qui contrôlent le système MPS variant que nous étudions. Dans ce chapitre, nous allons utiliser les résultats du précédent (chapitre 3) pour la création des IHMs pour chaque station du système étudié.

4.2 WinCC flexible

WinCC flexible est un logiciel d'ingénierie pour la conception et la configuration d'IHM sur pupitre classique, pupitre tactile ou écran d'ordinateur, pour la supervision d'automates programmables de type Siemens. Il est utilisé avec toutes les gammes d'outils de programmation des automates Siemens (STEP7, TIA Portal), il permet d'intégrer des projets STEP7. Le WinCC flexible combine plusieurs avantages : simplicité, ouverture, flexibilité.[73]. Il rend possible la réalisation d'interfaces homme-machine sur tous les pupitres d'opérateur, du micro-panel au multi-panel, il offre également un logiciel de surveillance RUNTIME pour les monopostes basés sur PC[74]. WinCC flexible est accessible dans le menu de démarrage sous "WinCC flexible".



FIG. 4.1 : L'icône WinCC flexible sur le bureau de Windows.

4.2.1 WinCC flexible Runtime

WinCC flexible Runtime est un logiciel permettant de visualiser le processus, ainsi que la simulation des vues IHM du processus. Au moment de l'exécution, l'opérateur peut effectuer le contrôle du processus[73]. Après la mise en communication avec les API, nous aurons le pouvoir de :

- Visualiser des vues sur l'écran.
- Contrôler des procédés, comme les spécifications d'instructions ou l'ouverture et de la fermeture de vannes.
- Archiver des informations en temps réel, des valeurs du processus et des évènements d'alarme.

4.2.2 Démarrage de WinCC flexible

Afin de lancer le logiciel WinCC flexible, il faut juste double-cliquer sur l'icône WinCC flexible sur le bureau de Windows.

4.2.3 Création d'un projet

Une fois WinCC flexible s'ouvre, une fenêtre apparaîtra avec une liste d'options, nous pouvons choisir l'une de ces options (ouvrir le dernier projet édité, créer un projet avec l'assistant de projet, ouvrir un projet, créer un projet vide), comme indiquer dans la figure 4.2.

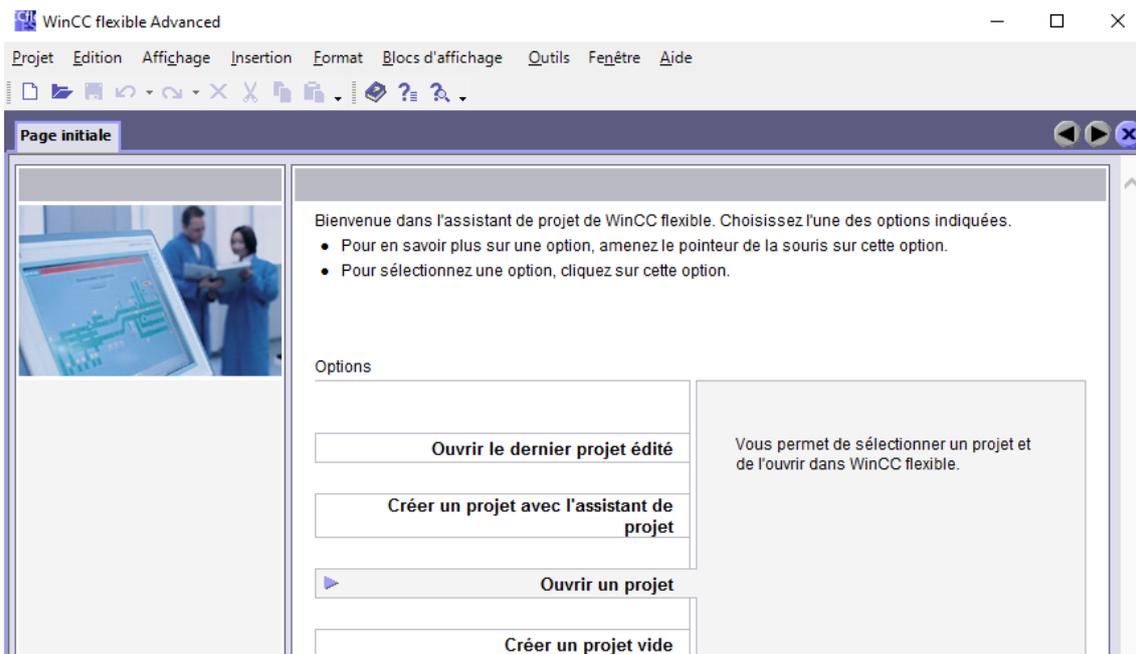


FIG. 4.2 : L'interface principale du logiciel WinCC flexible.

Pour commencer, nous devons d'abord créer un nouveau projet en cliquant sur l'icône "nouveau" de la barre d'outils.

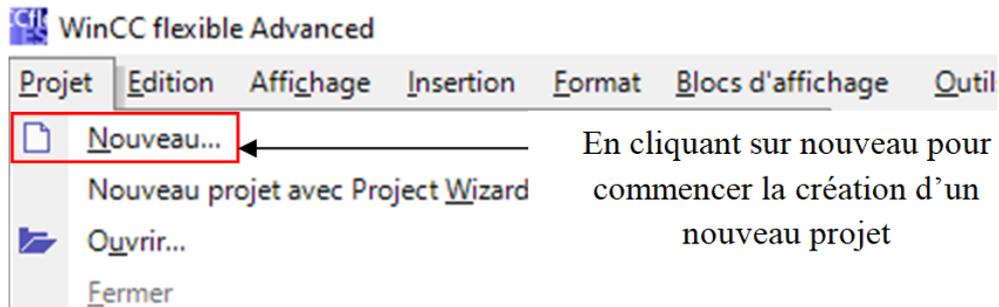


FIG. 4.3 : Création d'un nouveau projet.

4.2.4 Sélection du pupitre

Après avoir cliqué sur " nouveau ", une fenêtre s'affiche pour choisir le type de pupitre. Dans notre travail, nous avons choisi WinCC flexible Runtime afin de contrôler le fonctionnement de notre système directement par le PC. Autrement dit, notre PC est devenu un écran de supervision.

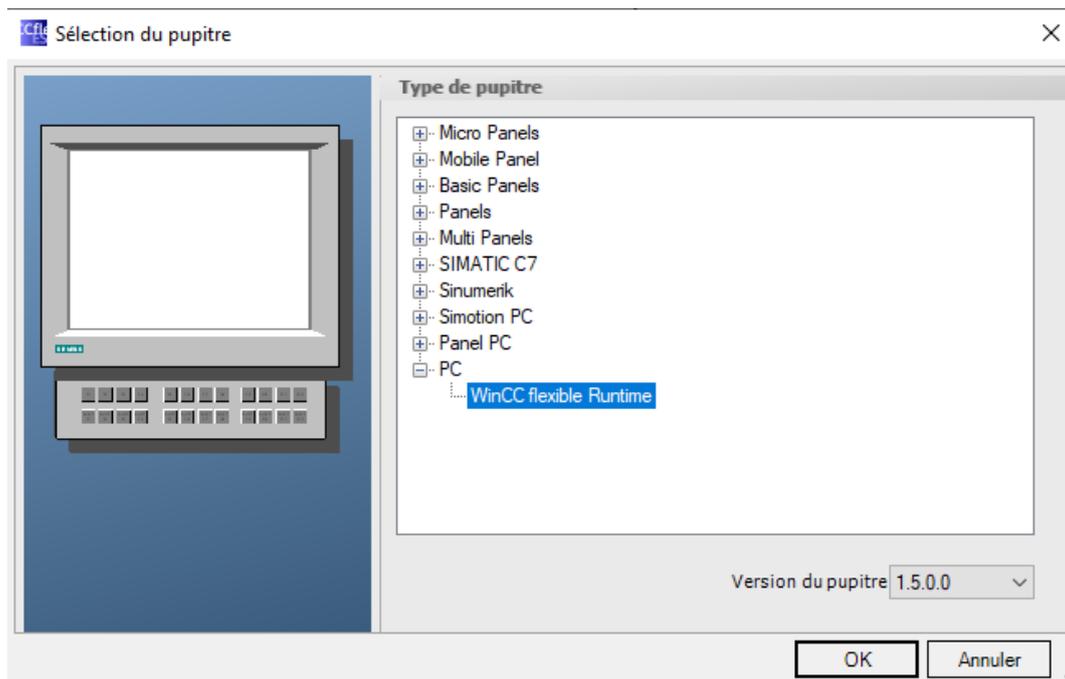


FIG. 4.4 : Sélection du pupitre de commande.

Nous validons le choix de notre type de pupitre en appuyant sur OK, le projet sera créé comme le montre la figure 4.5.

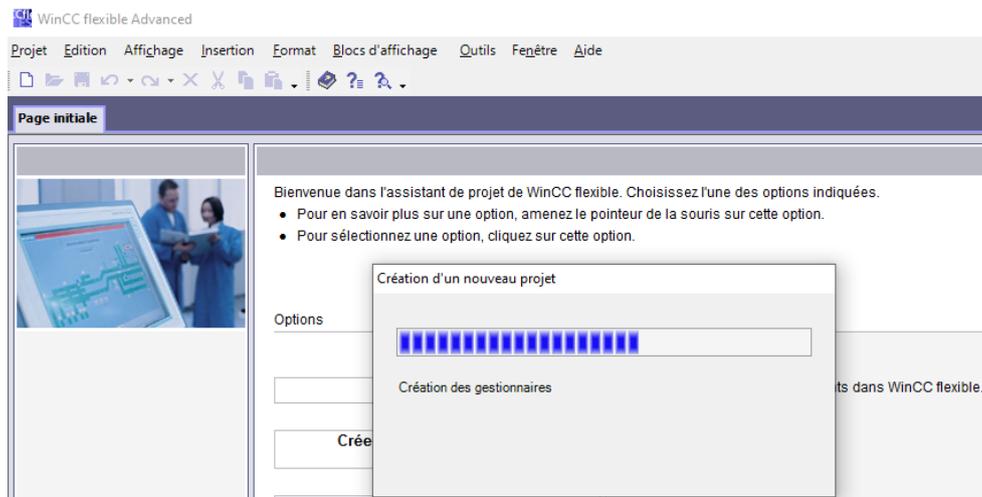


FIG. 4.5 : Chargement d'un nouveau projet.

Après la création d'un nouveau projet, le pupitre que nous avons choisi s'affiche dans la fenêtre de travail du WinCC flexible.

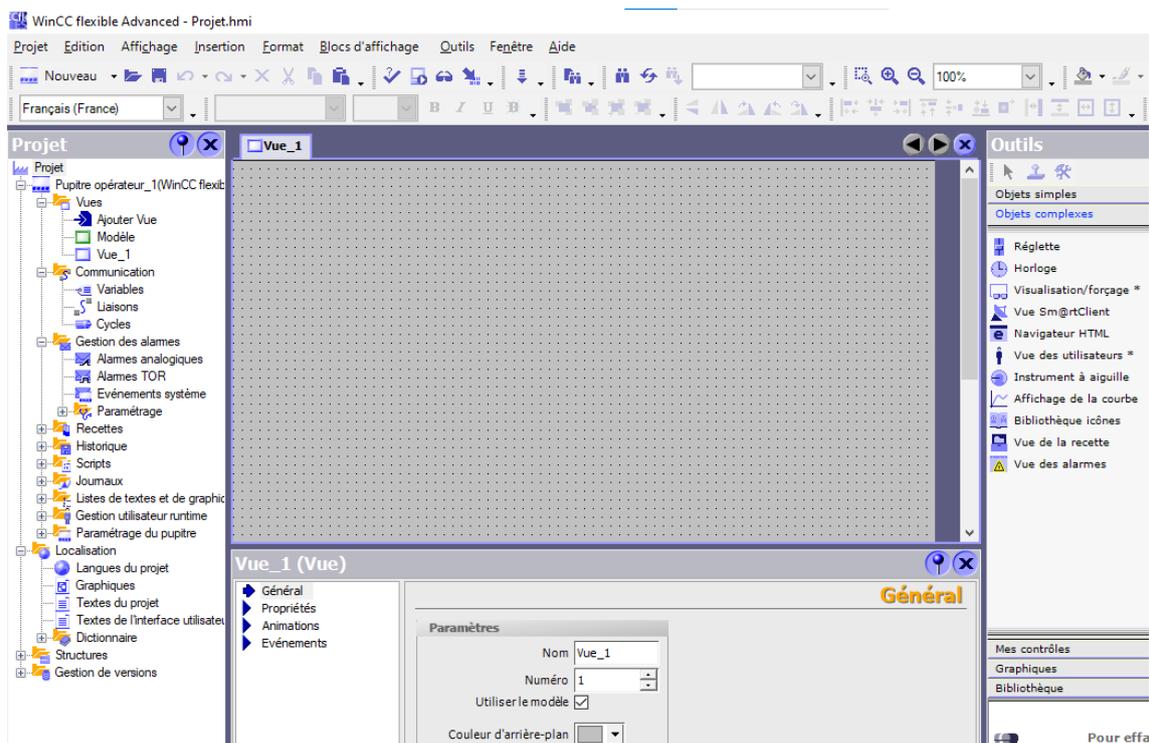


FIG. 4.6 : L'interface de travail au niveau du WinCC flexible.

4.2.5 Intégration du projet avec le SETP7

Le logiciel WinCC flexible permet l'intégration des projets réalisés par STEP7, pour cela le nous allons intégrer dans WinCC flexible le programme que nous avons déjà réalisé sur STEP 7, ceci va nous faciliter la réalisation de notre IHM. De plus, cette intégration permet de réduire les risques d'erreurs et de réduire les tâches de configuration

Lors de l'intégration d'un projet WinCC flexible dans un projet Step 7, nous bénéficions de plusieurs avantages :

- Le SIMATIC Manager peut être considéré comme un outil central pour la création, l'édition et la gestion des automates SIMATIC et des projets WinCC flexibles.
- Les mnémoniques doivent être définies une seule fois dans STEP 7, afin de pouvoir les utiliser sous STEP 7 et sous WinCC flexible. Donc, quand on configure des variables et des pointeurs au niveau de la zone de travail, nous pouvons accéder aux mnémoniques de STEP 7 directement sous WinCC flexible.
- Lors de la création du projet flexible WinCC, les paramètres de communication de l'automate sont saisis par défaut. Chaque modification sous STEP 7 entraîne une mise à jour des paramètres de communication sous flexible WinCC.

Pour intégrer notre projet WinCC flexible dans le projet STEP 7, nous devons cliquer sur "Projet", après qu'une liste apparaisse, nous appuyons sur "Intégrer dans le projet STEP 7". Comme indique la figure 4.7.

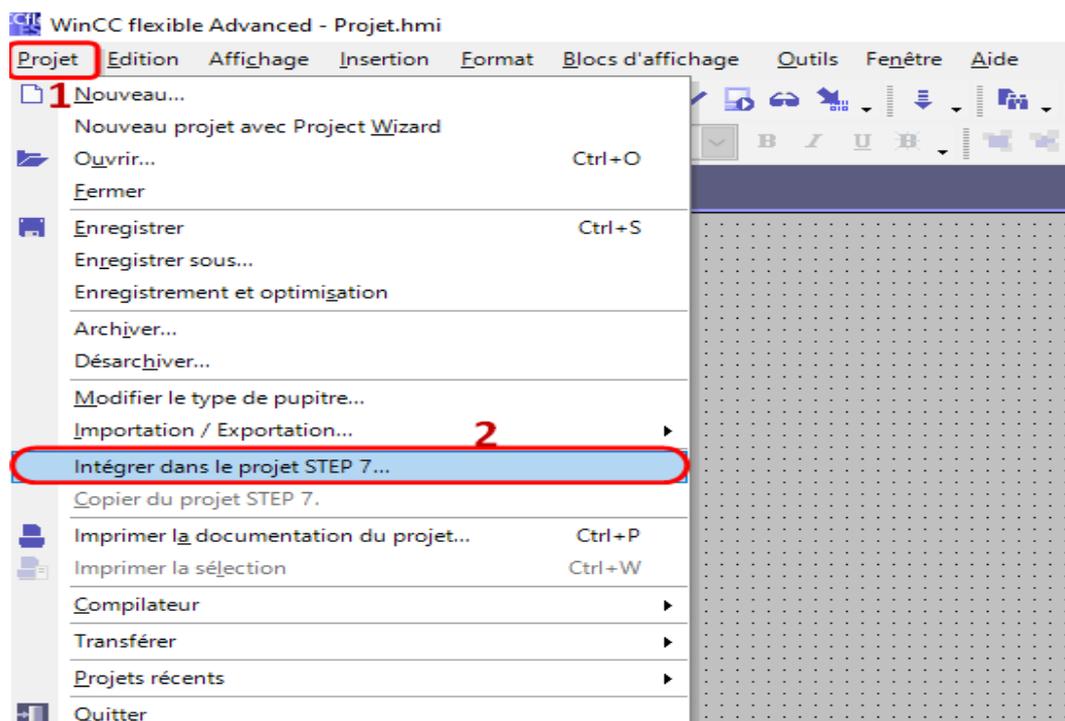


FIG. 4.7 : L'intégration d'un projet WinCC Flexible à SIMATIC Manager.

Après avoir cliqué sur "Intégrer dans le projet STEP 7", une fenêtre s'affiche pour choisir le projet STEP 7 que nous devons lier avec ce projet de WinCC flexible, et nous cliquons sur OK.

