

République Algérienne Démocratique et Populaire

MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE

SCIENTIFIQUE



UNIVERSITE ABOU BEKR BELKAID DE TLEMCCEN
FACULTE DE TECHNOLOGIE
DEPARTEMENT DE GENIE ELECTRIQUE ET ELECTRONIQUE



MEMOIRE

Présenté pour l'obtention du diplôme

Master II en Génie Industriel

Intitulé :

Pilotage et supervision de système automatisé 3 modules (module bande transporteuse, module manipulateur tournante et module contrôle de la pièce) du laboratoire Productique MELT Université de Tlemcen

Soutenu le 29/06/2019 Par :

- ✓ BOULECHFAR Zohier
- ✓ HANI Laid

Devant le Jury :

Mr. GHOMRI Latéfa	MCA	Présidente
Mr. HASSAM Ahmed	MCB	Examineur
Mr. HADRI Abdelkader	MAA	Examineur
Mr. MELIANI Sidi Mohammed	MCA	Encadrant
Mr. MKEDDER Med Amin	Ingénieur de recherche et développement	Co-Encadrant

Année Universitaire : 2018/2019

Dédicace

La vie n'est qu'un éclair,
Et un jour de réussite est un jour très cher.

Je dédie ce mémoire

A mon cher père

Et ma très chère mère

Pour l'éducation et le grand amour dont ils m'ont
entouré depuis ma naissance.

Et pour leurs patiences et leurs sacrifices.

A mes chers frères ;

A mes chères sœurs ;

A tous mes proches : mes tantes, mes cousines
Sami, Anis et Et surtout Mr : Sidi Mohammed
MELIANI et Mr : Amine MKEDDER qui m'ont
aidé pendant mon travail, mes voisins et toute la
famille BOULECHFAR.

A tous ceux qui m'aiment ;

A tous mes ami (e) s;

A tous ceux que j'aime;

A tous mes collègues de promo Master II

en

Génie industriel (2018/2019) ;

Boulechfar zohier

Dédicace

La vie n'est qu'un éclair,
Et un jour de réussite est un jour très cher.

Je dédie ce mémoire

A mon cher père **TOUHAMI**

Et ma très chère mère **MERIEM**

Pour l'éducation et le grand amour dont ils m'ont
entouré depuis ma naissance.

Et pour leurs patiences et leurs sacrifices.

A mes chers frères **ZAKARIA .DJABER ET ZOUBIR ;**

A mes chères sœurs **AICHA, LATIFA, ASMAA,
BATOUL, ET FATIMA ;**

A tous mes proches : mes tantes **SAMIHA, HADDA
ET RACHIDA**, mes cousines **AHMED ET DJAMEL** et

Et toute personne m'aider pendant ce travail, mes
voisins et toute la famille **HANI**.

A tous ceux qui m'aiment ;

A tous mes ami (e) s **B-ADIMO B-NADIR,
KOUSOU-K, B-Mohamed ; B-SMAIL, RAGHDI-Y,
Z-ELAMRI, AYOUB-B ;**

A tous ceux que j'aime ...;

A tous mes collègues de promo Master II
en

Génie industriel (2018/2019) ;

Hani Laid

Remerciement

Tout d'abord, nous remercions DIEU tout puissant de nous avoir accordé des connaissances de la science et de nous avoir aidé à réaliser ce travail.

Nous tenons à remercier en cette occasion tout le corps enseignant qui a contribué à notre formation, pour son dynamisme et la qualité de l'enseignement qui nous ont été dispensés, nos encadreurs Mr MELIANI Sidi Mohammed et Mr MKEDDER Med Amin pour leurs aides, leurs écoutes, leurs disponibilités et les précieux conseils qu'ils ont bien voulu nous donner.

Nous remercions aussi vivement les membres de jury qui vont se pencher sur notre travail et vont lui prêter toute l'attention qu'il mérite pour l'évaluation à sa juste valeur.

Nous n'oublions pas nos parents pour leurs contributions, leur soutien et leur patience, nos proches et nos amis qui nous ont soutenu et encouragé, ainsi que toutes les personnes qui vont nous ont aidé de près ou de loin.

Merci à tous.

Sommaire

Chapitre I

Introduction Générale.....	1
I.1 Introduction.....	3
I.2. Généralité sur les systèmes automatisés	3
I.2.1 Historique [1].....	3
I.2.2 Définition	4
I.2.3 Décomposition du système automatisé.....	4
I.2.4 Echange d'informations dans un système automatisé	5
I.2.5 Chaîne fonctionnelle	5
I.2.6 Exemple de circulation de l'information	6
I.2.7 Objectif du système automatisé	7
I.2.8 Avantages et inconvénients du système automatisé	7
I.3 Généralité sur le système SCADA.....	8
I.3.1 Définition du système SCADA.....	8
I.3.2 Architecture du SCADA	8
I.3.3 Protocole employé dans un environnement SCADA.....	10
I.3.4 Logiciel HIM de SCADA	13
I.3.5 Supervision dans un environnement SCADA.....	14
I.3.6 Domaines d'application	14
I.3.7 Exemple d'application de supervision sous logiciel WinCC.....	15
I.4 Conclusion	16

Chapitre II

II.1 Introduction	17
II.2 Description du système étudié.....	17
II.2.1 Présentation du module « bande transporteuse MCS-700/EV »	17
II.2.2 Présentation du module manipulateur tournant MC S-510/EV	18
II.2.3 Présentation du module de contrôlées des pièces MCS-570/EV.....	19
II.2.4 Fonctionnement du système	21
II.3 Automate programmable (API).....	22
II.3.1 Définition.....	22
II.3.2 Type de conception d'API.....	23
II.3.3 Présentation du S7-300.....	24
II.3.4 Architecture Général de l'API.....	25

II.3.5 Cycle d'un API [9]	26
II.4 Logiciels de développement	26
II.4.1 STEP 7	26
II.4.2 Langages de programmation	27
II.4.3 Grafcet	27
II.5 WinCC flexible.....	31
II.5.1 WinCC flexible Runtime [12]	31
II.6 Conclusion	32

Chapitre III

III.1 Introduction	34
III.2 Programmation sur Step7	34
III.2.1 Principe.....	34
III.2.2 Démarrage de STEP 7	34
III.2.3 Création d'un nouveau projet.....	35
III.2.4 Configuration matérielle	36
III.2.5 Création de la table des Mnémoniques	41
III.2.6 Elaboration du programme.....	45
III.3 Programmation sur WinCC.....	52
III.3.1 Création et configuration du projet	52
III.3.2 Configuration des vues.....	55
III.3.3 Représentation de l'IHM de chaque module.....	60
III.3.4 Importation des données du système.....	62
III.4 Compilation et simulation du programme.....	64
III.4.1 Simulation sous Step7	64
III.4.2 Simulation avec WinCC.....	66
III.4.3 Simulation de programme PLCSIM avec WinCC	67
III.5 Injection du projet dans API réel S7-300	67
III.5.1 Configuration des Adresses selon le Hardware de notre API S7-300	67
III.5.2 Commande du système via IHM pupitre.....	68
III.6 Conclusion.....	72
Conclusion générale.....	73
Références.....	75
Résumé.....	76

Liste des figures

Chapitre I

Figure I. 1 : Structure interne du système automatisé.	5
Figure I. 2 : Echange de l'information dans un système automatisé.	5
Figure I. 3 : Chaîne de fonctionnelle.	6
Figure I. 4 : Fonctionnement de station de pompage d'un château d'eau.	7
Figure I. 5 : Schéma général d'un système SCADA.	9
Figure I. 6 : Architecture de la supervision dans un environnement SCADA.	9
Figure I. 7 : Communication par le Protocole MODBUS.	11
Figure I. 8 : Communication par le protocole DNP3.	12
Figure I. 9 : Communication par le Protocole PROFIBUS.	13
Figure I. 10 : Interface de Contrôle de la température.	15
Figure I. 11 : Interface de Contrôle de niveau d'eau dans un réservoir.	16