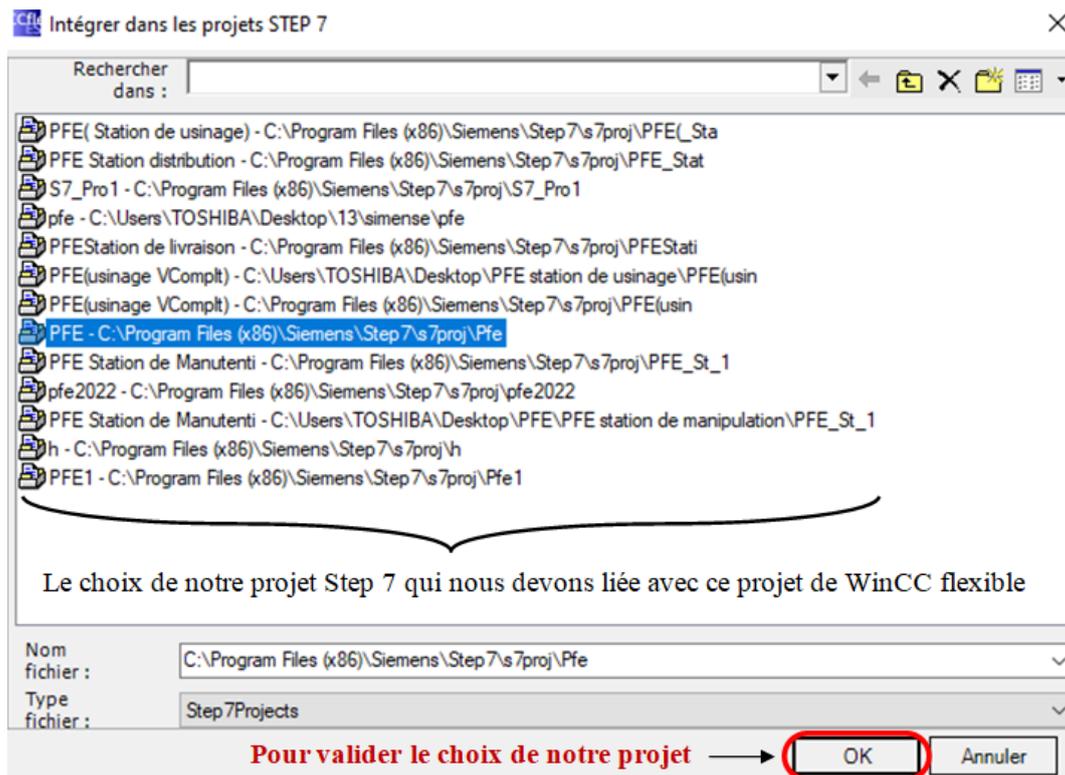


FIG. 4.8 : Le choix de projet Step 7 qui nous devons lier avec WinCC flexible

Nous constatons qu'un pupitre opérateur a été créé pour notre projet STEP 7.

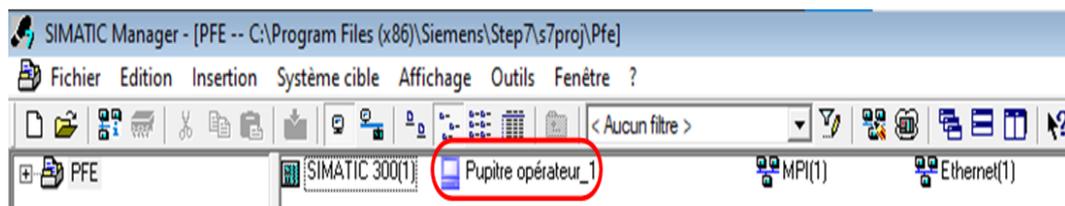


FIG. 4.9 : L'évidence d'un pupitre opérateur dans notre projet Step 7.

4.2.6 Communication SIMATIC et IHM

Il est nécessaire d'établir un lien de communication afin que le système WinCC flexible soit capable d'échanger des données avec l'automate. Dans un projet intégré, les connexions sont créées à l'aide de :

- WinCC flexible.
- NetPro.

Les connexions peuvent être configurées soit avec le logiciel flexible WinCC, soit avec NetPro.

✓ Communication à l'aide de NetPro

Nous devons d'abord ouvrir notre projet sur STEP 7, puis en cliquant sur "Pupitre opérateur", pour éditer la communication, nous appuyons sur "Configuration".

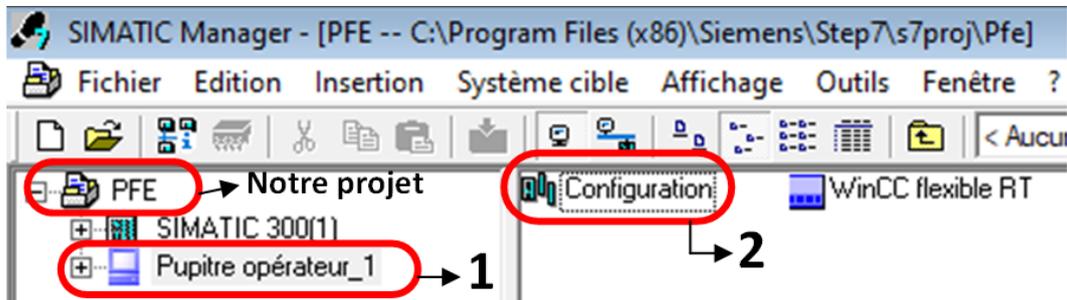


FIG. 4.10 : Cliquons sur "Configuration".

Après avoir cliqué sur "Configuration", la fenêtre (HW Config) s'ouvre. Dans cette fenêtre, nous allons choisir l'interface de communication. Dans notre cas, nous avons choisi la communication Ethernet, pour ajouter cette communication, nous cliquons sur Station PC SIMATIC, puis nous choisissons CP industriel Ethernet, puis nous faisons glisser CP1411 dans le rack (0) HMI. Comme le montre la figure 4.11.

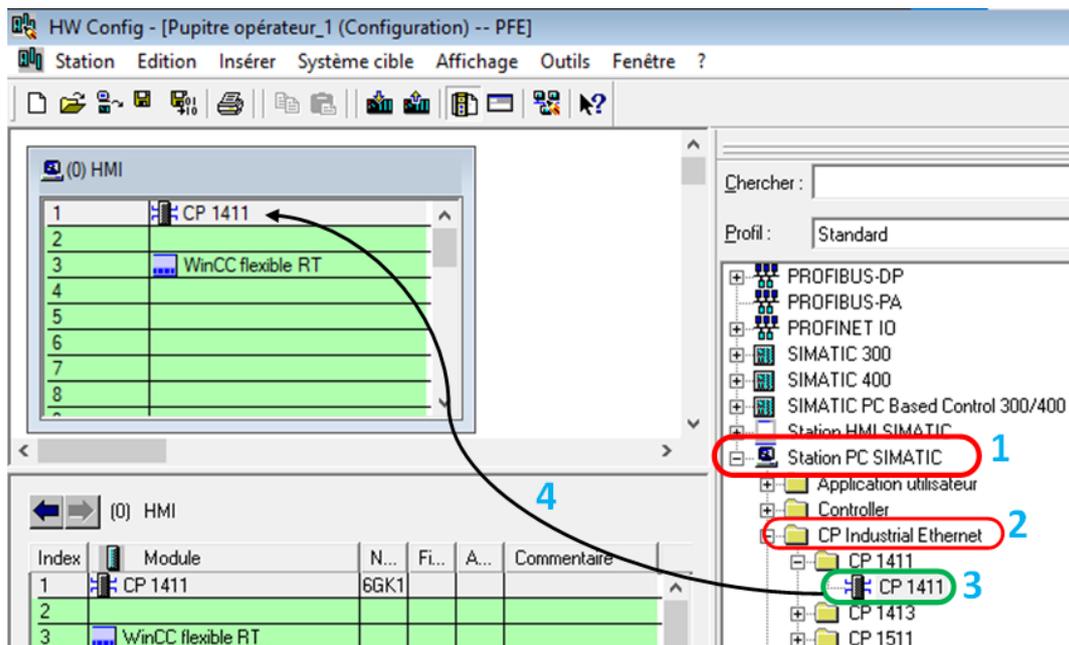


FIG. 4.11 : Configuration du pupitre de commande.

Une fenêtre d'interface Ethernet s'affiche, qui permet de configurer le réseau. Donc, nous cliquons sur "Nouveau" et nous changeons l'adresse IP, une autre fenêtre apparaît. Ensuite nous cliquons sur OK, un câble ETHERNET sera ajouté. Nous voyons que la carte CP industriel Ethernet est bien installée.

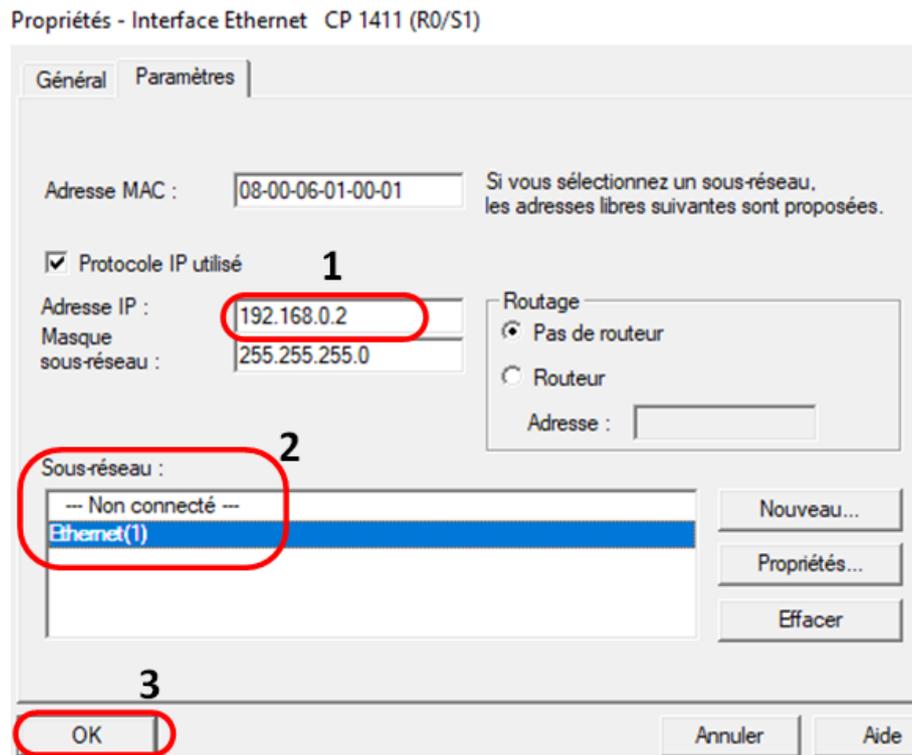


FIG. 4.12 : Fenêtre d'interface Ethernet

Ensuite, nous cliquons sur "enregistrer" et nous compilons. À la fin, nous accédons à la fenêtre de configuration du réseau et nous vérifions la connexion Ethernet entre les deux stations (SIMATIC et HMI). Les deux stations doivent appartenir au même réseau Ethernet.

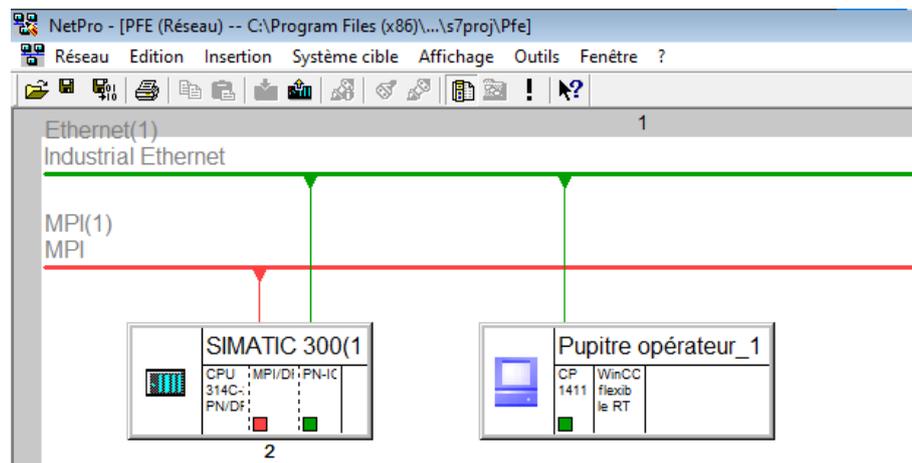


FIG. 4.13 : Établissement de la connexion Ethernet au niveau du Step 7.

✓ Activation de la liaison avec WinCC flexible

Après la configuration avec NetPro, un lien sera créé automatiquement entre STEP 7 et WinCC flexible. Dans ce dernier, nous vérifions cette connexion comme indique la figure 4.14.

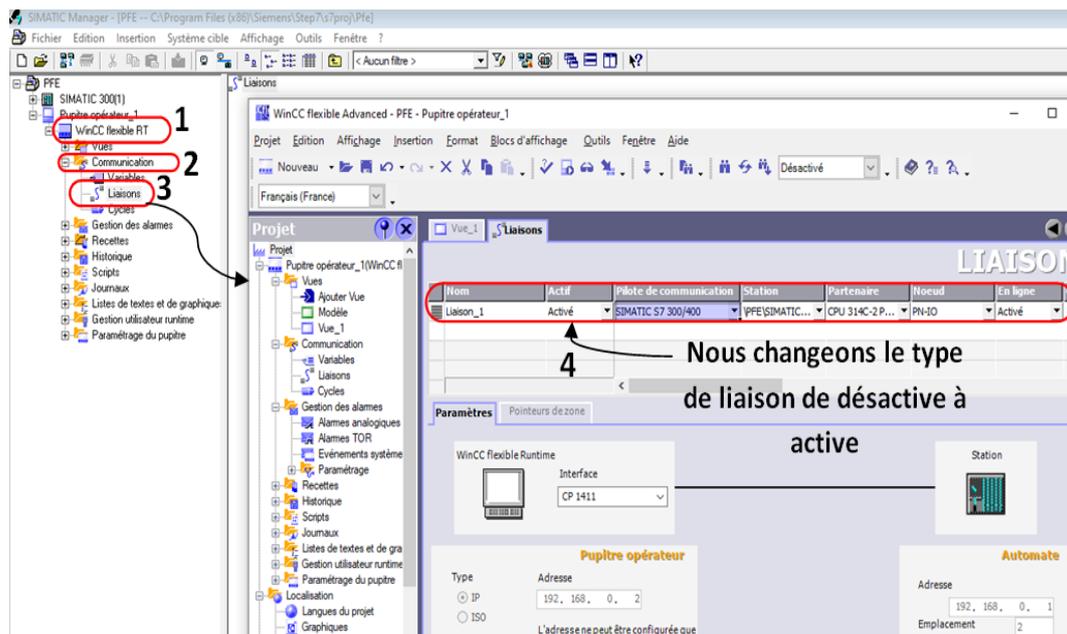


FIG. 4.14 : Activation de la liaison au niveau de WinCC flexible.

4.2.7 Création des vues

Le but de la création d'une vue est de contrôler l'installation d'une machine ou le fonctionnement d'un processus. Pour créer une vue, nous utilisons les différents objets proposés par le logiciel WinCC flexible. Ces objets servent à décrire l'installation du système, à afficher les procédures et à définir les valeurs du processus.

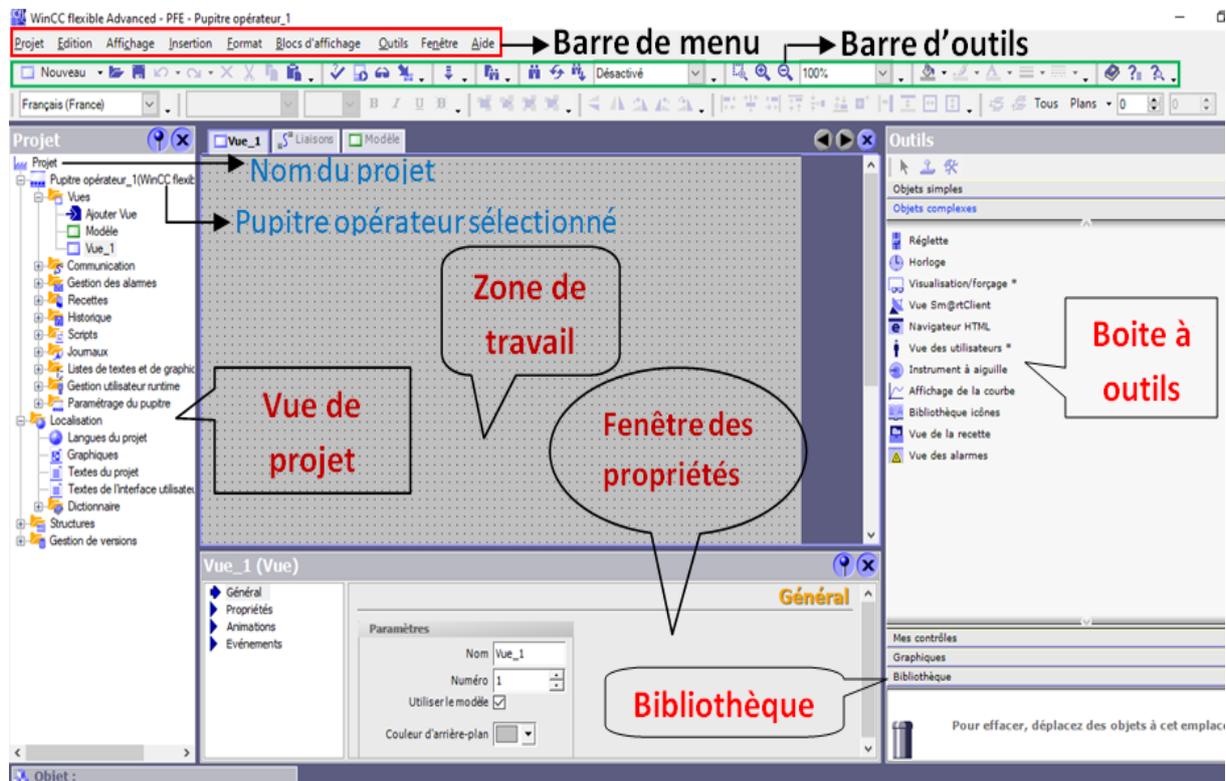


FIG. 4.15 : Fenêtre d'une vue dans le WinCC flexible avec ses divers éléments.

L'interface utilisateur de WinCC flexible se compose des éléments suivants :

- **Barre de menu et d'outils**

Les menus et les barres d'outils nous donnent la possibilité d'accéder à toutes les fonctions accessibles sous WinCC flexible. Lorsque nous positionnons le pointeur de la souris sur une fonction, nous obtiendrons une info-bulle.

- **Espace de travail (zone de travail)**

L'espace de travail est utilisé pour modifier les objets du projet. L'ensemble des éléments du WinCC flexible sont placés autour de l'espace de travail. À l'exception de l'espace de travail, il est possible de placer paramétrer et transférer, par exemple, tous les objets à votre guise.

- **Fenêtre du projet**

Dans la fenêtre du projet, tous les éléments et les éditeurs qui existants dans un projet sont exposés sous forme d'une liste et peuvent être ouverts à partir de cette fenêtre. Sous chaque éditeur se situent des dossiers cachés, qui permettent un stockage structuré des objets. Depuis la fenêtre du projet, nous pouvons accéder aux configurations du pupitre, à la localisation et à la gestion des versions.

- **Fenêtre des propriétés**

La fenêtre des propriétés contient des options qui dépendent du choix actuel dans le champ de travail. Quand un objet a été choisi, nous pouvons modifier les caractéristiques de cet objet dans la fenêtre des propriétés, par exemple la couleur des objets de la vue.

- **Boîte à outils**

Cette fenêtre nous offre un large choix d'objets (entre simples et complexes) à insérer dans les vues, par exemple, bouton, rectangle, cercle, vanne, capteur, moteur, pompe.

- **Bibliothèque**

La bibliothèque est située dans la fenêtre de l'outil. Elle nous permet d'accéder à des objets de vue prédéfinis. En effet, la bibliothèque est l'endroit central où sont stockés les objets couramment utilisés, comme les graphiques et les variables.

4.2.7.1 Utilisation des objets dans la zone de travail

Les objets représentent les éléments graphiques qui nous donnent la possibilité de configurer la disposition des vues concernant le processus du projet. Les différents objets tels que les objets simples (bouton, champ de texte), les objets complexes (capteurs, actionneurs), les graphiques (différents symboles), la bibliothèque, se trouvent dans la boîte à outils. Ces objets sont glissés à la zone de travail, après avoir ajouté un objet dans le champ de travail, il est possible de le modifier en utilisant la fenêtre des propriétés. Les objets créés peuvent être associés à des variables qui ont déjà été définies dans STEP 7.

4.3 Création, conception et simulation de nos IHMs

4.3.1 IHM de la sous-station de distribution

Dans cette sous-station, nous avons travaillé dans un environnement professionnel, car nous utilisons des objets (outils) industriels comme des capteurs, des actionneurs (vérins). Dans cette IHM nous avons représenté les boutons (Start, Reset, Stop) du pupitre de commande par des boutons simple et nous changeons leurs états par l'instruction "Événement". En d'autres termes, n'utilisons pas les mêmes paramètres que Grafcet (nous n'assignerons pas à chaque bouton sa lampe), si nous pressons un bouton par exemple "Reset" celui-ci s'allume. Nous prenons comme exemple le bouton Start.

Nous sélectionnons le bouton Start et cherchons l'instruction "Événements" dans la fenêtre des propriétés. Ensuite, nous choisissons le type d'événement, dans notre cas, nous devons utiliser "Presser", puis nous choisissons la fonction qui se produira lorsque nous appuyerons sur ce bouton. Lorsque nous appuyons sur le bouton Start, il est réglé sur 1, donc nous sélectionnons Positionner-Bit et nous allons donner à ce bouton son nom à partir de la table mnémorique. Ce bouton peut être mis à 1 et remis à 0, donc pour la remise à 0 nous aurons besoin d'un autre événement qui est "Relâcher", ensuite nous choisissons la fonction qui se produira lorsque nous appuyons sur ce bouton, donc nous choisissons RAZ-Bit pour remettre le bouton à 0 et nous allons donner à ce bouton un nom à partir de la table mnémorique.

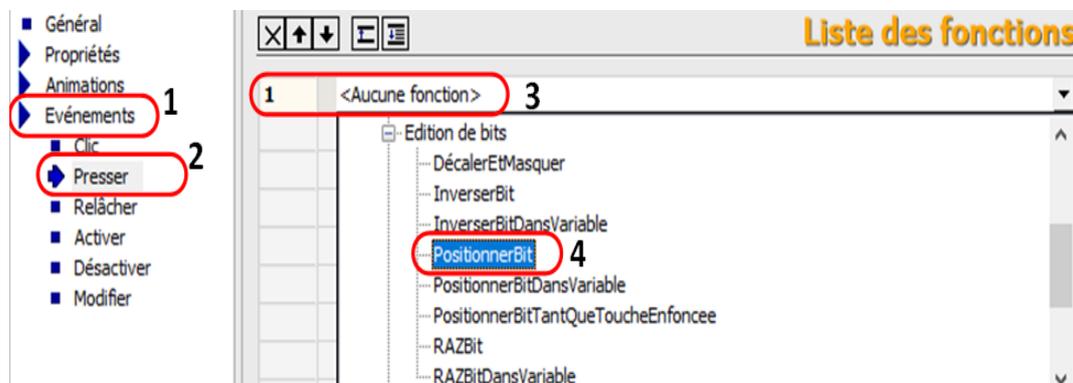


FIG. 4.16 : Configuration des objets (a)

Pour choisir la couleur du bouton Start qui s'affiche lorsque on "Presser" ou "Relâcher" ce bouton, nous cherchons l'instruction "Animations" dans la fenêtre des propriétés, puis nous sélectionnons "Représentation", nous choisissons la variable et son type (Bit) et nous attribuons leurs valeurs avec différentes couleurs, comme le montre la figure 4.17.

Nous représentons les objets réels (actionneurs, capteurs) par leur symbole (soit par les symboles dans la fenêtre des objets complexes, soit par l'image du dispositif depuis la bibliothèque), nous attribuons à chaque dispositif son état de fonctionnement avec leur couleur spécifiée "0 en arrêt avec une couleur rouge", "1 en marche avec une couleur verte".

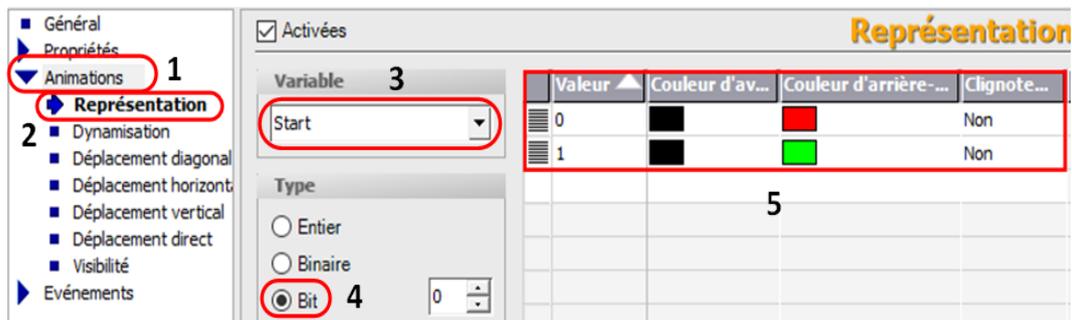


FIG. 4.17 : Configuration des objets (b)

Dans notre IHM de la sous-station de distribution, nous avons utilisé l’instruction de visibilité parce qu’on ne peut pas voir le fonctionnement détaillé d’un système par exemple, la rotation du vérin oscillant de la station suivante vers le magasin ou l’inverse. Cette fonctionnalité nous permet donc de faire apparaître des objets à un instant où une variable est active et les faire disparaître lorsque la variable arrête son exécution. Donc le changement des objets se fait de façon dynamique.

Toutes les procédures que nous faisons dans la sous-station de distribution, nous devons appliquer la même chose dans les autres sous-stations (usage, manipulation, tri).

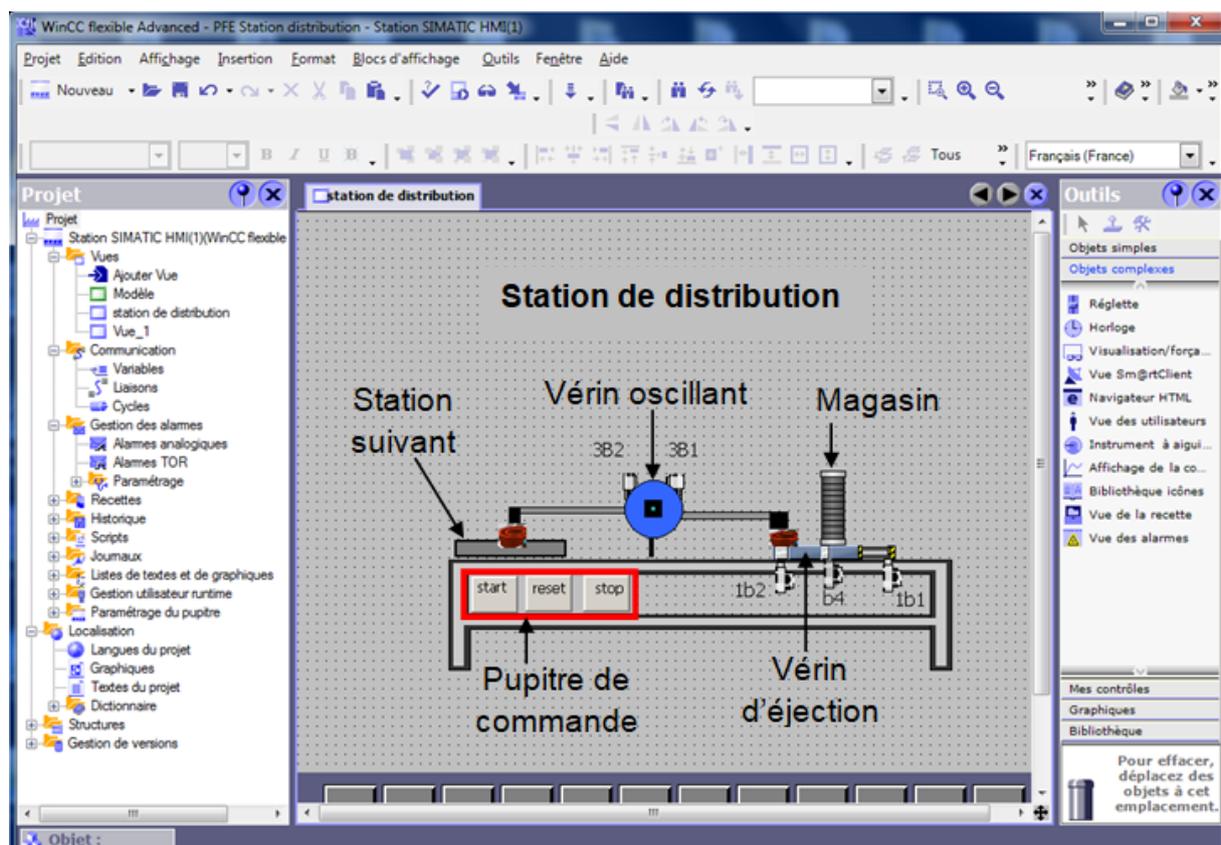


FIG. 4.18 : IHM de la sous station de distribution avant simulation sous Runtime.

4.3.2 IHM de la sous-station d’usage

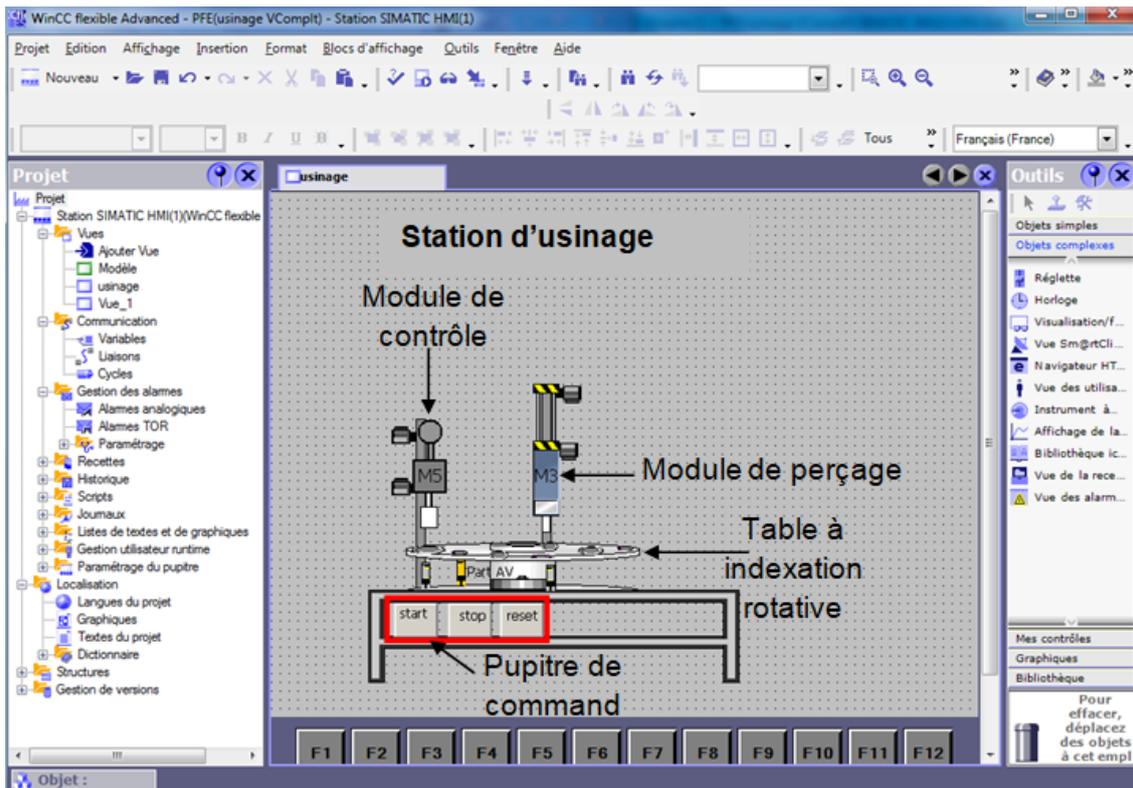


FIG. 4.19 : IHM de la sous-station d'usage avant simulation sous Runtime.

4.3.3 IHM de la sous-station de manipulation

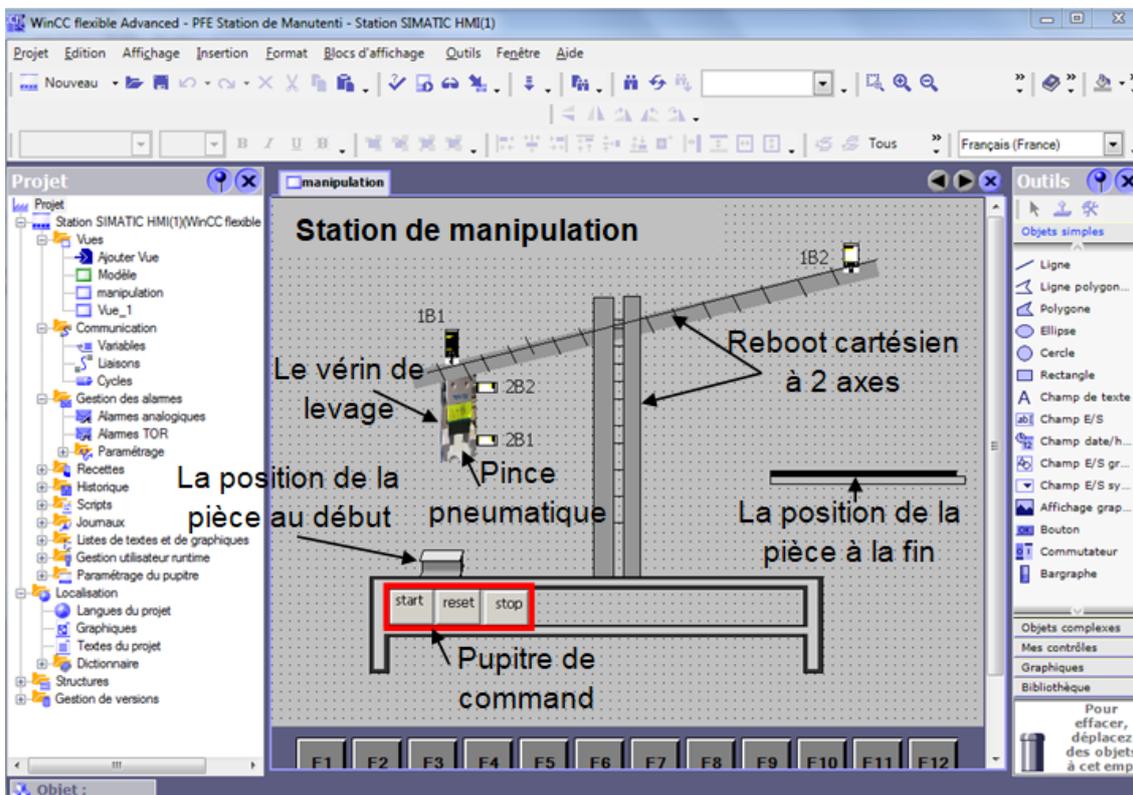


FIG. 4.20 : IHM de la sous-station de manipulation avant simulation sous Runtime.