Chapitre II

Figure II. 1 : Vue général du module Mod.700/EV.	18
Figure II. 2 : Vue générale du module mod.MCS-510.	19
Figure II. 3 : Vue générale du module mod.MCS-570/EV.	20
Figure II. 4 : Automate programmable industriel.	23
Figure II. 5 : Présentation d'API de type boîte unique.	24
Figure II. 6 : Présentation d'API de type modulaire et rack.	24
Figure II. 7 : Présentation du S7-300.	25
Figure II. 8 : Cycle d'un API.	26
Figure II. 9 : Structure et interprétation du Grafcet.	28
Figure II. 10 : Etape initiale.	29
Figure II. 11 : Exemple de deux critères d'actions.	29
Figure II. 12 : Exemple de deux types d'étapes.	29
Figure II. 13 : Transitions.	30
Figure II. 14 : Réceptivité.	30
Figure II. 15 : Liaison orientée.	30
Figure II. 16 : Aperçu de la fenêtre WinCC flexible.	31

Chapitre III

Figure III. 1 : Fenêtre SIMATIC Manager.	34
Figure III. 2 : Assistance de STEP7.	35
Figure III. 3 : Non existence de CPU 314 PN/DP.	35
Figure III. 4 : Fenêtre pour un nouveau projet.	36
Figure III. 5 : Ajout d'une nouvelle station.	37
Figure III. 6 : Ouverture de la fenêtre de configuration matérielle (HW Config).	37

Figure III. 7 : Comment ajouter le rack.....	38
Figure III. 8 : Comment ajouter alimentation.....	39
Figure III. 9 : Comment ajouter notre CPU.....	39
Figure III. 10 : Confirmation de communication par câble Ethernet.	40
Figure III. 11 : Configuration réseau dans STEP 7.....	41
Figure III. 12 : Hiérarchie du programme STEP7.	41
Figure III. 13 : Comment créer une table mnémotechnique.	42
Figure III. 14 : Variables utilisées dans notre programme.....	43
Figure III. 15 : Suite des variables utilisées dans notre programme.	44
Figure III. 16 : Suite des variables utilisées dans notre programme.	44
Figure III. 17 : Différents blocs de programmation utilisés.....	46
Figure III. 18 : Comment crée un bloc FC.	46
Figure III. 19 : Déplacement de la pièce droite à gauche.	47
Figure III. 20 : Déplacement de la pièce gauche à droite.	47
Figure III. 21 : Comment ajouter un Memento de cadence.	48
Figure III. 22 : Comment crée un bloc fonctionnel.	49
Figure III. 23 : Aperçu sur le programme de la bande transporteuse.....	49
Figure III. 24 : Aperçu sur le programme du module manipulateur.....	50
Figure III. 25 : Aperçu sur le programme du module contrôle de pièce.	50
Figure III. 26 : Comment créer un bloc d'organisation manuellement.....	51
Figure III. 27 : Exemple comment ajouter les autres blocs dans bloc OB.....	51
Figure III. 28 : Insertion d'un nouveau Station.	52
Figure III. 29 : Choix du pupitre.....	53
Figure III. 30 : Comment ouvrir la fenêtre de configuration du réseau.	53
Figure III. 31 : Réseau Ethernet après configuration.	54
Figure III. 32 : Comment ouvrez WinCC à travers Step7.	54
Figure III. 33 : Configuration de réseau pupitre vers l'automate.....	55
Figure III. 34 : Déplacement de la pièce droite à la gauche.....	56
Figure III. 35 : Comment changer la couleur automatiquement dans le WinCC.....	57
Figure III. 36 : Comment utiliser la visibilité pour montrer le mouvement du manipulateur.....	58
Figure III. 37 : Utilisation de la visibilité.	59
Figure III. 38 : Partie1 : configuration du bouton.	59
Figure III. 39 : Partie2 : configuration le bouton.	60
Figure III. 40 : IHM du module bande transporteuse.	60
Figure III. 41: IHM du module manipulateur.	61
Figure III. 42 : IHM du module contrôle de pièce.	61
Figure III. 43 : IHM principale.	62
Figure III. 44 : Message envoyé par le système.....	62
Figure III. 45 : Ouverture de la fenêtre d'importation.	63
Figure III. 46 : Fenêtre de choix des données.	63
Figure III. 47 : Importation du texte de la variable.....	64
Figure III. 48 : Activation de la simulation.	64
Figure III. 49 : Chargement du programme.....	65
Figure III. 50 : Vue générale du logiciel S7-PLCSIM.	65
Figure III. 51 : Démarrage de la simulation.	66
Figure III. 52 : Activation de la visualisation avec Runtime.	66

Figure III. 53 : Page d'accueil du système sur le pupitre.....	67
Figure III. 54 : Exécution de l'IHM via le simulateur PLCsim.....	67
Figure III. 55 : Configuration des adresses DI/DO de chaque module.	68
Figure III. 56 : Connexion PC avec l'automate Via MPI adaptateur.....	68
Figure III. 57 : Paramétrage de l'interface.....	69
Figure III. 58 : Adaptateur MPI.	70
Figure III. 59 : Connexion PC avec l'automate Via câble Ethernet.....	70
Figure III. 60 : Configuration réseau PC / automate Via câble Ethernet.....	71
Figure III. 61 : Configuration d'adresse IP de l'ordinateur.	71

Liste de Tableaux

Tableau 1 : Différence entre les bits.	48
Tableau 2 : Table de calcul d'intervalle de changement de couleur de pièce.....	57
Tableau 3 : Exemple de calcul d'intervalle.	57

Introduction Générale

Le concept de la supervision est né au début des années 1980, lorsque la création des réseaux informatiques dans les entreprises a explosé. La taille et la variabilité croissantes de ces derniers sont un réel problème de gestion et d'administration, ce qui complique les besoins de la main-d'œuvre des fonctionnaires administratifs. Les réflexions initiales sur le nouveau concept, à savoir la supervision, ont donc été mises en œuvre.

La productivité industrielle est un facteur très important dans le progrès d'une Entreprise. En outre, la disponibilité, la sécurité, la qualité, l'efficacité et la conformité répondent au succès de l'entreprise. Les problèmes liés au système de production, tels que les pannes, les interruptions, les pannes et divers problèmes techniques. Les décideurs de l'industrie ont réagi en investissant dans la rénovation des unités de production en continuité opérationnelle. Avec la conception de la supervision en temps réel en utilisant l'interface du dispositif humain, la communication sans fil devient une nécessité indispensable.

Les systèmes de supervision offrent des vues modulaires d'équipements ou d'ensembles d'équipements pour visualiser leurs situations physiques ou fonctionnelles. Les systèmes de supervision situés dans les salles de contrôle ou en sein de la machine permettent de relayer et de contrôler la vision physique des organes de manière centralisée. (Capteurs, moteurs) sont parfois très éloignés les uns des autres. L'interface de communication mis en œuvre, qui sont contactés pour se connecter aux systèmes de contrôle (API, SNCC, machines spéciales, etc.), sont contactés par des protocoles de communications industrielles standard ou privés.

Notre travail consiste à réaliser un programme sous STEP7, avec un automate Programmable SIEMENS S7-300, a fin de commander un système automatisé de contrôle de la pièce, suivi d'une supervision sous WinCC flexible.

En base, le système automatisé étudié dans notre projet appelé « Système mécatronique (mod. MCS-G1/EV) » est un système de gestion en boucle fermée d'un magasin multiniveau pourvu de station d'identification sur table tournant avec transporteur. L'ensemble est situé au labo de recherche MELT de l'université ABOU BEKR BEKAID à TLEMCEN.

Le système étudié ne contient pas un programme. Nous sommes le premier groupe qui travaille sur ce système et le premier qui vont créer un programme pour faire fonctionner ce système dans l'université.