4.3.4 IHM de la sous-station de livraison (tri)

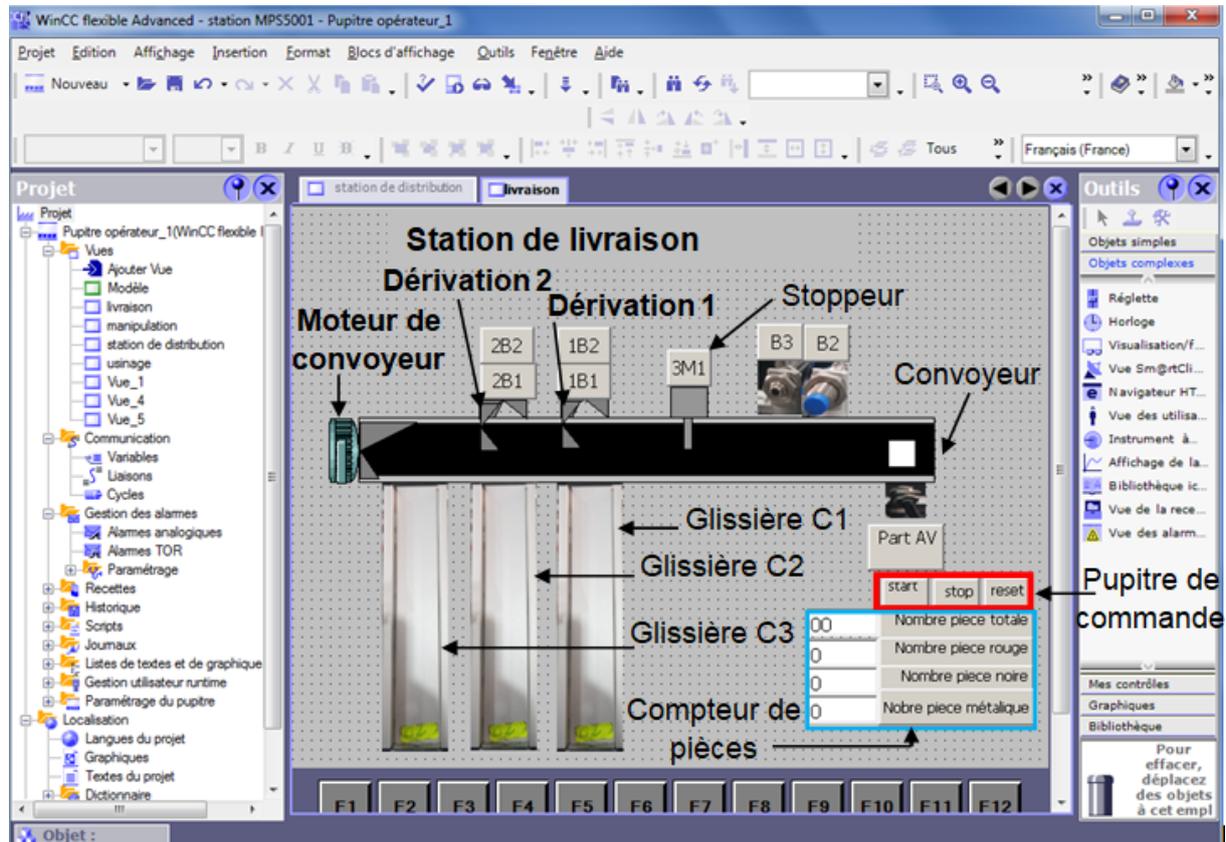


FIG. 4.21 : IHM de la sous-station de livraison avant simulation sous Runtime.

4.4 Simulation des IHMs sous S7-PLCSIM et WINCC FLEXIBLE RUNTIME

Au début, nous lançons le simulateur S7-PLCSIM, puis nous chargeons tous les blocs et le Grafcet du programme (l'IHM ne peut pas être chargé) et nous cliquons sur les boutons de visualisation (icône sous forme de lunettes), puis nous exécutons le simulateur Runtime et nous appuyons sur RUN-P pour voir la simulation de notre IHM.



FIG. 4.22 : L'icône pour lancer le simulateur Runtime

Pour la configuration des boutons (Start, Reset, Stop) dans les IHM que nous allons développer, nous allons utiliser les boutons et non pas les lampes, c'est-à-dire que lorsque nous activons un bouton, il s'allume (Change de couleur).

4.4.1 Sous-station de distribution

Lorsque nous activons le bouton Reset "E 1.3", celui-ci s'allume. Les positions des vérins d'éjection et oscillant reviennent à leur état initial (l'état de ces vérins est également indiqué par ses capteurs). Comme illustré dans l'IHM de la station de distribution de la figure 4.23.

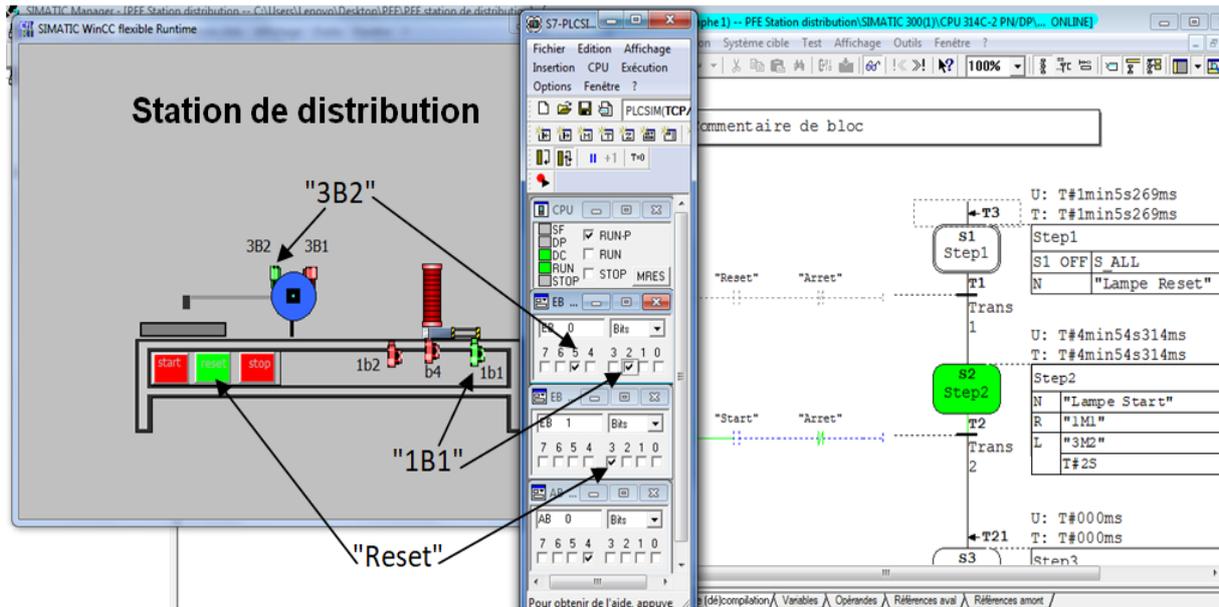


FIG. 4.23 : Simulation de la sous-station de distribution (PLCSIM et Runtime) (a).

Quand nous activons le bouton Start "E 1.0", il s'allume et le cycle de fonctionnement de la station de distribution commence, le vérin d'éjection éjecte une pièce du magasin, puis le vérin oscillant est déplacé en position "magasin". Le vérin pneumatique aspirera cette pièce, une fois que le capteur 2B1 "E 0.3" détecte que la pièce a été aspirée, le vérin oscillant transfère cette pièce à la station suivante.

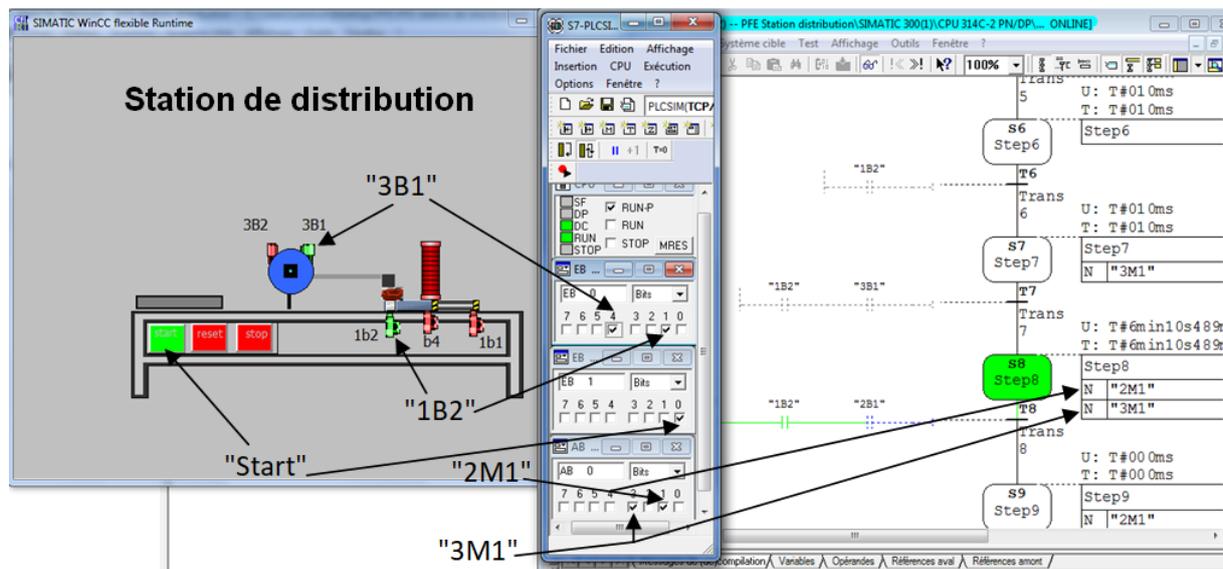


FIG. 4.24 : Simulation de la sous-station de distribution (PLCSIM et Runtime) (b).

Le capteur 2B1 "E 0.3" détecte que la pièce a été aspirée, le vérin oscillant transfère cette pièce vers la station suivante. Lorsqu'il arrive à cette station, l'aspiration de la pièce par le vérin va durer quelques secondes le temps que vérin oscillant se stabilise, puis il lance cette pièce à l'usage.

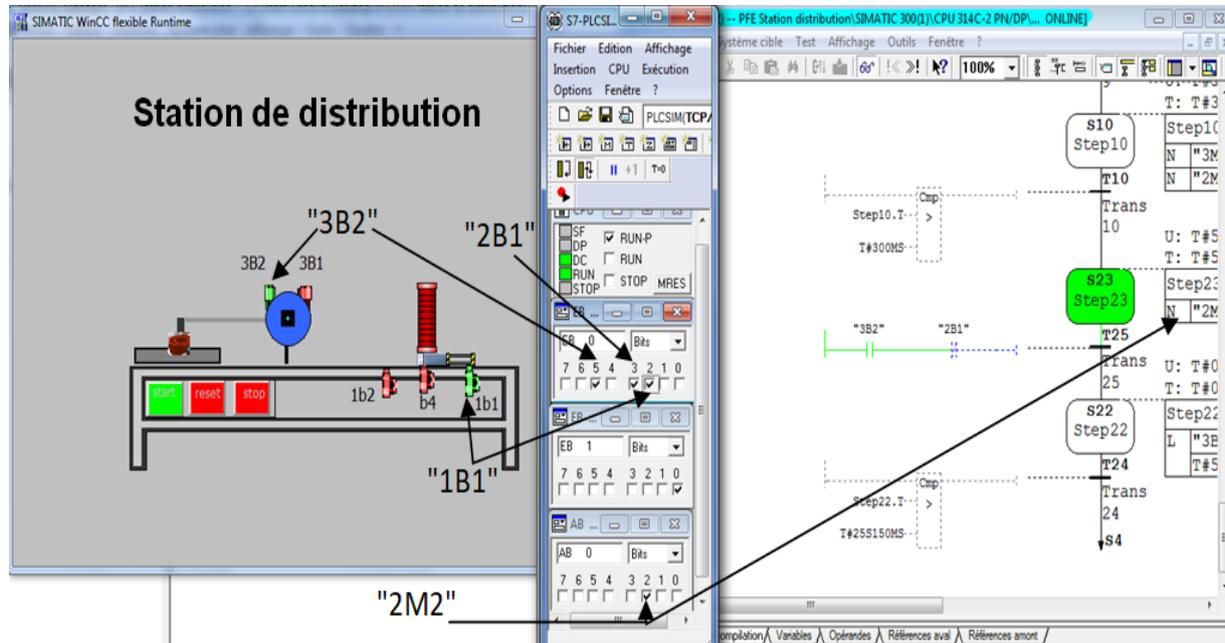


FIG. 4.25 : Simulation de la sous-station de distribution (PLCSIM et Runtime) (c).

Si nous activons le bouton d'arrêt "E 1.1", celui-ci s'allume. Le système s'arrête directement et revient à l'étape initial (Step 1). Après l'arrêt du système, il attend l'enfoncement du bouton Reset "E 1.3" pour effectuer la réinstallation du système.

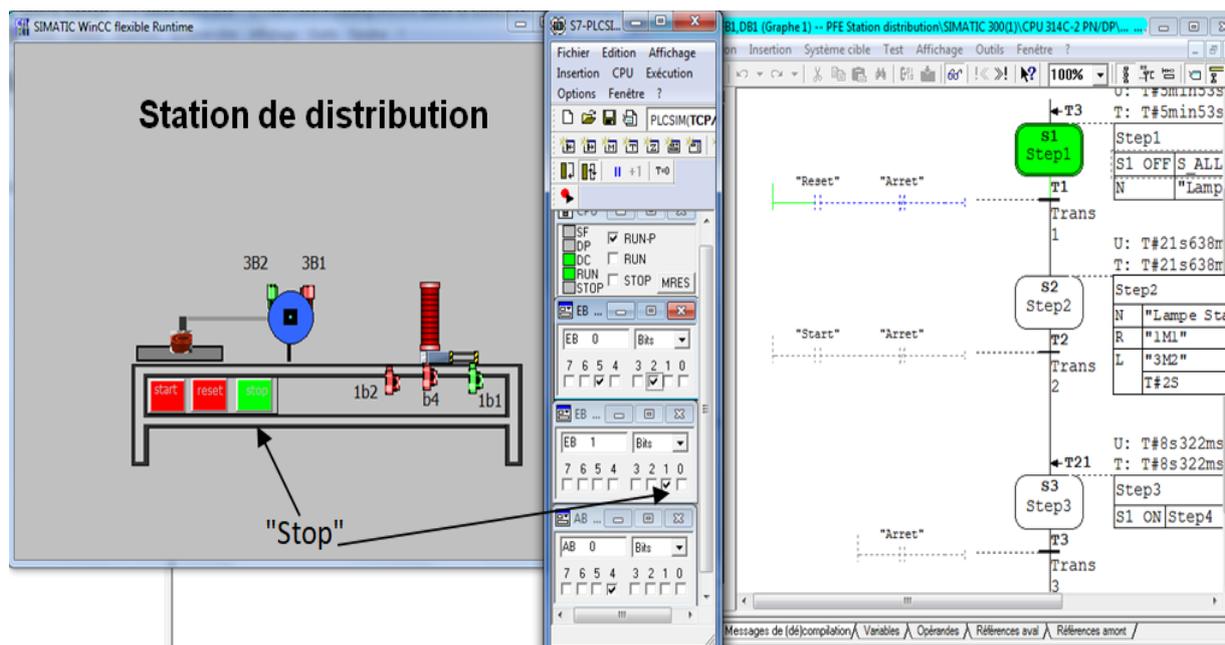


FIG. 4.26 : Simulation de la sous-station de distribution (PLCSIM et Runtime) (d).

4.4.2 Sous-station d'usinage

Lorsque nous activons le bouton Reset "E 1.3", celui-ci s'allume. La position des différents éléments utilisés dans cette station (table à indexation rotative, module de contrôle, module de perçage, module de serrage) revient à leur état initial (l'état de ces éléments est également indiqué par ses capteurs). Comme illustré dans l'IHM de la sous-station d'usinage de la figure 4.27.

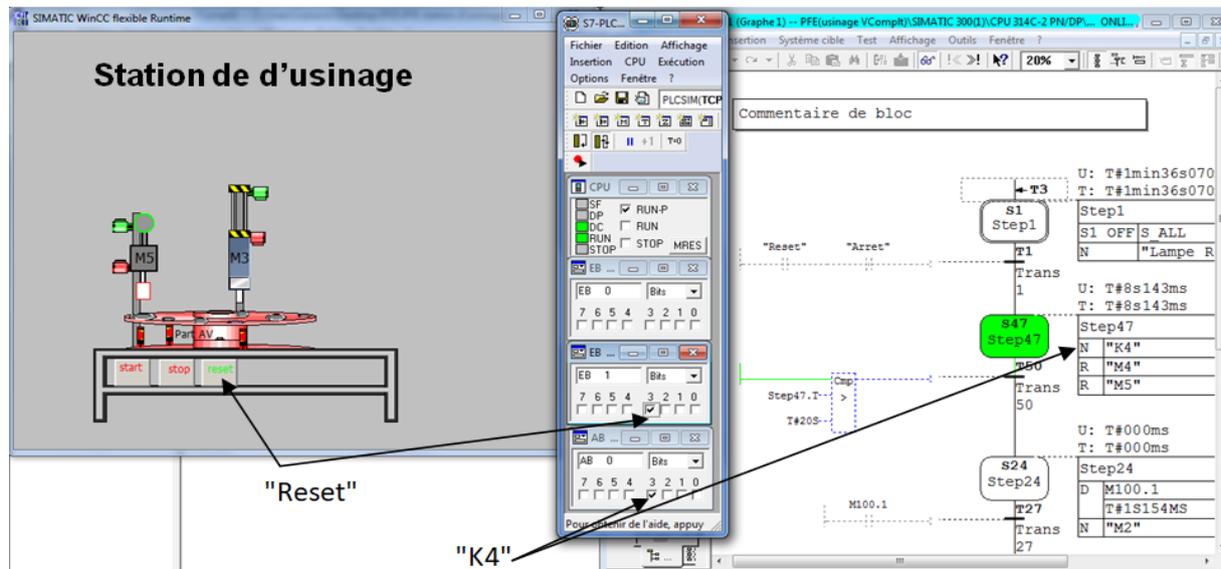


FIG. 4.27 : Simulation de la sous-station d'usinage (PLCSIM et Runtime) (a).

Lorsque nous activons le bouton Start "E 1.0", celui-ci s'allume. Si la pièce est détectée par le capteur Part-AV "E 0.0" le moteur de la table d'indexation (M2 "A 0.1") est mis en marche et la table d'indexation va tourner pendant un temps déterminé jusqu'à atteindre la position de contrôle.

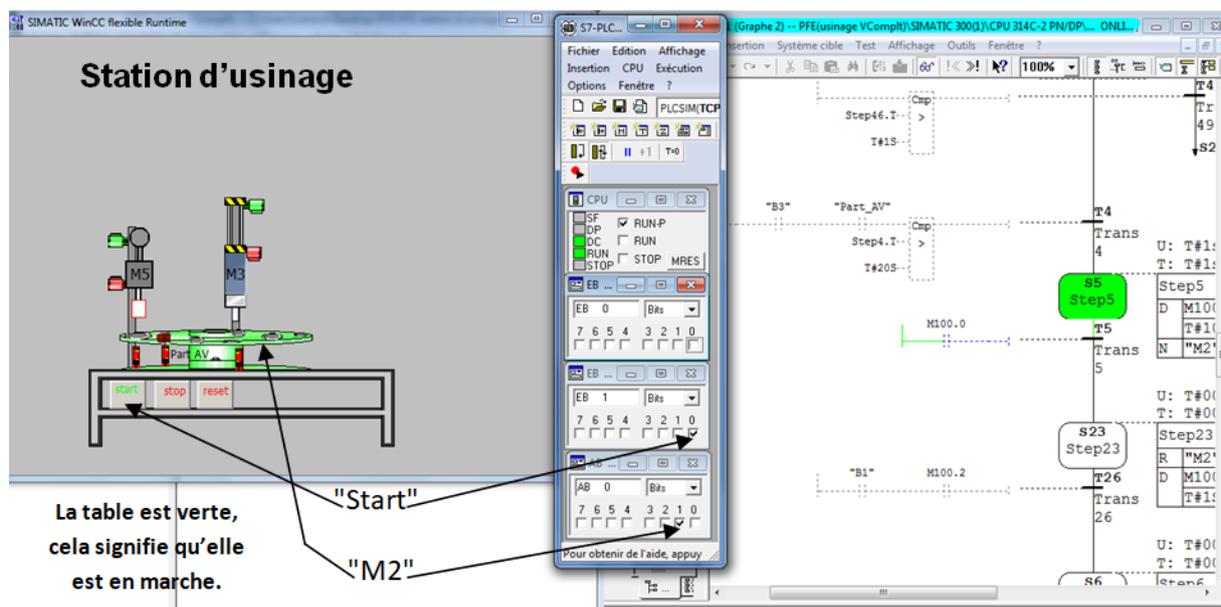


FIG. 4.28 : Simulation de la sous-station d'usinage (PLCSIM et Runtime) (b).

Quand la pièce se trouve dans l'unité de contrôle indiquée par le capteur B1 "E 0.2", la table d'indexation s'arrête, l'induit de la bobine de contrôle M5 "A 0.5" descend et contrôle la pièce.

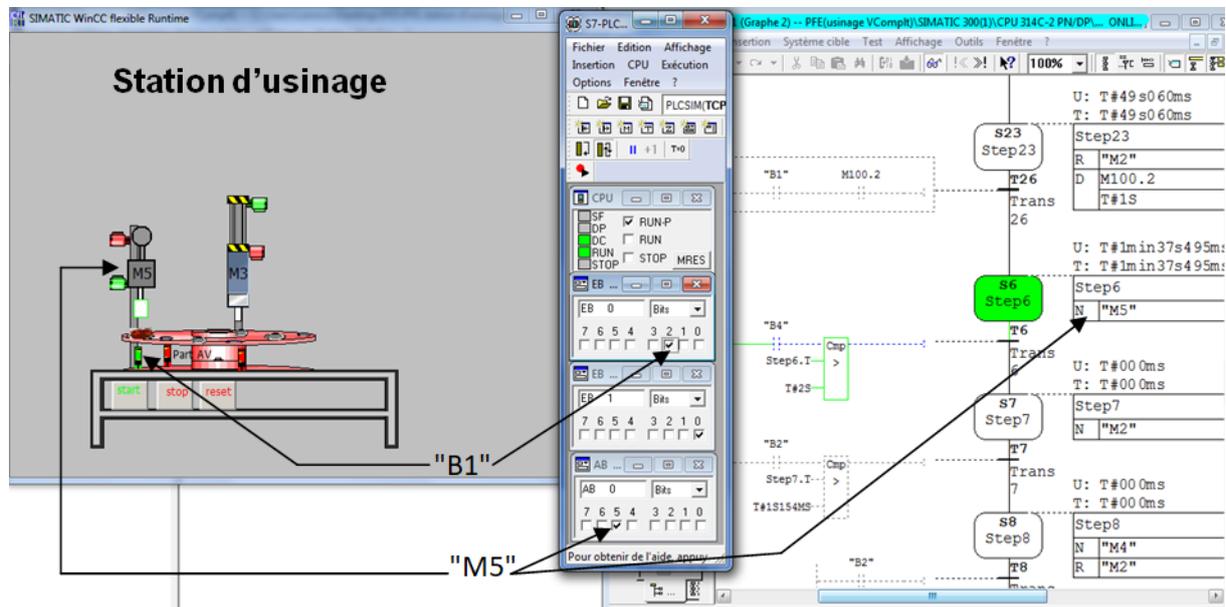


FIG. 4.29 : Simulation de la sous-station d'usinage (PLCSIM et Runtime) (c).

Quand l'unité de contrôle termine la vérification de la pièce, la table à indexation tourne jusqu'à atteindre l'unité de perçage, la présence de la pièce dans l'unité de perçage indiquée par le capteur B2 "E 0.1", la table à indexation s'arrête. Ensuite, le dispositif de serrage M4 "A 0.4" serre la pièce. Le moteur du perçage "A 0.0" est mis en marche et la perceuse se déplace vers le bas K3 "A 0.2" suivant l'axe linéaire.

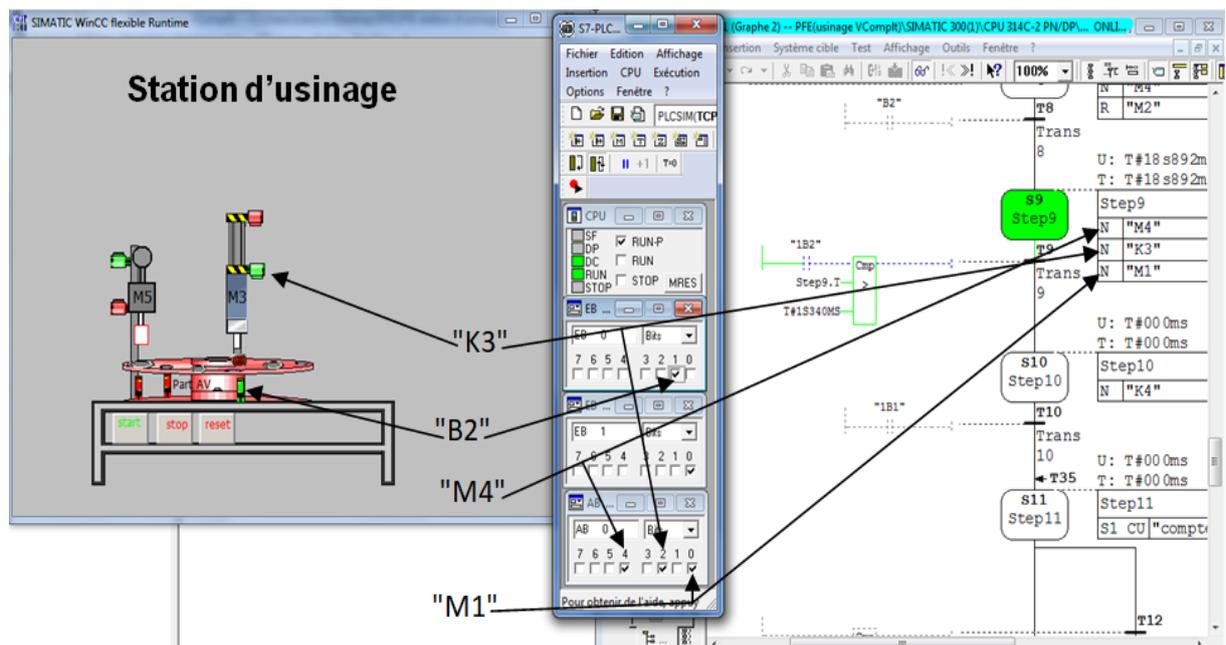


FIG. 4.30 : Simulation de la sous-station d'usinage (PLCSIM et Runtime) (d).

Lorsque l'unité de perçage termine sa tâche, la table à indexation tourne vers l'unité de contrôle pour effectuer le traitement de la pièce une deuxième fois.

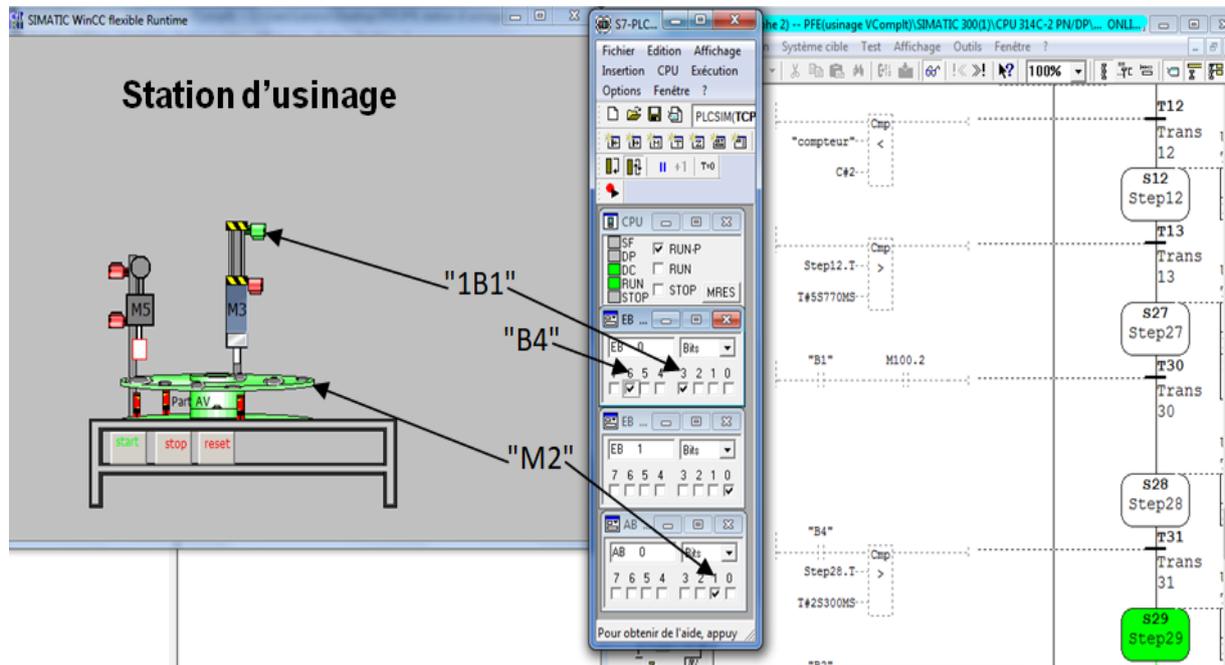


FIG. 4.31 : Simulation de la sous-station d'usinage (PLCSIM et Runtime) (e).

Si nous activons le bouton d'arrêt "E 1.1", celui-ci s'allume. Le système s'arrête directement et revient à l'état initial (Step 1). Après l'arrêt du système, ce dernier attend l'appuie sur le bouton Reset "E 1.3" pour effectuer la réinitialisation du système.

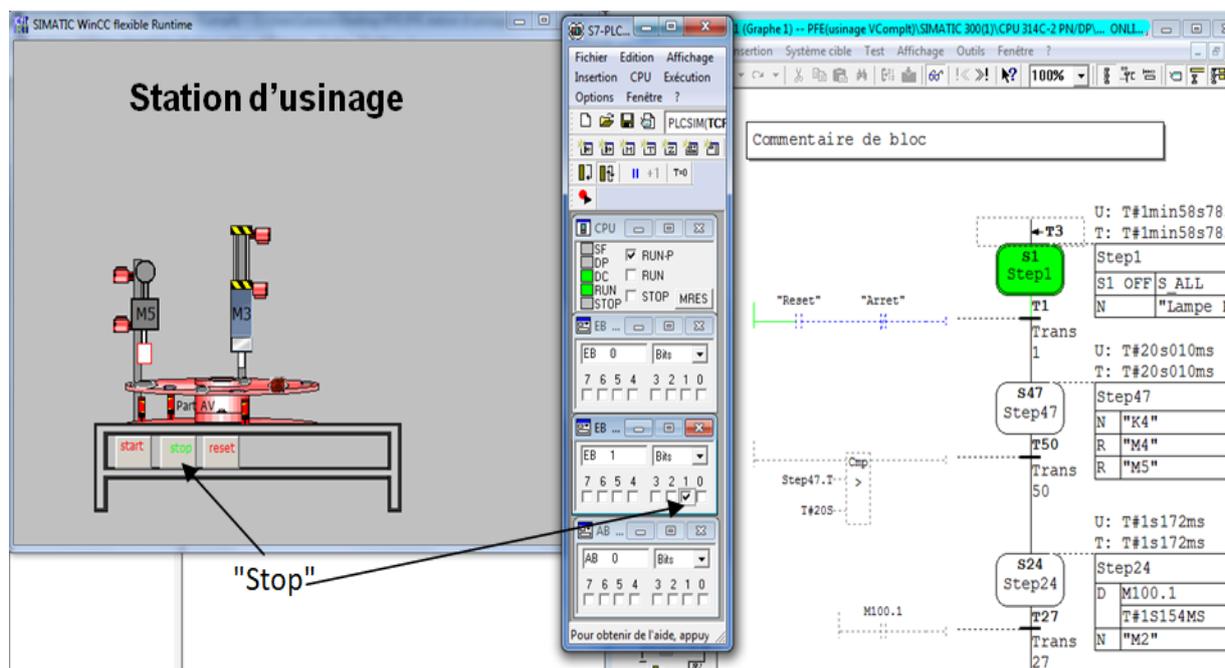


FIG. 4.32 : Simulation de la sous-station d'usinage (PLCSIM et Runtime) (f).

4.4.3 Sous-station de manipulation

Après avoir activé le bouton Reset "E 1.3", celui-ci va s'allumer. Les différents éléments utilisés dans cette station (l'axe linéaire "Robot cartésien", le vérin de levage et la pince) reviennent à leurs états initiaux (les états de ces éléments sont indiqués par les capteurs). Comme indiqué dans la figure 4.33 de l'IHM de la sous-station de manipulation.