Ce système est constitué par les éléments suivants :

- Module manipulateur rotatif
- Module de contrôle de pièces,

- Module bras pneumatique
- Module d'emmagasinement à contrôle cartésien
- Module bande transporteuse
- Caisse contenant les pièces et outils

Sachant que le système est très compliqué, comme première étape et durant ce PFE, nous avons jugé utile d'étudier uniquement les modules suivants :

- Module de contrôle de pièces
- Module bande transporteuse
- Module bras pneumatique

L'objectif de notre projet peut être résumé sur 4 points suivants :

- Développer un programme graphique commandant les 3 modules.
- Faire la supervision des 3 modules en temps réel.
- Tester notre contribution par simulation en utilisant PLC-CIM et RUNTIEM-simulation.
- Enfin injecter le tout au niveau du processus réel.

Notre travail se répartit en trois chapitres ; Dans le premier chapitre, on a présenté le système de supervision et de contrôle SCADA, avec son utilité et architecture. On a présenté quelques exemples d'application par le logiciel WinCC flexible.

Le deuxième chapitre est consacré à l'identification du matériel utilisé, et la présentation du système étudié, la structure de l'automate siemens S7-300, et la description du logiciel associée.

Le troisième chapitre est consacré à la programmation et l'élaboration de la supervision du système proposé et nous terminons notre travail par une conclusion générale.

Chapitre I

Systemes automatisés dans le domaine SCADA

I.1 Introduction

Le contrôle de surveillance et l'acquisition des données (SCADA) est une collection de logiciels et de matériel, ça première apparition était aux années 60. Il permet le contrôle à distance sans intervention manuelle d'un système automatisé. Presque dans le monde industriel, on utilise beaucoup les systèmes techniques automatisés pour la prise de décision et la commande de système sans intervention humaine, pour but la conduite à une très grand rapidité, une meilleure régularité des résultats et éviter des taches pénibles et répétitives.

Aujourd'hui, le système SCADA est utilisé dans la plupart des domaines technologies (réseaux, électronique, informatique...) et sont devenus omniprésents sur les installations à caractère industriel. Il permet la surveillance, la collecte et le traitement des données en temps réel pour des décisions plus intelligentes et communiquent les problèmes du système afin de réduire les temps d'arrêt.

Ce système est souvent utilisé pour surveiller les opérations de production et de transport ainsi que les systèmes d'approvisionnement en eau, comme par exemple commander la génération d'énergie électrique, la transmission et la distribution, les canalisations de gaz et de pétrole, et d'autres protocoles industriels.

Ce chapitre comporte deux parties. Dans la première partie, nous parlerons sur le système automatisé, les différentes parties du système et comment l'information est transmise en général.

Deuxième partie contiendra la généralité et le rôle du système SCADA dans le fonctionnement et l'architecture du système automatisé. On finira par simuler quelques applications utilisant le logiciel WinCC.

I.2. Généralité sur les systèmes automatisés

I.2.1 Historique [1]

Les automates programmables (API) sont apparus à la fin des années soixante, à la demande de l'industrie automobile américaine, qui réclamait plus d'adaptabilité de leur système de commande. Les couts de l'électronique permettant alors de remplacer avantageusement les technologies utilisées. La technologie utilisée avant était les systèmes câblés à base de relais électromécaniques et des systèmes pneumatiques et hydrauliques. Avec l'apparition des microprocesseurs et des systèmes programmés, les contraintes engendrées par l'utilisation des systèmes câblés sont devenues injustifiables. Parmi ces inconvénients on peut citer :

- Les câbles sont encombrants et couteux.
- Pas de flexibilité.
- Pas de communication possible.

I.2.2 Définition

Un système automatisé ou automatique est un système mécanique réalisant une opération et capable de faire une action sans présence humaine et dont son réglage se fait à travers des programmes.

Le but d'un système automatisé est de réaliser des tâches complexes ou dangereuses pour l'homme, effectuer des tâches pénibles ou répétitives ou encore gagner en efficacité et en précision. [2]

L'énergie nécessaire à la transformation du produit est fournie par une source extérieure, le constituant automate (communication Homme-Machine) dirige la succession des opérations. L'homme surveille le système et peut dialoguer avec lui par l'intermédiaire du pupitre [2]

I.2.3 Décomposition du système automatisé

Les systèmes automatisés se composent à la base de deux parties principales en relation avec d'autres parties pour assurer le fonctionnement de systèmes et l'interaction entre les parties.

I.2.3.1 Partie commande (PC)

On peut dire que c'est le cerveau du système qui gère le fonctionnement du système automatisé tout en assurant le pilotage et la coordination entre les actionneurs de la partie opérative.

I.2.3.2 Partie opérative (PO)

Elle consomme l'énergie (électrique, pneumatique ou hydraulique) nécessaire pour faire une action.

I.2.3.3 Pupitre

Permet le dialogue entre l'opérateur et la partie commande. L'opérateur envoie des consignes et reçoit des informations visuelles.

I.2.3.4 Éléments d'interfaces

Entre la partie commande et la partie opérative, on trouve les éléments d'une interface pour assurer le dialogue entre les deux parties, On distingue deux organes :

Pré-actionneur : distribue l'énergie disponible à l'actionneur.

Le capteur : détecte un phénomène physique et de rendre compte de ce phénomène à Pc.

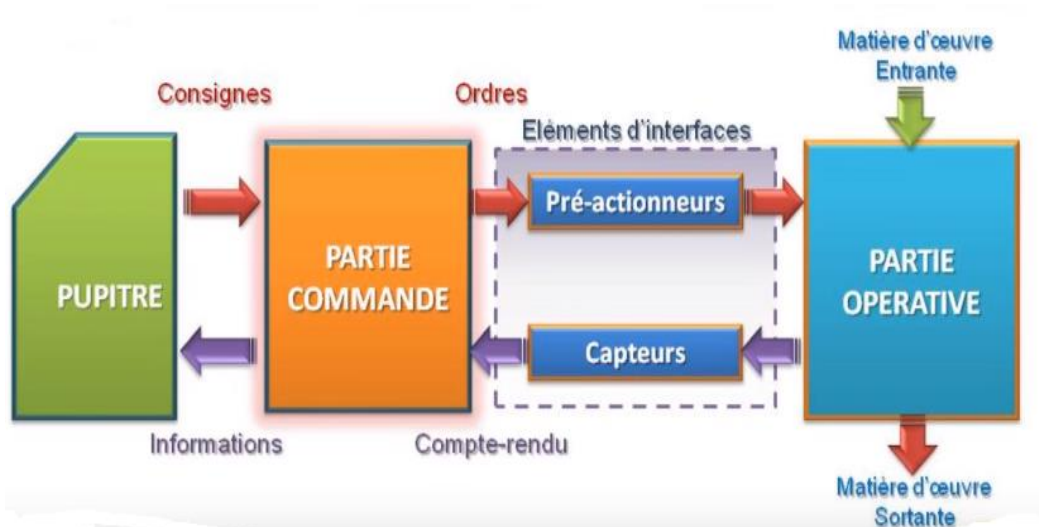


Figure I. 1 : Structure interne du système automatisé.

I.2.4 Echange d'informations dans un système automatisé

Le système automatisé est composé de deux parties principales : la partie commande (PC) et la partie opérative (PO) pour le fonctionnement du système. L'opérateur va donner la consigne à la PC, ce dernier traduit ces consignes en ordres qui vont être exécutés par la PO à travers l'actionneur (pour produire un phénomène physique) et envoie l'information à la PC à travers le capteur qui va à son tour le signaler à l'opérateur pour faire le jugement de système.