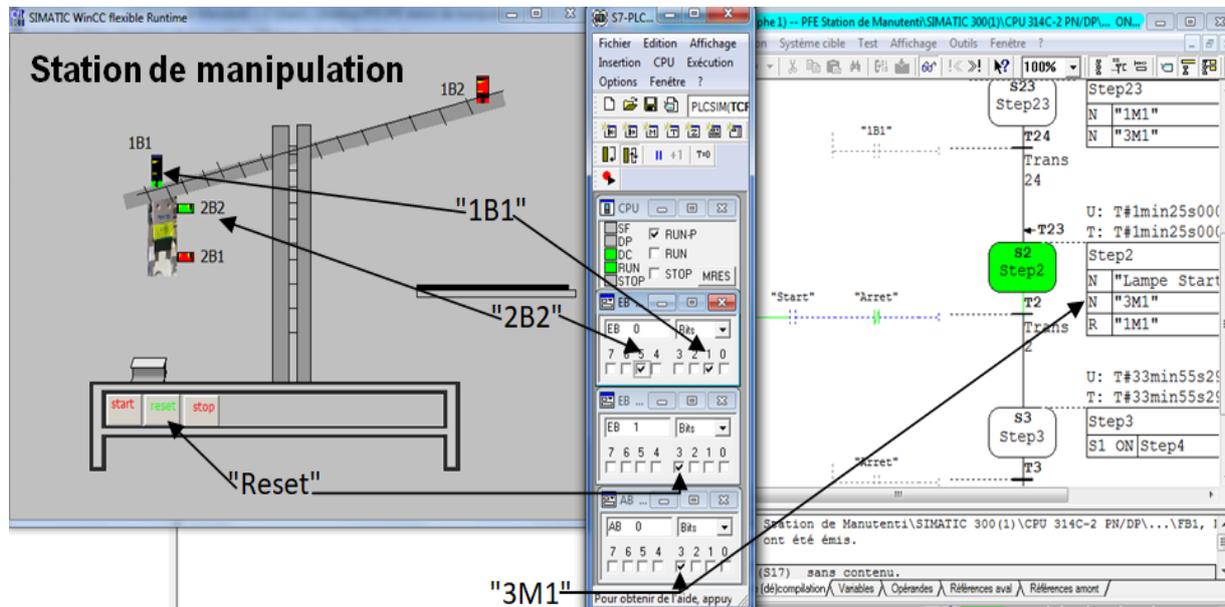


FIG. 4.33 : Simulation de la sous-station de manipulation (PLCSIM et Runtime) (a).

Lorsque nous activons le bouton Start "E 1.0", celui-ci s'allume et si une pièce est identifiée dans la position de prélèvement de la pièce, le vérin de levage sort 2M1 "A 0.2", la pince saisit la pièce et se ferme.

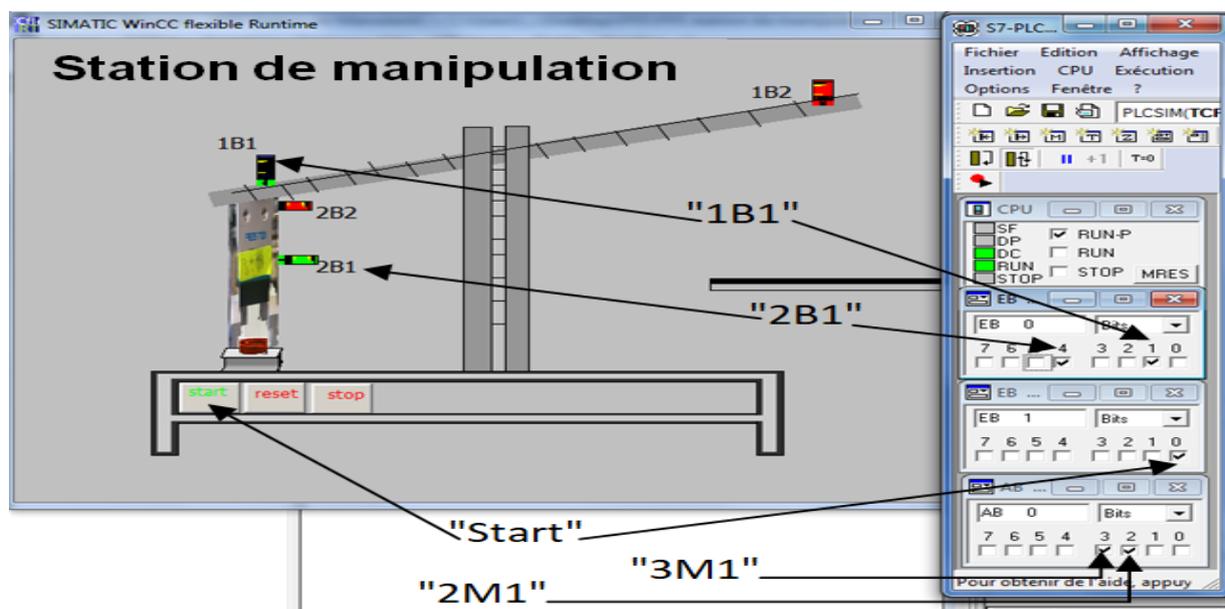


FIG. 4.34 : Simulation de la sous-station de manipulation (PLCSIM et Runtime) (b).

Quand la pince saisit la pièce et se ferme, le vérin de levage rentre, le manipulateur 1M2 "A 0.1", se déplace vers la station suivante.

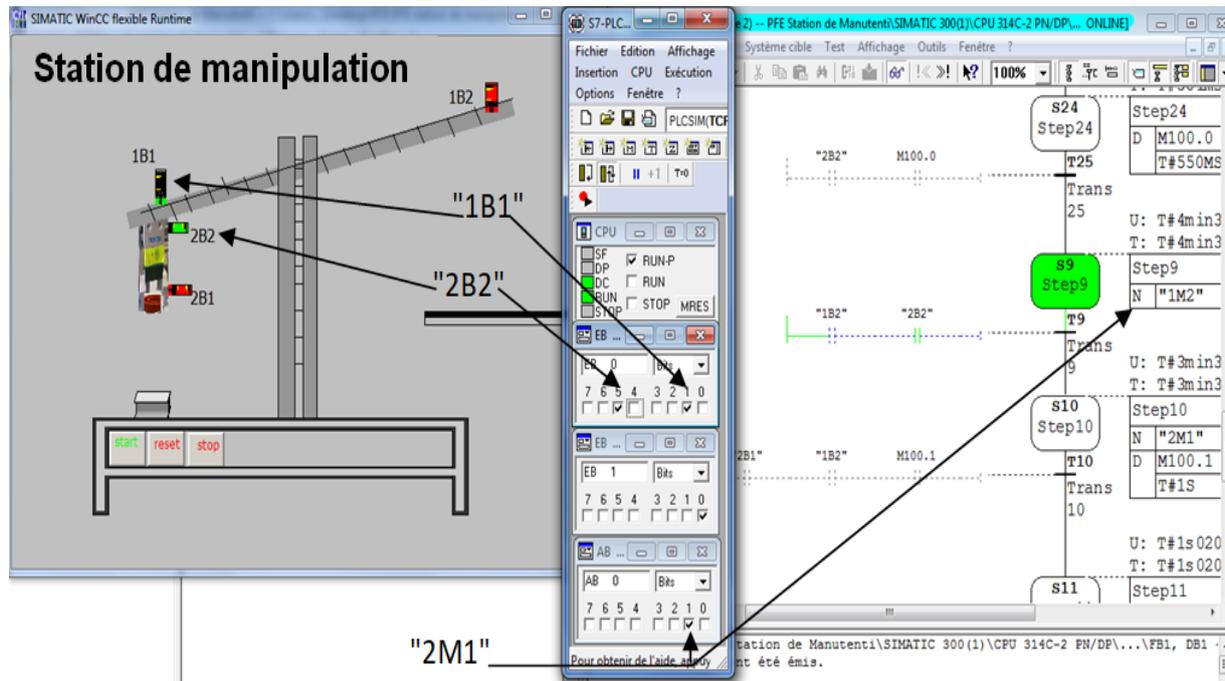


FIG. 4.35 : Simulation de la sous-station de manipulation (PLCSIM et Runtime) (c).

Quand le manipulateur 1M2 "A 0.1" arrive à la station suivante, il attend un instant puis le vérin de levage descend.

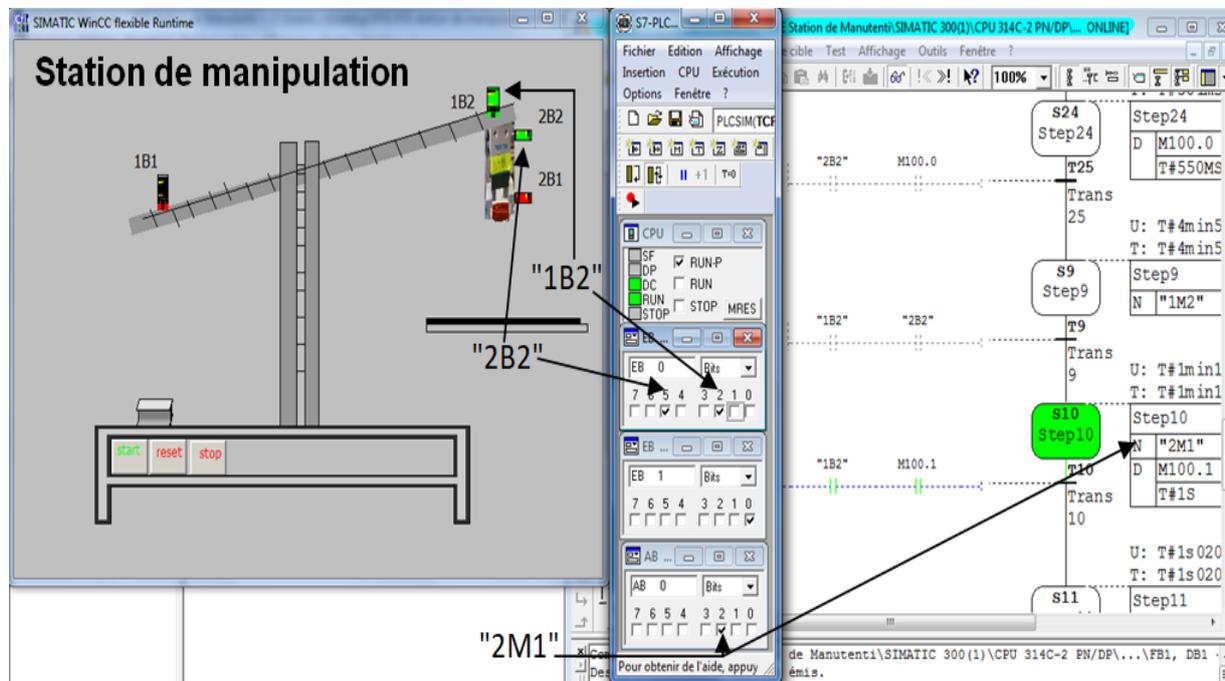


FIG. 4.36 : Simulation de la sous-station de manipulation (PLCSIM et Runtime) (d).

Le vérin de levage sort puis il attend un instant afin qu'il se stabilise, puis la pince s'ouvre et libère la pièce. Après avoir éjecté la pièce, le vérin de levage se rétracte.

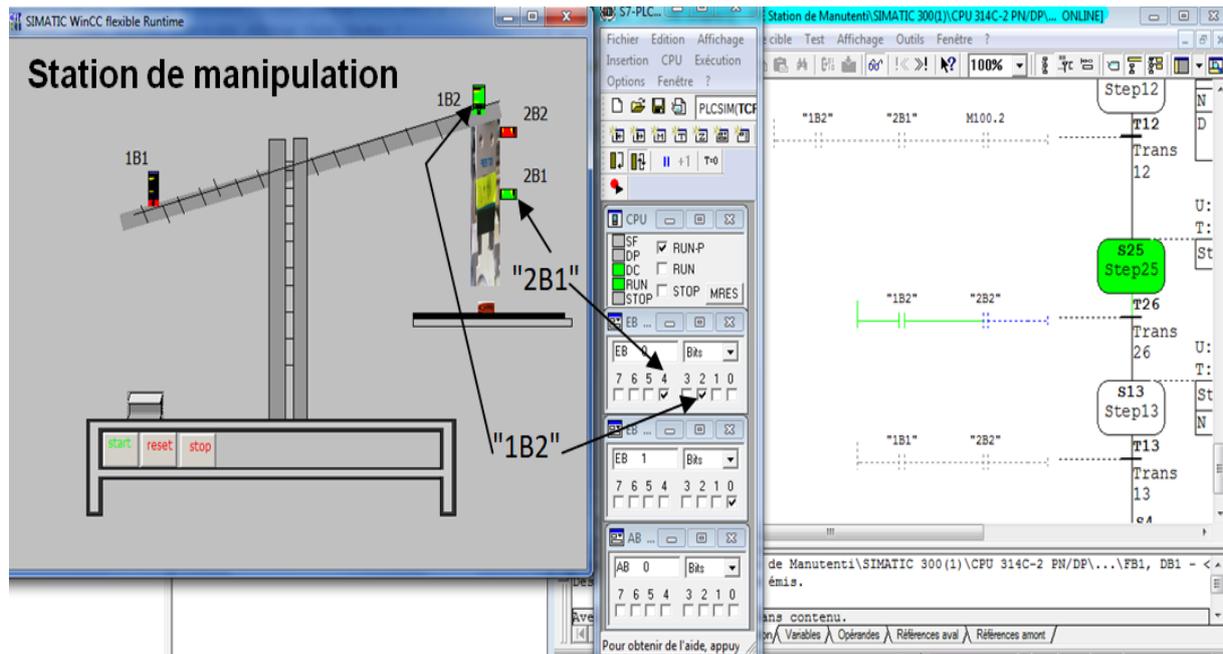


FIG. 4.37 : Simulation de la sous-station de manipulation (PLCSIM et Runtime) (e).

Si nous activons le bouton d'arrêt "E 1.1", celui-ci s'allume. Le système s'arrête directement et revient à l'état initial (Step 1). Après l'arrêt du système, ce dernier attend l'action sur le bouton Reset "E 1.3" pour effectuer la réinitialisation du système.

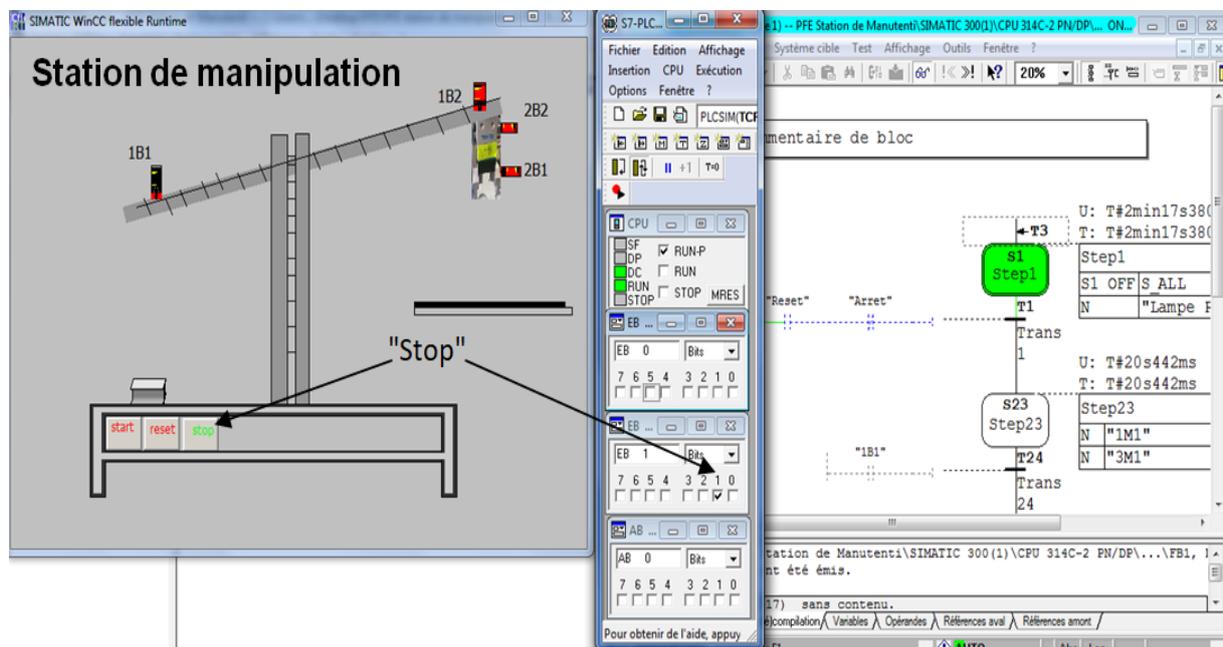


FIG. 4.38 : Simulation de la sous-station de manipulation (PLCSIM et Runtime) (f).

4.4.4 Sous-station de livraison (tri)

Quand nous activons le bouton Reset "E 1.3", celui-ci s'allume. La position des différents éléments utilisés dans cette station (moteur de convoyeur, les deux dériviations,

stoppeur) revient à leur état initial (l'état de ces éléments est également indiqué par les capteurs). Comme le montre l'IHM de la sous-station de livraison (trie) de la figure 4.39.

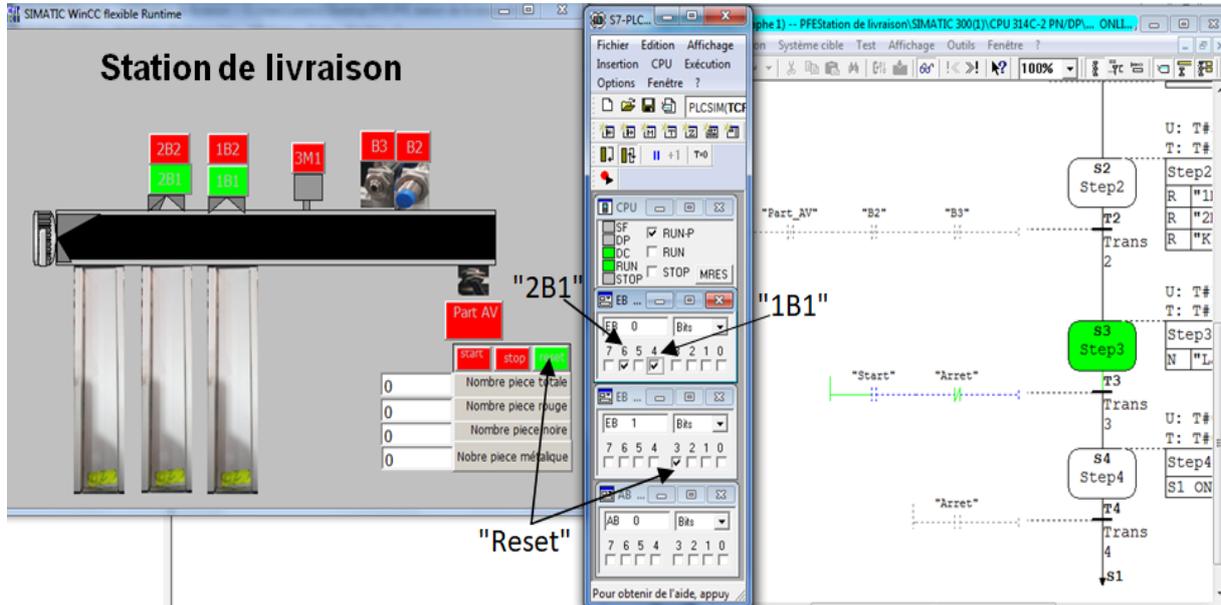


FIG. 4.39 : Simulation de la sous-station de livraison (PLCSIM et Runtime) (a).

Quand nous activons le bouton Start "E 1.0", celui-ci s'allume et si une pièce est identifiée par le capteur Part-AV "E 0.0", le moteur de la bande s'active K1 "A 0.0". Une fois que la pièce arrive aux capteurs d'identification de couleur/matière, elle sera bloquée par le stoppeur. Si une pièce noire est identifiée, le stoppeur se rétracte et permet à la pièce noire de passer, la dérivation 2 2M1 "A 0.2", fera glisser la pièce dans la glissière au milieu de la bande puis se rétractera. Nous constatons que les différents compteurs commencent à compter les pièces.

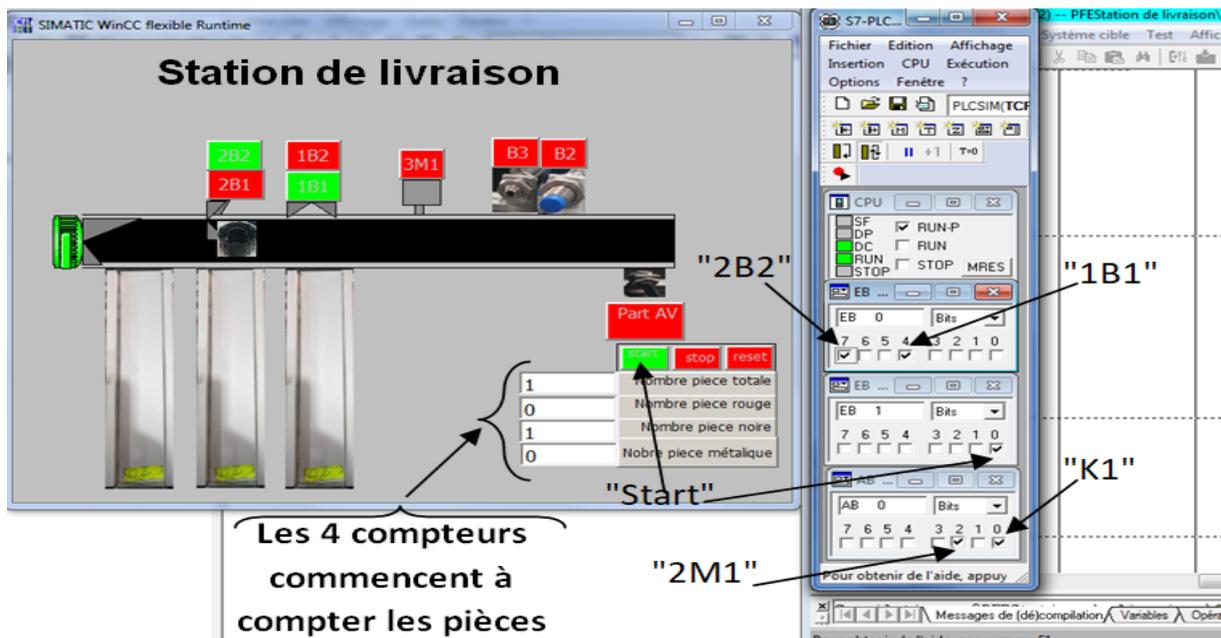


FIG. 4.40 : Simulation de la sous-station de livraison (PLCSIM et Runtime) (b).

Lorsque la pièce métallique est identifiée, le stoppeur se rétracte et permet à la pièce de passer, la dérivation 1 1M1 "A 0.1" sort pour faire glisser la pièce métallique dans la glissière au début de la bande puis se rétractera.

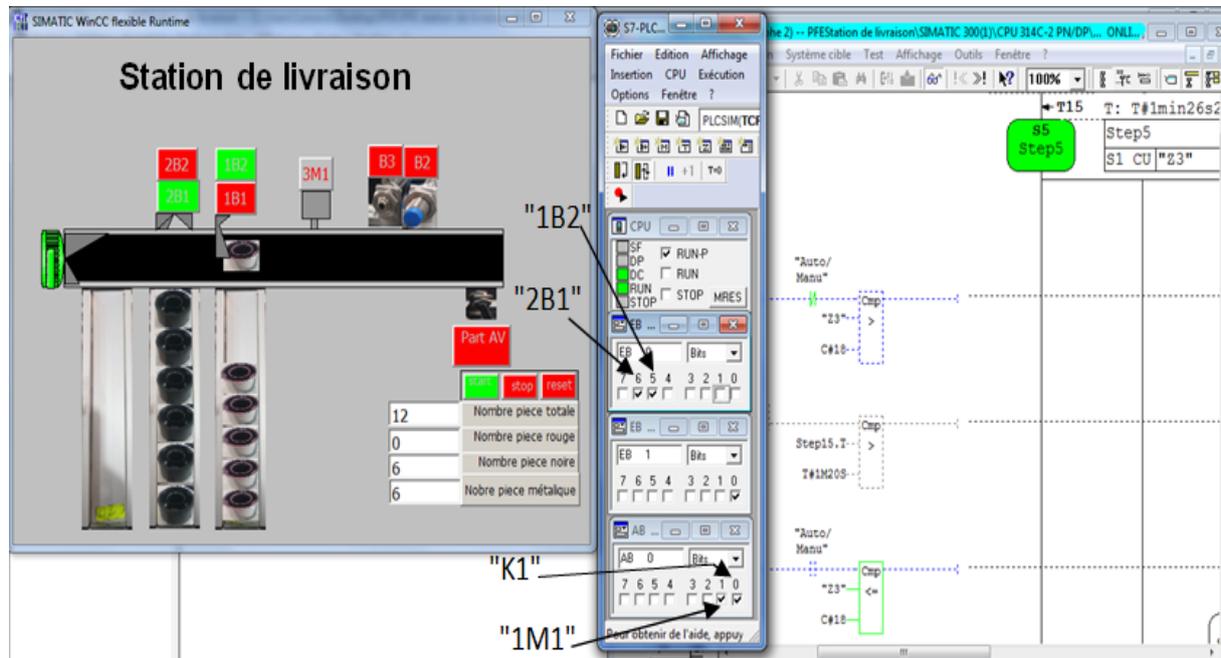


FIG. 4.41 : Simulation de la sous-station de livraison (PLCSIM et Runtime) (c).

Nous pouvons voir sur la figure 4.42 que les deux premières glissières C1 et C2 sont pleines et C3 vide. Donc, il ne nous reste plus que la glissière C3 à remplir. Quand la pièce rouge est identifiée, le stoppeur se rétracte et permet à la pièce de passer jusqu'à ce qu'elle arrive à la fin de la bande, où la pièce rouge glissera dans la dernière glissière.



FIG. 4.42 : Simulation de la sous-station de livraison (PLCSIM et Runtime) (d).

Dans la figure 4.43 nous voyons que les trois glissières C1, C2 et C3 sont pleines, une fois que la 19ème pièce est présente devant le capteur Part-AV "E 0.0", le système s'arrête pendant un temps d'attente (Temps de déchargement des trois glissières), afin de vider les trois glissières. Dès que les trois glissières sont vides le système démarre pour un nouveau cycle de fonctionnement.

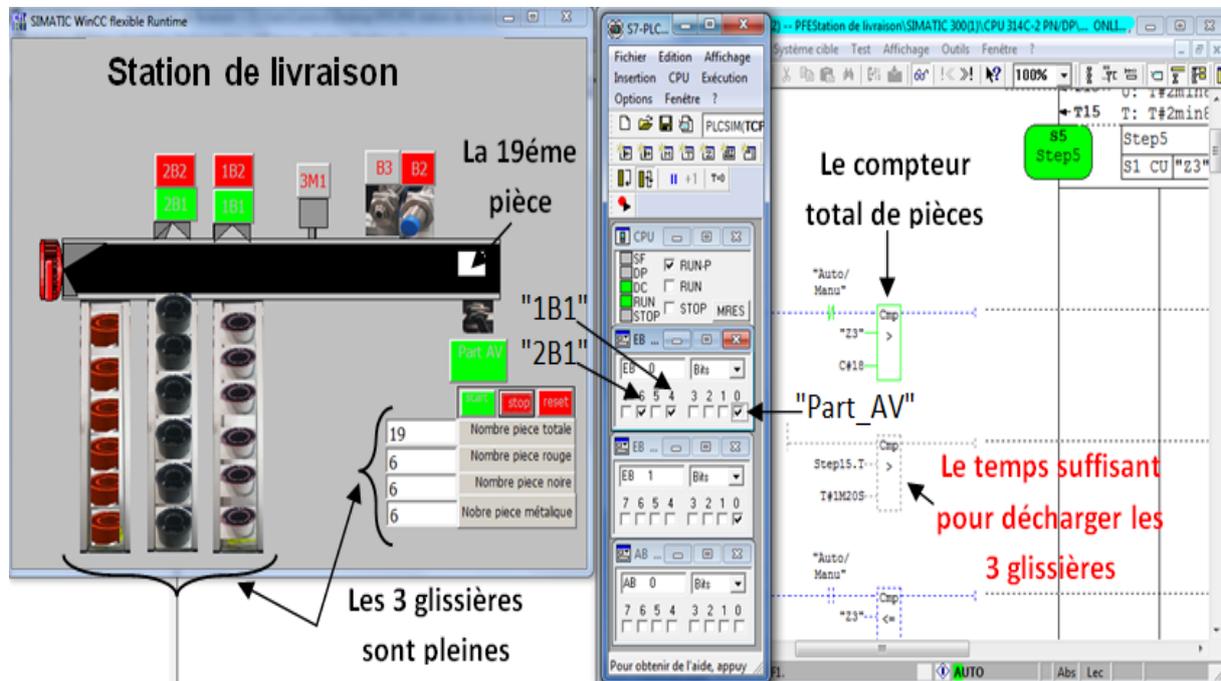


FIG. 4.43 : Simulation de la sous-station de livraison (PLCSIM et Runtime) (e).

4.5 Conclusion

Dans ce chapitre, nous avons donné une idée sur l'outil WinCC flexible et son simulateur Runtime. Par la suite, nous avons présenté les différentes étapes à suivre pour configurer la connexion entre nos pupitres opérateur et le projet (programme) STEP7. Puis, nous avons montré les différentes étapes de la création des IHMs.

Dans la dernière partie de ce chapitre, nous avons exposé les résultats obtenus après le développement d'une IHM pour chaque sous-station avec une synchronisation entre le simulateur PLC-SIM et WinCC Runtime. Les IHM ont suivi exactement le fonctionnement réel des sous stations, où chaque IHM a indiqué l'état exact et en temps réel de la sous-station qui lui correspond.

CONCLUSION GÉNÉRALE

Avec le développement des nouvelles technologies, l'avancée énorme et rapide de l'informatique et avec l'industrie 4.0 et son utilisation de l'intelligence artificielle et des objets connectés, les entreprises industrielles utilisent de plus en plus les services internet et les réseaux informatiques afin de se connecter avec leurs filiales ou leurs représentants à distance pour commander, superviser, maintenir et gérer les processus automatisés reposant majoritairement sur des architectures de contrôle/commande distribuées. Pour cela la majorité des systèmes industriels moderne utilisent les automates programmables industriels avec un système de supervision complet de type SCADA ou avec des interfaces homme-machine, contrôlant chacune une ou plusieurs stations.

Notre mémoire est divisé en quatre chapitres, dans le premier, nous avons parlé de manière globale sur les systèmes automatisés de production SAP ou nous avons expliqué la composition des deux parties qui composent de tels systèmes. Nous avons commencé par présenter la partie opérative puis nous avons montré les composants de la partie commande.

Dans le second chapitre, nous avons parlé principalement des systèmes de supervision de manière générale, puis nous avons présenté de manière plus détaillée les interfaces homme-machines où nous avons montré les différentes fonctionnalités des IHMs, les avantages des IHMs, la différence entre l'IHM et le SCADA. Nous avons parlé aussi de la conception des IHMs, des différents types d'IHM et de la connexion des IHMs avec les APIs nous avons clôturé ce chapitre par donnée quelques applications des IHMs.

Le troisième chapitre était consacré à la première partie de notre travail avec lequel nous avons configuré et programmé les automates programmables du système MPS variant que nous avons proposé à partir du MPS original de FESTO. Tout d'abord nous avons présenté les quatre sous-stations qui composent le nouveau système, puis nous avons présenté le nouveau cahier de charge adapté au fonctionnement de ces quatre sous-stations qui forment le MPS variant. Ensuite, nous avons programmé les automates de chaque sous-station via le logiciel STEP 7 vu que les API sont tous de type S7-314-2 DP/PN de SIEMENS. Après la programmation, nous avons simulé le fonctionnement des programmes via S7-PLCSIM. En fin, nous avons connecté le PC avec EasyPort USB qui jouer le rôle d'émulateur de l'API et après injection des programmes dans l'EasyPort USB nous avons

simulé encore le système pour confirmer le bon fonctionnement de nos programmes et valider le travail effectué.

Le dernier chapitre était consacré au développement d'IHMs afin de superviser le système MPS variant proposé. Pour cela nous avons utilisé le logiciel d'ingénierie de conception et de configuration d'IHM, WinCC Flexible. Au début, nous avons présenté le logiciel WinCC Flexible et son interface graphique, puis nous avons expliqué en détail toutes les étapes à suivre pour la création du projet de supervision, comment nous avons créé les IHMs des sous-stations et comment intégrer les programmes STEP 7 avec les projets de WinCC Flexible. À la fin du chapitre, nous avons présenté les simulations des interfaces WinCC Flexible de toutes les sous stations du système MPS variant étudié, où nous avons remarqué que le fonctionnement des IHMs étaient parfait.

Comme perspective à notre travail, nous proposons :

- L'introduction et exploitation des capteurs IP-FI et IP-FO dans les programmes des APIs afin que toutes les sous-stations puissent communiquer entre elles.
- La liaison directe des sous-stations avec le PC de supervision via les moyens de communication pour voir l'exécution des processus sur les IHMs en temps réel et sans passé par S7-PLCSIM et EasyPort USB.
- Le développement d'une seule Interface homme machine qui englobant les quatre sous-stations, tout en gardant les interfaces individuelles de chaque sous-station.