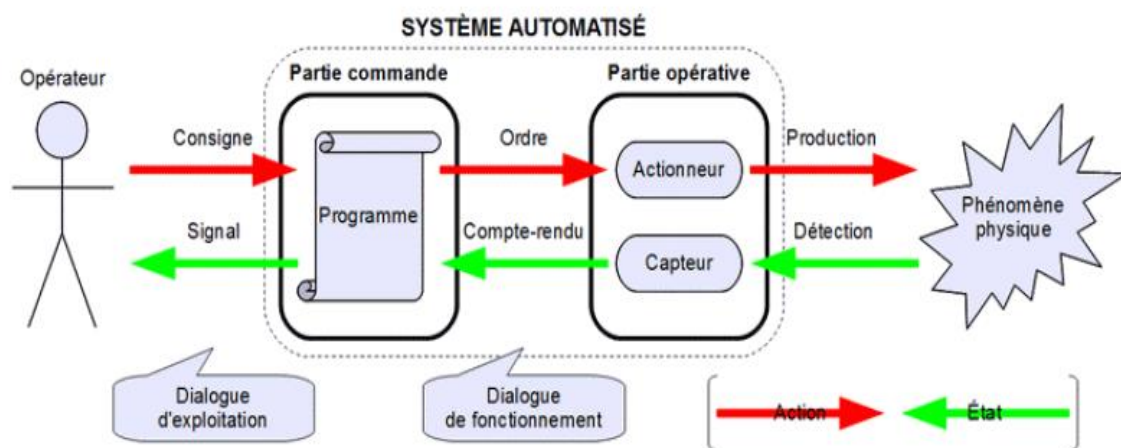


Figure I. 2 : Echange de l'information dans un système automatisé.

I.2.5 Chaîne fonctionnelle

La chaîne fonctionnelle est organisée en sous-ensemble en constituant des composants. L'ensemble des organisations et les composants qui réalisent le même genre de fonction s'appelle une chaîne de fonctionnelle.

Dans une chaîne fonctionnelle on distingue les trois chaînes suivantes :

- ✓ La chaîne d'information
- ✓ La chaîne d'énergie

✓ La chaîne d'action

I.2.5.1 Chaîne d'information

Elle permet d'acquérir, de traiter et de communiquer les informations et qui est constitué par des capteurs et d'organe à commande programmable. A l'acquisition d'une grandeur physique (force, température, la pression ...) ou une consigne de l'opérateur par le biais des capteurs, les constituants réutilisent cette grandeur sous forme logique, numérique ou analogique exploitable par l'unité de traitement.

I.2.5.2 Chaîne d'énergie

Du point de vue énergie (énergie d'entrée, énergie de puissance), il permet d'alimenter, de distribuer, de convertir et de transmettre des consigne aux interfaces de commande, exemple carte d'entrée sortie industrielle, distributeur pneumatique hydraulique.

I.2.5.3 Chaîne d'action

C'est un point de vue de transformation des matières d'œuvre. Elle permet de distribuer, de transformer, de transmettre et d'adapter l'énergie de puissance nécessaire pour faire une action sur la matière d'œuvre. Elle est constituée par des actionneurs et des effecteurs.

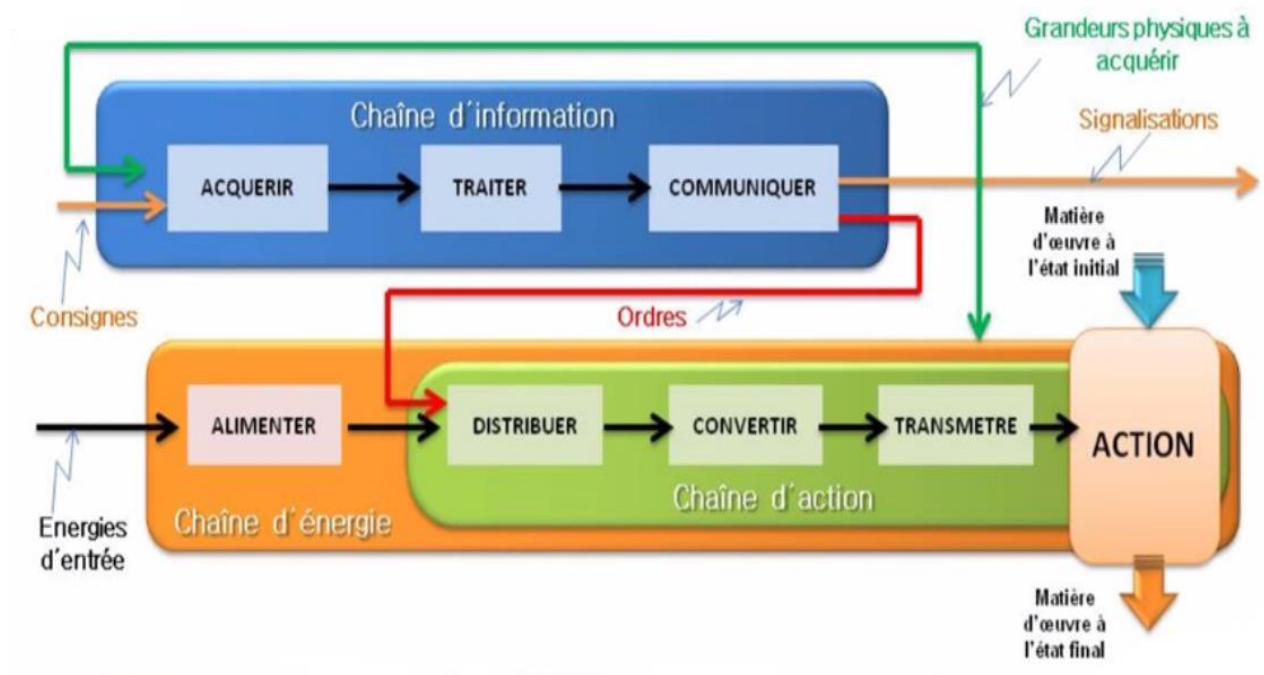


Figure I. 3 : Chaîne de fonctionnelle.

I.2.6 Exemple de circulation de l'information

Pour montrer la circulation de l'information entre PC et PO, on va observer le fonctionnement de station de pompage d'un château d'eau.

CHAPITRE I : Systèmes automatisés dans le domaine SCADA

Un château d'eau permet de stocker l'eau courante pour la distribuer dans les habitations. Pour que le réservoir soit toujours rempli à un niveau d'eau sévissant, un flotteur actionne un capteur S1 (niveau bas) et S2 (niveau haut) dans le réservoir.

On remarque aussi que la PO communique au la PC des informations sur l'état de remplissage de réservoir grâce aux capteurs S1 et S2. Ces informations reçues puis traiter par l'unité de traitement grâce à des programmes, vont envoyer des ordres à la PO via le pupitre de commande.

La PO est constituée d'un pré-actionneur (un contacteur électrique) qui distribue l'énergie électrique à l'actionneur qui est le moteur-pompe.

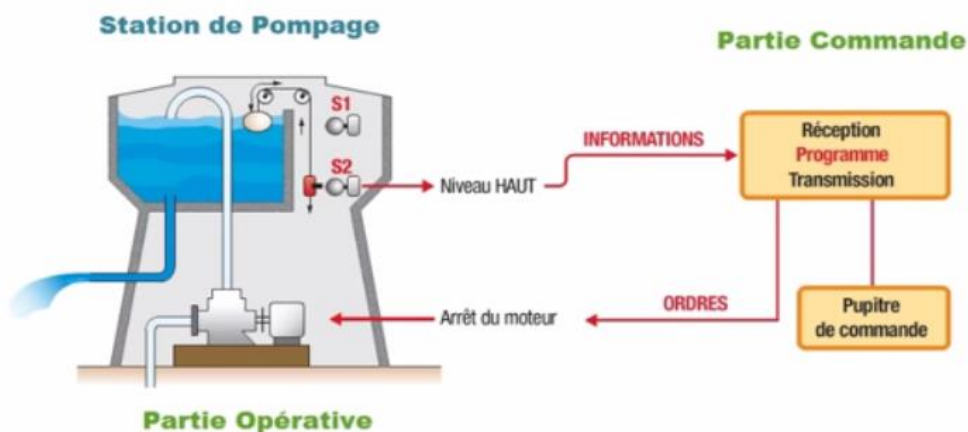


Figure I. 4 : Fonctionnement de station de pompage d'un château d'eau.

I.2.7 Objectif du système automatisé

Tout système a un objectif. Dans ce cas précis, son rôle est de :

- Réduit les couts et délai de production, simplifier le travail humain,
- Eliminer la tache pénible et répétitive,
- Augmenté la productivité et améliore la qualité,
- Réduit es risque de travaille et augmenté la sécurité,
- Economie la matière première et là l'énergie,
- Maintenir la qualité.