BIBLIOGRAPHIE

- [1] F. LAMB, *Industrial automation : hands-on*. New York : McGraw-Hill Education, 2013, 1 p., ISBN : 978-0-07-181647-2.
- [2] M. KCHAOU, « Cours 2ème année II - Introduction à l'Automatisme GRAFCET & GEMMA- Université de Sousse Institut Supérieur des Sciences Appliquées et de Technologie.pdf »,
- [3] P. LE BRUN, '*Automates programmables industriels*' Lycée Louis ARMAND ,173 Bd de Strasbourg.pdf.
- [4] M. AIDOUD et A. SEBBAGH, « Automatisme industriels. Université 8 mai 1945 – GUELMA (2020).pdf »,
- [5] « Automatisme industriels. AUTOMATE PRO.pdf »,
- [6] A. MTIBAA, « Les Systèmes Automatisés de Production (Ecole Nationale d'Ingénieurs de Monastir. Université de Monastir. Tunis) (2013).pdf »,
- [7] K. TLILI, « Support du cours automatisme industrielle . Institut Supérieur des Etudes Technologiques de Sousse, Tunis..pdf »,
- [8] MAHBAB, « Automate Programmable Industriel Lycee . Technique Mohammedia, Maroc ..pdf »,
- [9] M. LARIBI, « Mémoire de master. ' Commande d'un ascenseur par PLC'. Faculté de génie électrique et d'informatique. Département d'électrotechnique, université Mouloud Mammeri de TIZI-OUZOU(2017) ..pdf »,
- [10] N. MOULAYAT et I. BENAHMED, « Développement d'un système de production FESTO via logiciels CIROS, WinCC et STEP7 », p. 72,
- [11] M. O. ENGOUANG, « Cours d'automatisme et informatique industrielle 1ère partie. », p. 32,
- [12] O. VITRY et P. ROUSSET, « Principe, Caractéristiques, maintenance 'Les actionneurs'. Lycée Léon de Lepervanche..pdf »,

- [16] B. ABDERREZAK et L. MASSINISSA, « Étude et simulation sur Wincc de la supervision d'une ligne de production d'huile 5L », p. 106,
- [17] A. BRIXI, « Cours API Part IV - Relais, Capteurs, actionneurs et leurs câblages avec un API. M2 Instrumentation Electronique. Université de Tlemcen.pdf », p. 77,
- [18] D. H. HANSEN, *Programmable Logic Controllers*, p. 419.
- [19] M. BENDIB E., « Cours Metrologie et instrumentations. Département de Génie Electrique. Ecole Supérieure en Génie Electrique et Energétique Oran. 2018/2019 »,
- [23] « Introduction à l'automatisme .Département de Physique ,Faculté des Sciences, Université de Douala..pdf »,
- [24] A. BRIXI, « Cours API Part I-Introduction aux API. M2 Instrumentation Electronique. Université de Tlemcen.pdf », p. 27,
- [25] Y. BOUDOUAOUI, « Automatisme-industrielle-Chapitre-III-pdf. Centre Universitaire Nour Bachir –El-BAYADH..pdf »,
- [26] W. BOLTON, *Programmable logic controllers*, 4th ed. Amsterdam Boston, MA : Elsevier, 2007, ISBN : 978-0-7506-8112-4.
- [32] A. BRIXI, « Cours API Part II-Structure et composants d'un API. M2 Instrumentation Electronique. Université de Tlemcen.pdf », p. 41,
- [33] T. VAIRA, « Cours Réseaux Locaux Industriels », p. 17,
- [35] L. A. BRYAN et E. A. BRYAN, *Programmable controllers : theory and implementation*, 2nd ed. Marietta, Ga : Industrial Text Co, 1997, 1035 p., ISBN : 978-0-944107-32-4.
- [36] « PROFIBUS avec STEP 7 V13 », p. 106,
- [39] D. F. S. COS-P et S. IM, « PROFINET avec STEP 7 », p. 345,
- [40] « SIMATIC. Network transitions, SIMATIC PN/CAN LINK. Edition 03/2018 »,
- [43] A. BRIXI, « Cours API Part III -Programmation des API (1ère partie). M2 Instrumentation Electronique. Université de Tlemcen.pdf », p. 35,
- [45] B. ROHÉE, « Contribution à la conception d'applications de pilotage des systèmes manufacturiers », p. 173,
- [46] D. IDOUGH, « Contribution à un cadre de spécification et conception d'IHM de supervision à base de services web dans les systèmes industriels complexes », p. 239,
- [47] O. GOUBALI, A. BIGNON, P. BERRUET, P. GIRARD et L. GUITTET, « Anaxagore, un exemple d'ingénierie dirigée par les modèles pour la supervision industrielle. », p. 8,
- [52] « Ikhlef Boualem « contribution à l'étude de supervision industrielle automatique dans un environnement SCADA » mémoire magistère université M'HAMED BOUGARA de BOUMERDES 2009. »,
- [54] H. ID, « Supervision et diagnostic des procédés de production d'eau potable », p. 164,
- [56] J. WILES, T. CLAYPOOLE, P. DRAKE et al., *Techno Security's Guide to Securing SCADA : A Comprehensive Handbook On Protecting The Critical Infrastructure*. Syngress, 23 août 2008, 350 p., Google-Books-ID : sHtIdWn1gnAC, ISBN : 978-0-08-056999-4.

- [57] R. L. KRUTZ, *Securing SCADA Systems*. John Wiley & Sons, 7 nov. 2005, 240 p., Google-Books-ID : 2MEWw2cdNI8C, ISBN : 978-0-471-78768-6.
- [58] S. KESRAOUI, « Intégration des techniques de vérification formelle dans une approche de conception des systèmes de contrôle-commande : application aux architectures SCADA », p. 255, 2017.
- [63] « Eric D. Knapp, Joel Thomas Langill-Industrial Network Security, Second Edition_ Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems-Syngress (2014) »,
- [73] « WinCC flexible 2008 Compact / Standard / Advanced », p. 488, 2008.
- [74] R. MAHJOUR, « Description de WinCC flexible ; Les étapes de création d'un projet WinCC ; Application. », p. 13,

WEBOGRAPHIE

- [13] K. KARRAY et O. ABID, *Les Actionneurs | PDF | Transmission (Mécanique) | Champ magnétique*, Scribd, 2010. adresse : <https://fr.scribd.com/document/45617091/Les-Actionneurs> (visité le 30/03/2022).
- [14] *16AM series - Moteur pneumatique à palettes by GAST | DirectIndustry*. adresse : <https://www.directindustry.fr/prod/gast/product-7063-16637.html> (visité le 30/03/2022).
- [15] M. BELHADJ, *Capteurs et Instrumentation. Département de Génie Electrique, Faculté de Technologie, Université Tahri Mohamed de Bechar (2017/2018)*, Publisher : Unpublished, 2017. DOI : 10.13140/RG.2.2.10963.84005. adresse : <http://rgdoi.net/10.13140/RG.2.2.10963.84005> (visité le 27/02/2022).
- [20] G. TRONIC, *Capteur de pression MPX5500DP*, GO TRONIC. adresse : <https://www.gotronic.fr/art-capteur-de-pression-mpx5500dp-3797.htm> (visité le 30/03/2022).
- [21] *note_technique_capteurs_FR.pdf*. adresse : https://www.celduc-relais.com/wp-content/PDF/note_technique_capteurs_FR.pdf (visité le 26/04/2022).
- [22] (. 2. CONRAD, *Interrupteur de fin de course TRU COMPONENTS XZ-9/104 1426606 250 V/AC 10 A levier pivotant à galet à rappel IP65 1 pc(s | Conrad.fr*. adresse : <https://www.conrad.fr/p/interrupteur-de-fin-de-course-tru-components-xz-9104-1426606-250-vac-10-a-levier-pivotant-a-galet-a-rappel-ip65-1-pcs-1564392> (visité le 30/03/2022).
- [27] *6ES7314-6CH04-4AB0 | Siemens Simatic S7-300 - Compact CPU | PLC-City @ 2226.00 EUR*. adresse : <https://www.plc-city.com/shop/fr/siemens-simatic-s7-300-cpu-compact/6es7314-6ch04-4ab0.html> (visité le 29/05/2022).
- [28] *17008___catalogue_automatismes_4b_749341.pdf*. adresse : https://energie-industrie.com/media/Presentation/17008___catalogue_automatismes_4b_749341.pdf (visité le 31/03/2022).

- [29] *S7-1200 - Automate programmable compact by Siemens Safety Integrated | DirectIndustry*. adresse : <https://www.directindustry.fr/prod/siemens-safety-integrated/product-14423-2230615.html> (visité le 25/05/2022).
- [30] *Etude de l'automatisation et de la supervision d'un procédé de lavage de filtres Niagara a CEVITA.pdf*. adresse : <http://www.univ-bejaia.dz/xmlui/bitstream/handle/123456789/3550/Etude%20de%20l%E2%80%99automatisation%20et%20de%20la%20supervision%20d%E2%80%99un%20proc%C3%A9d%C3%A9%20de%20lavage%20de%20filtres%20Niagara%20a%20CEVITA.pdf?sequence=1&isAllowed=y> (visité le 31/03/2022).
- [31] *6ES7314-1AG14-0AB0 | Siemens Simatic S7-300 - CPU | PLC-City @ 756.13 EUR*. adresse : <https://www.plc-city.com/shop/fr/siemens-simatic-s7-300-cpu-standard/6es7314-1ag14-0ab0.html> (visité le 24/05/2022).
- [34] A. SENSE, *Cours bus de terrain ASI (Consulté le 19/03/2022)*. adresse : <https://www.automation-sense.com/blog/automatisme/bus-de-terrain-asi-siemens.html> (visité le 19/03/2022).
- [37] BY, *What is profinet and how to configure in siemens PLC ? (consulté le 28/03/2022)*, Inst Tools, 6 mai 2020. adresse : <https://instrumentationtools.com/profinet-in-siemens-plc/> (visité le 28/03/2022).
- [38] *Industrial Wireless Sensor Networks de Elsevier Science - Livre électronique | Scribd*. adresse : <https://www.scribd.com/book/288390030/Industrial-Wireless-Sensor-Networks-Monitoring-Control-and-Automation> (visité le 28/03/2022).
- [41] *Multi Point Interface*, Dictionnaires et Encyclopédies sur 'Academic'. adresse : <https://fr-academic.com/dic.nsf/frwiki/2052106> (visité le 29/05/2022).
- [42] *05 Ladder | PDF | Programme informatique | Programmation*, Scribd. adresse : <https://fr.scribd.com/document/415039053/05-Ladder> (visité le 28/03/2022).
- [44] D. O. KACI, *Chapitre 2 Programmation des API*. adresse : https://www.academia.edu/39124686/Chapitre_2_Programmation_des_API (visité le 28/03/2022).
- [48] *HAMMOUDA Inaam (Mise en Place d'un Système.pdf*. adresse : <http://di.univ-blida.dz:8080/jspui/bitstream/123456789/5450/1/HAMMOUDA%20Inaam%20%28Mise%20en%20Place%20d%E2%80%99un%20Syst%C3%A8me.pdf> (visité le 23/04/2022).
- [49] *Partie 2 Supervision - Khadiri | PDF | Scada | Laminage*, Scribd. adresse : <https://fr.scribd.com/document/474999514/Partie-2-Supervision-Khadiri> (visité le 09/04/2022).
- [50] B. AOUIFI et R. TOUMI, *Contribution à l'étude de supervision industrielle par l'approche structurelle : Application sous un environnement SCADA*, thèse de doct., 17 mai 2017. DOI : [10.13140/RG.2.2.35624.70405](https://doi.org/10.13140/RG.2.2.35624.70405).
- [51] *mémoire_19.pdf*. adresse : https://dspace.univ-guelma.dz/jspui/bitstream/123456789/3989/1/m%C3%A9moire_19.pdf (visité le 23/04/2022).
- [53] *resumsdi.pdf*. adresse : http://staff.univ-batna2.dz/sites/default/files/benlamoudi_azzeddine/files/resumsdi.pdf (visité le 23/04/2022).

- [55] *mc1i01.pdf*. adresse : <https://accelconf.web.cern.ch/ica99/papers/mc1i01.pdf> (visité le 24/04/2022).
- [59] *SCADA system - components, hardware & software architecture, types*, electricalfundablog.com, 12 fév. 2019. adresse : <https://electricalfundablog.com/scada-system-components-architecture/> (visité le 26/04/2022).
- [60] *Human machine interface (HMI) and its importance in industrial automation | ManufacturingTomorrow*. adresse : <https://manufacturingtomorrow.com/article/2022/01/human-machine-interface-hmi-and-its-importance-in-industrial-automation/18191> (visité le 26/04/2022).
- [61] *Supervision industrielle.pdf*. adresse : <https://www.cours-exercices-pdf.com/pdf/Technologie/industrie/Supervision%20industrielle.pdf> (visité le 07/05/2022).
- [62] *What is HMI ?*, Inductive Automation. adresse : <http://www.inductiveautomation.com/resources/article/what-is-hmi> (visité le 26/04/2022).
- [64] *Qu'est-ce qu'une IHM : à quoi sert l'Interface Homme Machine ?*, IP Systèmes. adresse : <https://www.ip-systemes.com/details-qu'est-ce-qu'une-ihm-a-quoi-sert-l-interface-homme-machine--814.html> (visité le 26/04/2022).
- [65] *Qu'est-ce qu'une IHM ? Interface homme-machine | COPA-DATA*. adresse : <https://www.copadata.com/fr/produits/zenon-software-platform/visualisation-contrôle/quest-ce-qu-une-ihm-interface-homme-machine-copa-data/> (visité le 26/04/2022).
- [66] *Machine level HMI*, Siemens France, mobility-portfolio-overview. adresse : <https://new.siemens.com/fr/fr/produits/automatisation-entrainements/simatic-hmi/panels.html> (visité le 26/04/2022).
- [67] *HMI Guide*. adresse : <https://www.anaheimautomation.com/manuals/forms/hmi-guide.php> (visité le 26/04/2022).
- [68] *Les IHM optimisent la collaboration entre opérations et maintenance*, Production Maintenance, 9 déc. 2020. adresse : <https://production-maintenance.com/avis-dexpert-ihm-collaboration-operations-maintenance/> (visité le 07/05/2022).
- [69] FESTO, *CIROS® Automation Suite, Virtual Learning Environments, V1.0*, 2011. adresse : <https://www.festo-didactic.com/ch-fr/services/logiciels/logiciels/ciros.htm?fbid=Y2guZnIuNTM5LjE2LjMyLjgyNS43ODI5>.
- [70] SIMATIC, *S7-PLCSIM V5.4 .pdf*, juill. 2011. adresse : https://cache.industry.siemens.com/dl/files/828/54667828/att_110512/v1/s7svhdc_fr-FR.pdf (visité le 30/05/2022).
- [71] *FESTO EASYPORT USB D16A MANUAL Pdf Download*, ManualsLib. adresse : <https://www.manualslib.com/manual/1752006/Festo-Easyport-Usb-D16a.html> (visité le 30/05/2022).
- [72] A. SENSE, *PG/PC et configuration des interfaces PG/PC sous Step7 Siemens*. adresse : <https://www.automation-sense.com/blog/automatisme/pg-pc-et-configuration-des-interfaces-pg-pc-sous-step7-siemens.html> (visité le 30/05/2022).

Résumé

L'industrie du Vingt et unième siècle est devenue de plus en plus automatisée et connectée, surtout avec la nouvelle génération industrielle représentée par le concept d'industrie 4.0 qui utilise les nouvelles sciences et technologies du numérique telle que l'intelligence artificielle, le big data, l'internet des objets, entre autres. Pour cela l'automatisation des systèmes industriels est devenu une obligation. Un système automatisé doit forcément être commandé par un ou plusieurs automates programmables industriels et supervisé par un système de supervision SCADA ou au minimum avoir une ou plusieurs IHM. Dans notre travail, nous allons proposer un système de production automatisé moderne qu'on va appeler MPS variant qui est une variante du MPS original de FESTO. Ce système se compose de quatre sous-stations (Distribution, usinage, manipulation et livraison), tous d'abord nous allons modéliser, configurer et programmer les quatre sous-stations en utilisant le logiciel de programmation des APIs SIEMENS STEP 7. Ensuite, nous allons développer des interfaces homme-machine IHM, pour la visualisation et la supervision des sous-stations du système étudié. Pour cette tâche, nous allons utiliser le logiciel d'ingénierie de conception et de développement des IHMs WinCC Flexible.

Mots clés :

Programmation API, STEP7, PLCSIM, WinCC Flexible, RUNTIME, EasyPort USB, Simulation.

الملخص

أصبحت صناعة القرن الحادي والعشرين مؤتمتة ومتصلة بشكل متزايد ، خاصة مع الجيل الصناعي الجديد الذي يمثله مفهوم الصناعة 4.0 الذي يستخدم علومًا جديدة وتقنيات رقمية مثل الذكاء الاصطناعي والبيانات الضخمة وإنترنت الأشياء ، من بين أمور أخرى. لهذا ، أصبحت أتمتة الأنظمة الصناعية التزامًا. يجب بالضرورة التحكم في النظام الآلي بواسطة واحد أو أكثر من وحدات التحكم المنطقية (PLCs) والإشراف عليها بواسطة نظام SCADA أو على الأقل به واحد أو أكثر من HMI. سنقترح في عملنا نظام إنتاج آلي حديث نسماه متغير MPS وهو البديل من MPS الأصلي من FESTO. يتكون هذا النظام من أربع محطات فرعية (التوزيع والمعالجة والنقل والتسليم) ، أولاً سنعمل على نمذجة وتهيئة وبرمجة المحطات الفرعية الأربع باستخدام برنامج برمجة PLC 7 STEP SIEMENS ثم سنقوم بتطوير HMIs للتصور والإشراف على المحطات الفرعية النظام المدروس. لهذه المهمة سوف نستخدم البرامج الهندسية لتصميم وتطوير HMIs Flexible WinCC.

كلمات مفتاحية :

تحسين البرمجة , PLC, STEP7, PLCSIM, WinCC Flexible, EasyPort RUNTIME, USB, المحاكاة ..

Abstract

The industry of the twenty-first century has become increasingly automated and connected, especially with the new industrial generation represented by the concept of industry 4.0 which uses new sciences and digital technologies such as artificial intelligence, big data, internet of things, among others. For this, the automation of industrial systems has become an obligation. An automated system must necessarily be controlled by one or more PLCs and supervised by a SCADA system or at least have one or more HMI. In our work we will propose a modern automated production system that we will call MPS variant which is a variant of the original MPS from FESTO. This system consists of four substations (Distribution, Machining, Handling and Delivery), first we will model, configure and program the four substations using the SIEMENS STEP 7 PLC programming software and then we will develop HMIs for the visualization and supervision of the substations of the studied system. For this task we will use the engineering software of design and development of HMIs WinCC Flexible

Keywords : PLC programming, STEP7, PLCSIM, WinCC Flexible, RUNTIME, EasyPort USB, Simulation, Emulation.