I.2.8 Avantages et inconvénients du système automatisé

I.2.8.1 Avantage

- Sécurité,
- Précision,
- Réduction du cout de fabrication,
- Flexibilité et rapidité,
- Amélioration des conditions de travail,
- Augmentation de la productivité.

I.2.8.2 Inconvénients

- Panne,
- Consommation d'énergies,
- Cout et délai de formation des personnes plus qualifiées,
- Cout de maintenance,
- Incidence sur l'emploi (licenciement -chômage),
- Investissement pour l'achat de machine.

I.3 Généralité sur le système SCADA

I.3.1 Définition du système SCADA

SCADA est un acronyme qui signifie le contrôle et la supervision par acquisition de données (en anglais : Supervisory Control and Data Acquisition) qui permet la centralisation des données, la présentation souvent semi-graphique sur des postes de « pilotage », Il collecte des données de divers appareils d'une quelconque installation, puis transmet ces données à un ordinateur central, que ce soit proche ou éloigné, qui alors contrôle et supervise l'installation. Ce dernier est subordonné par d'autres postes d'opérateurs, l'allure générale d'un système SCADA est montrée sur la figure ci-dessous. [3] [4]

I.3.2 Architecture du SCADA

En principe, le système SCADA est composé d'éléments suivants :

I.3.2.1 RTU (Remonte Terminal Unit) ou Plc

SCADA, à la base, commence par des unités terminales distantes (RTU) et des automates programmables (API ou PLC). Il sert à collecter les informations à partir de l'instrumentation du terrain et les transmettre au MTU à travers le système de communication. RTU peut être utilisé et remplacé par des automates programmables (API ou PLC) dans le cas où nous voulons contrôler certains points du système par un programme parce qu'il est avantageux d'écrire un programme pour commander le système contrairement à un RTU.

I.3.2.2 MTU (Master Terminal Unit)

On peut dire que la station principale ayant plusieurs stations (RTU) connectées par différents moyens de communication. Son rôle est de recueillir périodiquement les données de terrain des stations RTU et permet le contrôle à distance et la transmission d'informations à l'opérateur via l'IHM et envoyer les consignes de l'opérateur vers l'instrumentation du terrain.

I.3.2.3 Système de communication

La communication entre MTU et les différents RTU peut être différent moyen par exemple internet, réseau sans fil, câblage...

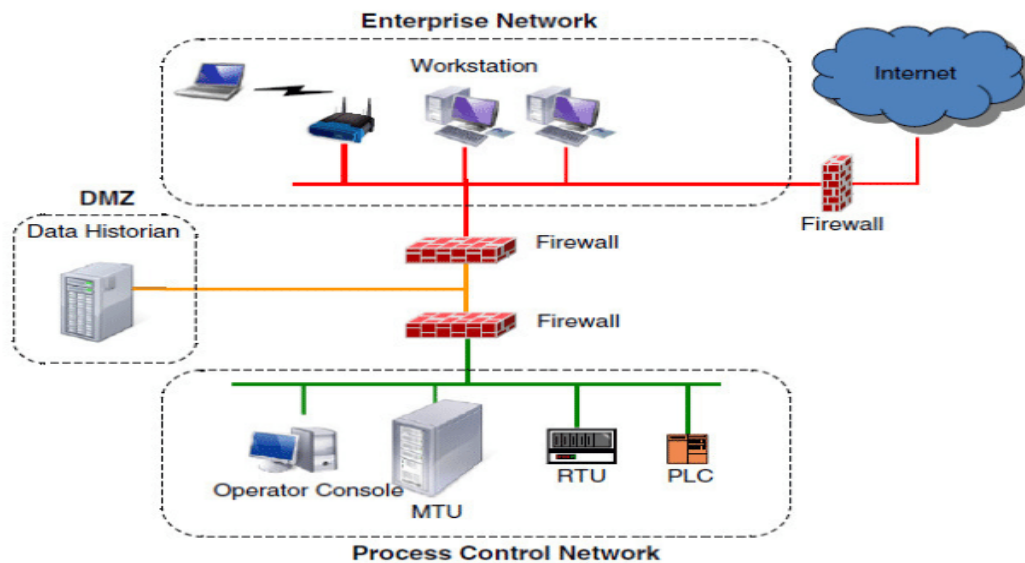


Figure I. 5 : Schéma général d'un système SCADA.

SCADA entoure un transfert de données entre le serveur (MTU, master terminal unités) et une ou plusieurs unités terminales distantes (remote terminal units RTUs) et entre le serveur et les terminaux des opérateurs. La figure ci-dessous représente un schéma sur l'architecture d'un réseau SCADA qui utilise des routeurs pour joindre le poste de pilotage par le biais d'internet [5].

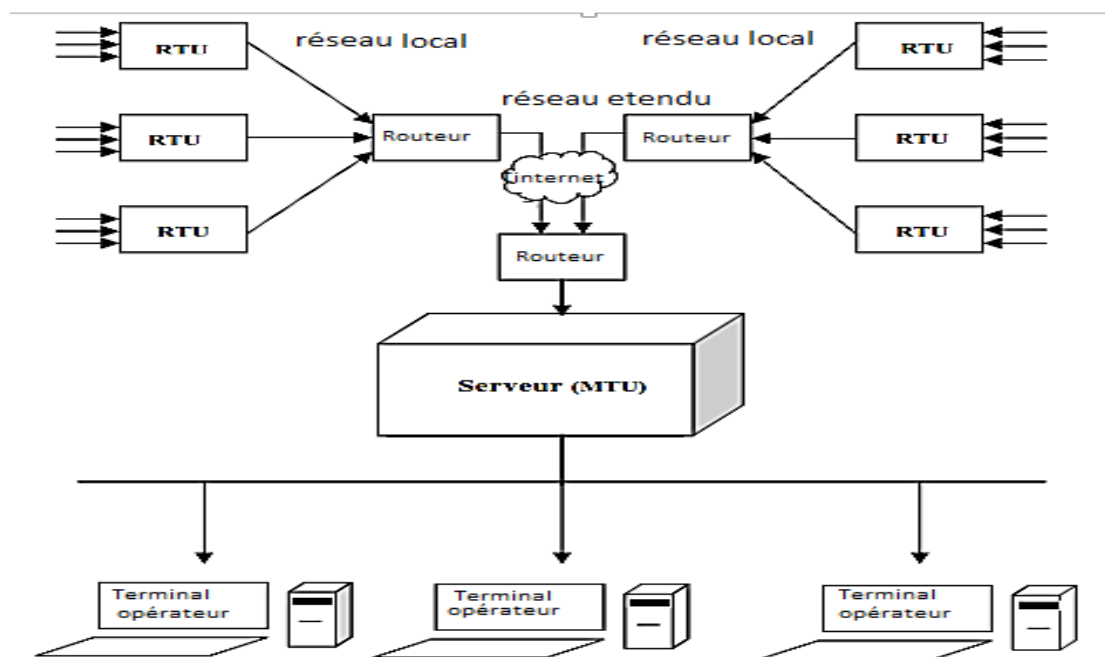


Figure I. 6 : Architecture de la supervision dans un environnement SCADA.

I.3.3 Protocole employé dans un environnement SCADA

Dans le monde réel de la communication d'automatisation de processus de périphérique à périphérique, un dialogue ou une conversation entre des appareils a lieu systématique, dans un nombreux cas, sur différents types de réseau de communication et dans différents langages.

Ce sont ces protocoles ouverts que de nombreux fabricants adaptent d'intégrer facilement leurs produits sur un marché, un protocole ouvert signifie que les spécifications sont publiées et peut être utilisé par quiconque librement ou par License.

Protocoles ouverts sont généralement appuyés par une combinaison de sociétés groupes d'utilisateur, société professionnelle et gouvernement. Cela offre aux utilisateurs un choix beaucoup plus large d'appareils.

Les protocoles de communication dans un environnement SCADA évoluent suite à la nécessité d'envoyer et de recevoir des données jugées critiques généralement pour de longues distances et en temps réel, cette optique a donné naissance de plusieurs protocole qu'on va Développer les plus utilises [5].

I.3.3.1 Protocole Modbus

Modbus est l'un des composants électroniques industriels utilisés actuellement, le plus ancien et de loin le protocole d'automatisation le plus populaire dans le domaine de l'automatisation des processus et du SCADA.

MODBUS est un protocole de communication publié par Modicon en 1979 pour une utilisation avec ses automates programmables. Modbus fournit un langage commun pour que les appareils et équipements puissent communiquer entre eux, par exemple, MODBUS active les périphériques sur un système qui mesure la température et l'humidité connectées sur le même réseau ou communiquer les résultats à un ordinateur de supervision ou à un automate.

Le protocole MODBUS est un protocole de transmission de données régissant le dialogue entre une station "Maitre" et des stations "Esclaves". L'échange Maitre-Esclave s'effectue par l'envoi de trames MODBUS le format de base est le suivant (voir Figure 1-7) [6]:

Champ Adresse	Champ Fonction code	Champ Données de commande	Champ de vérification d'erreur
---------------	---------------------	---------------------------	--------------------------------

Champ Adresse : correspond à l'adresse de la station Esclave destinatrice de la Requête.

Champ Fonction code : c'est la commande de lecture ou d'écriture de données à l'esclave.

Champ Données : si une commande d'écriture a été lancée par le maitre.

Champ de vérification d'erreur : est une valeur créée par le maître ou l'esclave au début de la transmission ou de la réponse, puis vérifié lorsque le message est reçu pour vérifier que le contenu est correct.

La réponse d'un esclave consiste en des champs confirmant qu'il a reçu à demande, les données à renvoyer et une erreur de vérification des données.

Aucune erreur ne se produit, la réponse de l'esclave contient les données demandées, si une erreur survient dans la requête de message reçue par l'esclave, ou si l'esclave est incapable d'effectuer l'action demandée, l'esclave retournera un message d'exception en réponse.

Le champ de vérification d'erreur du télégramme de l'esclave, permet au maître de confirmer que le contenu du message est valide.

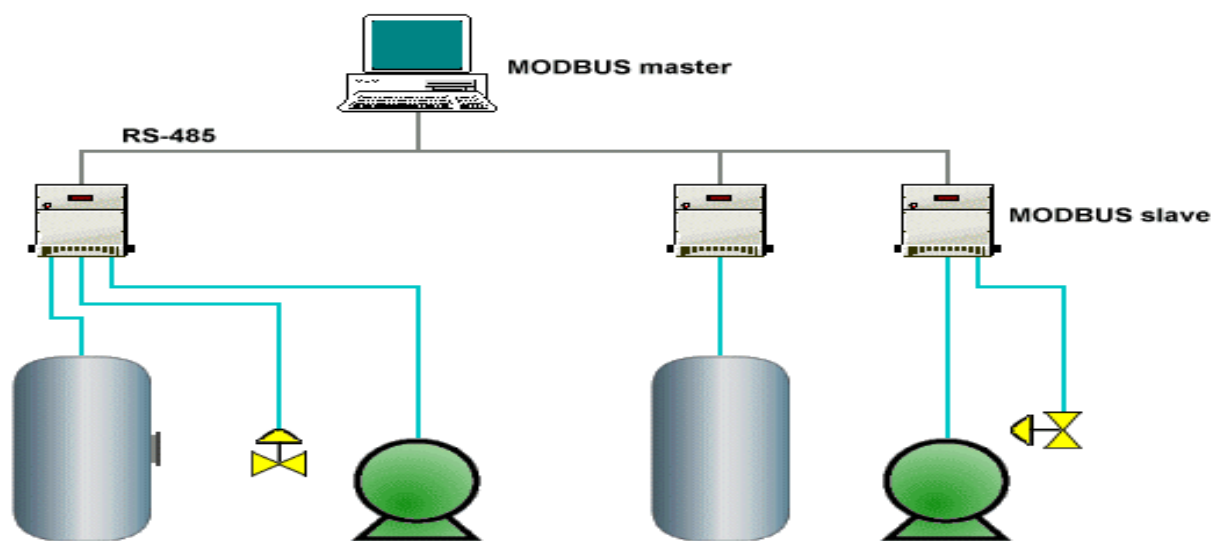


Figure I. 7 : Communication par le Protocole MODBUS.

I.3.3.2 Protocole DNP3

C'est un protocole de communication conçu pour le transfert des données et les commandes de contrôle d'un maître à un ou plusieurs dispositifs. DNP3 a été développé par Westronic Inc., en particulier pour une utilisation dans le système SCADA.

Le DNP3 gère une communication en fonction équilibrée ce qui signifie que la station maître et station esclave peuvent faire la transmission d'information. Il est construit sur le profil EPA (Enhanced Performance Architecture) qui est une version simplifiée du modèle OSI (Open System Interconnections). Il comporte 3 couches : Physique, liaison et application. Toutefois, pour permettre la transmission de messages de taille importante (2 kilo-octets ou plus), des Fonctions de segmentation et d'assemblage de données ont été ajoutées. L'ensemble de ces fonctions constitue une pseudo-couche Transport [7].

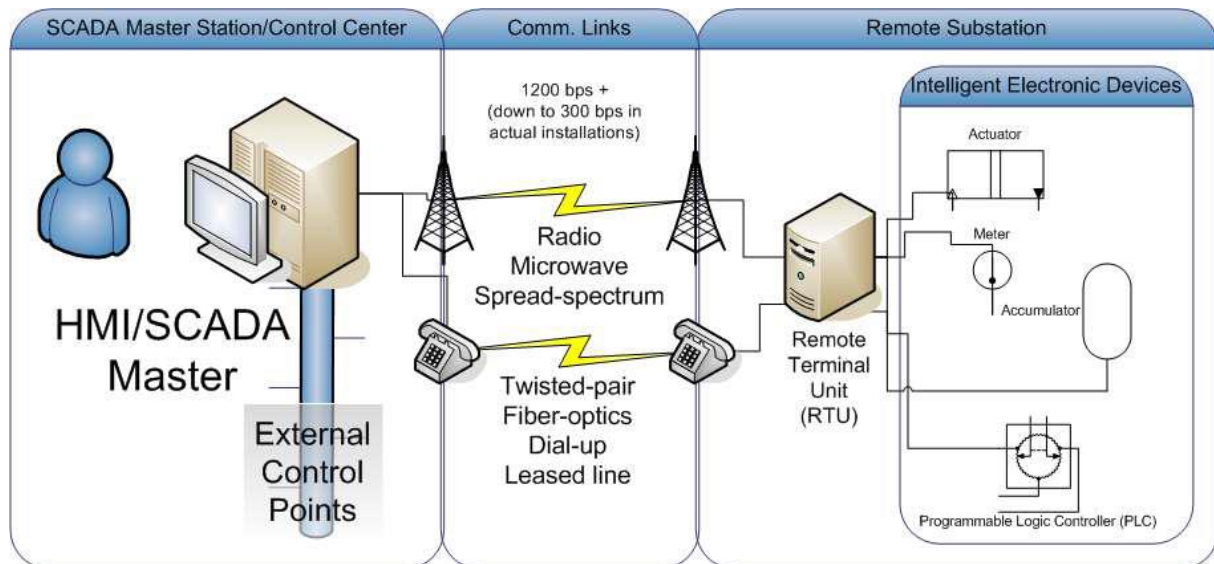


Figure I. 8 : Communication par le protocole DNP3.

I.3.3.3 Protocole PROFIBUS

PROFIBUS : (process Field bus) est un protocole pour la communication par bus de terrain en automatisation technologique, relie les systèmes d'automatisation et les contrôleurs à des appareils de terrain décentralisés tels que des capteurs, actionneurs et codeurs.

Les réseaux PROFIBUS échangent des données à l'aide d'un seul câble de bus, les utilisateurs peuvent combiner divers types de protocoles PROFIBUS avec propre logiciel et d'autres exigences, résultant en un profil d'application unique. Elle a été développée pour la première fois en 1989 par BMBF puis utilisée par SIEMENS.

PROFIBUS est aussi un protocole de type Maître-esclave comme MODBUS mais avec anneau à jeton supplémentaire qui est un protocole pour permettre à plusieurs maîtres de se connecter. Il peut emprunter trois supports de transmission ou supports physiques.

Le PROFIBUS-DP : utilisé pour faire le fonctionnement du capteur et des actionneurs via un contrôleur centralisé dans une usine de production, Application d'automatisation. Les nombreuses options de diagnostic standard, en particulier, sont centralisées ici. En générale la couleur du câble standard sera violette, mais il peut avec autre couleur.

Le PROFIBUS-PA : c'est le protocole conçu pour processus automatisés. En réalité, PROFIBUS PA est un type de profil d'application PROFIBUS DP. Il normalise le processus de transmission des données mesurées. Il a été conçu spécifique pour une utilisation dans des environnements dangereux.

Le PROFIBUS-FMS : est utilisé pour la communication non déterministe.

PROFIBUS : répond à des normes internationales unanimement reconnues. Son architecture repose sur 3 couches inspirées du modèle en 7 couches de l'OSI, la couche 1, physique, décrit les caractéristiques physiques de la transmission. La

couche 2, liaison de données, spécifie les règles d'accès au bus. Enfin, la couche 7, application, définit les mécanismes communs utiles aux applications réparties et la signification des informations échangées, la figure suivante représente l'architecture de la communication PROFIBUS. [5] [6]

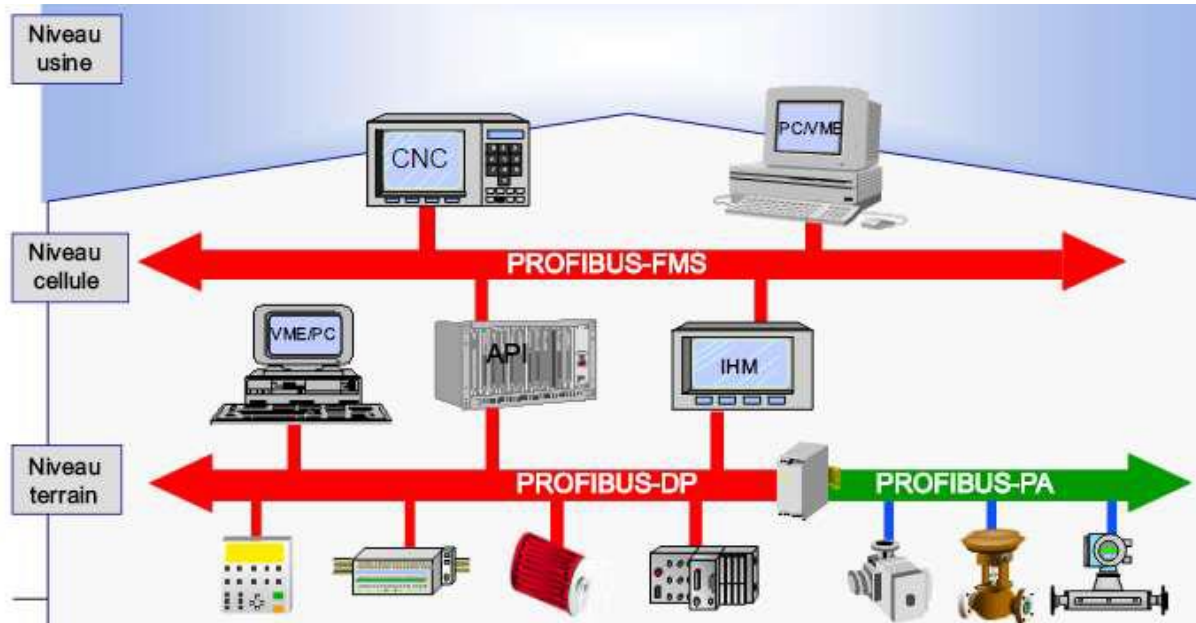


Figure I. 9 : Communication par le Protocole PROFIBUS.

I.3.4 Logiciel HIM de SCADA

HMI est l'abréviation de Humann Machine interface. Nous utilisons des IHM dans l'industrie pour contrôler et surveiller les machines. Plusieurs fois, une IHM sera la forme d'un écran, un peu comme un écran d'ordinateur, et plus de fois qu'autrement, ils sont écran tactile.

Un opérateur ou un personnel de maintenance peut utiliser et surveiller la machine à partir de l'IHM. Ils peuvent inclure des informations telles une température, la pression, les étapes du processus et le nombre de matériaux. Ils peuvent également afficher des niveaux très précis dans le réservoir et la position exacte des machines. L'information sur la machine utilisée auparavant pour plusieurs indicateurs peut désormais être visualisée sur un seul écran.

Pour confirmer la connexion HMI à la machine pour assurer le contrôle et la surveillance, il faut utiliser un logiciel spécial afin que les ingénieurs puissent les programmer correctement. Le logiciel permet à l'ingénieur de concevoir ce que l'opérateur verra réellement à l'écran, ce qu'ils peuvent surveiller à l'écran, quels boutons peuvent être enfoncés et comment l'opérateur peut manipuler la machine. La personne qui programme l'IHM doit programmer chaque indicateur et bouton à une adresse d'entrée ou de sortie spécifique d'un automate. L'interface homme-machine et l'automate doivent être compatibles, cela signifie qu'ils doivent pouvoir se parler les uns aux autres par les protocoles.

En plus l'interface graphique doit faciliter aux opérateurs toute ces tâches citées, l'HMI du SCADA est très important pour le bon déroulement de la procédure d'aide à la décision, il est le seul point d'interaction entre l'opérateur et les algorithmes d'aide à la décision. Ainsi, il aide l'opérateur dans sa tâche d'interprétation et de prise de décision, en lui offrant une très bonne visibilité sur l'état et l'évolution de l'installation, avec l'affichage en différentes couleurs des résidus, des alarmes et des propositions sur l'action à entreprendre. [8]

Les fonctions principales d'un logiciel SCADA sont les actions suivantes :

- La visualisation des données d'exploitation à travers la totalité des installations.
- L'acquisition, le stockage et l'extraction des données d'exploitation importantes avec les commentaires saisis par l'opérateur.
- La visualisation des tendances en temps réel à partir de données temps réel ou depuis les bases d'archivage.
- L'amélioration de la disponibilité des installations et la fourniture des informations fiables.
- La capture des notifications d'alarme adressées au personnel d'exploitation et de maintenance par message texte ou par voie vocale.
- La génération des rapports d'exploitation et les rapports réglementaires régulièrement.
- La gestion de la sécurité des processus et des procédés à travers l'ensemble des installations et l'administration des authentifications et les habilitations pour l'accès des personnels.

I.3.5 Supervision dans un environnement SCADA

Le système SCADA fonctionne par l'acquisition des données provenant de l'installation, Ces dernières sont affichées sur une interface graphique sous un langage très proche de Langage humain, ces opérations sont exécutées en temps réel, ainsi les systèmes SCADA Donnent aux opérateurs le maximum d'information pour une meilleure décision, ils permettent un très haut niveau de sécurité pour le personnels et pour l'installation et permettent aussi la réduction des coûts des opérations, les avantages qu'offre le SCADA sont obtenus avec la combinaison des outils softs et hard. [6]

I.3.6 Domaines d'application

I.3.6.1 Pilotage de grandes installations industrielles automatisées

- Métallurgie (laminoir) production pétrolière (distillation).
- Production et stockage agroalimentaire (lait, céréales...).
- Production manufacturière (automobile, biens de consommation...).

I.3.6.2 Pilotage d'installations réparties

- Alimentation en eau potable.
- Traitement des eaux usées.
- Gestion des flux hydrauliques (canaux, rivières, barrages...).
- Gestion de tunnels (ventilation, sécurité).

I.3.6.3 Gestion technique de bâtiments et gestion technique centralisée (GTC)

- Gestion des moyens de chauffage et d'éclairage (économies d'énergie).
- Gestion des alarmes incendies.
- Contrôle d'accès, gestion des alarmes

I.3.7 Exemple d'application de supervision sous logiciel WinCC

I.3.7.1 Contrôle de la température

La figure suivante présente l'exemple simulé pour le control de la température.

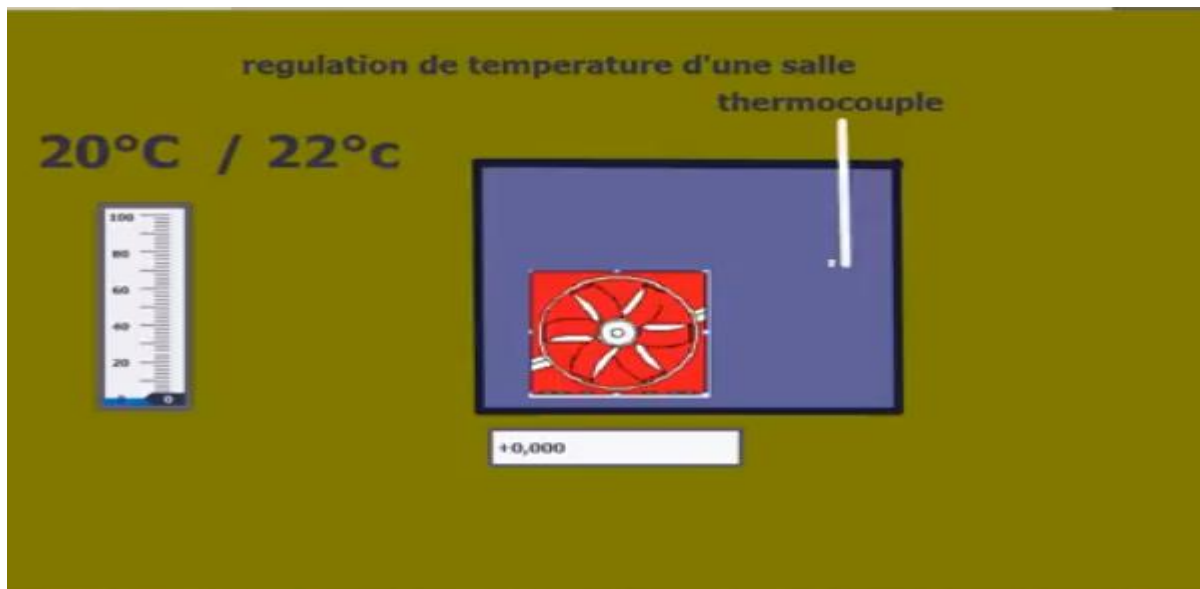


Figure I. 10 : Interface de Contrôle de la température.

Il s'agit d'un contrôle de température très simple à l'aide de WinCC où l'utilisateur peut définir la température minimum et la température maximale. Lorsque la température dépasse le niveau défini, il est également possible de connaître la condition de (HOT), (Froid).

Froid : ventilateur devient une couleur rouge.

Hot : devient une couleur verte.

Cette application propre de surveillance du système avec des seuils de température maximale et minimale.

I.3.7.2 Contrôle de niveau d'eau dans un réservoir

La figure suivante présente l'exemple simulé pour le control du niveau d'eau d'un réservoir.

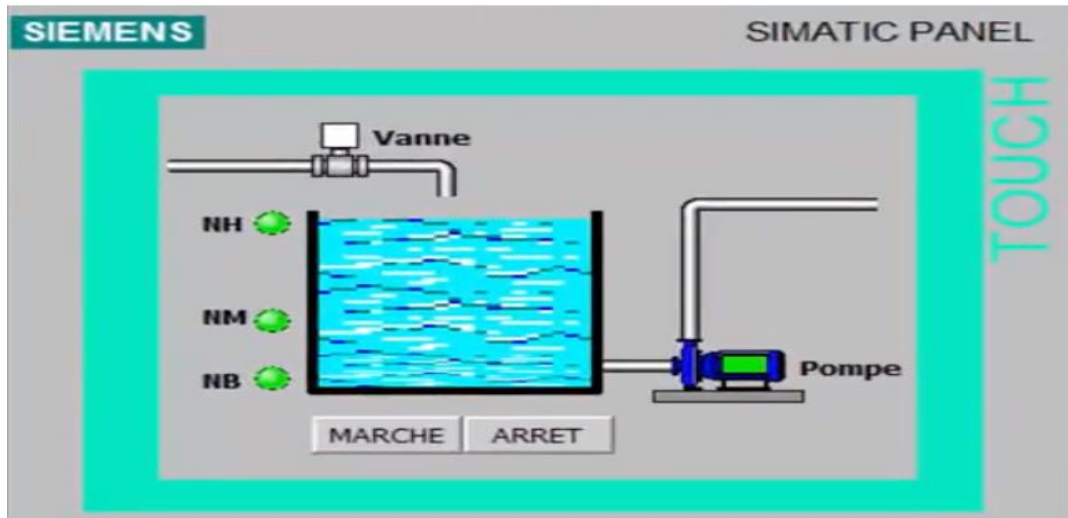


Figure I. 11 : Interface de Contrôle de niveau d'eau dans un réservoir.

Cet exemple présente le contrôle de niveau d'eau dans le réservoir simulé avec le logiciel WinCC, si vous stockez de l'eau ou divers liquides dans un réservoir ou une citerne, vous aurez sans doute besoin d'un système efficace pour réguler le niveau d'eau quand le niveau de liquide devient trop important ou est trop bas.

Cet exemple contient trois niveaux d'eau : NH (niveau haute), NM (niveau moyen), NB (niveau bas).

La Quantité de liquide augmente et atteint le niveau choisi, le régulateur sous la poussée du liquide s'incline. Grâce à une connexion aux pompes de remplissage, cela va arrêter ou enclencher le remplissage du réservoir. Le fonctionnement est le même quand le niveau est Trop bas et atteint le régulateur de niveau positionné au niveau bas. Pour vous assurer d'une bonne maîtrise du niveau de liquide, il vous faudra donc placer deux régulateurs un pour le niveau haut et un autre pour le niveau bas.

I.4 Conclusion

Dans ce chapitre, nous avons présenté la Généralité sur système automatisé et système SCADA en détail ainsi que tous les outils nécessaires pour sa mise en œuvre. Nous avons constaté aussi son importance pour le système automatisé dans le domaine industriel. L'arrêt d'un système industriel provoque des dommages énormes sur l'économie de l'entreprise, alors on peut conclure que le rôle de la supervision ne se limite pas à la sécurité, mais aussi à la continuité de la rentabilité de l'entreprise, autrement dit à la survie de l'entreprise.

Notre objectif est de mettre en place un système de supervision à distance qui offre la possibilité de commander, contrôler et modifier les paramètres en utilisant les services Web (La technologie Internet).

Chapitre II

Description des systèmes et logiciels utilisés

II.1 Introduction

La première apparition d'Automate Programmable Industriel (API) était en 1969. Son apparence a conduit à la révolution industrielle dans le monde. Tous les systèmes automatisés sont devenus contrôlés et gérés par l'API, ce qui entraîne une augmentation de la production et l'accès à des produits de haute qualité. C'était la demande et le but de toutes les usines.

Il est utilisé pour plusieurs raisons :

- Progrès scientifique et technique.
- Complexité accrue dans divers processus industriels.
- Besoin de rapidité et de précision dans les performances.
- Besoin de contrôler plus du processus en même temps.

Il existe plusieurs types d'API (s7-300, s7-400, s7-1200), et chaque type est différent de l'autre. Le type approprié est choisi à partir des plusieurs points selon les besoins requis :

- Nombre et type d'entrées / sorties
- Type de processeur
- Fonctions ou modules spéciaux
- Fonctions de communication

Dans ce deuxième chapitre, nous allons entamer en premier lieu dans une première partie la présentation du système étudié, dans une deuxième partie nous décrirons l'API S7-300, et en fin, nous aborderons quels logiciels associés seront utilisés.

II.2 Description du système étudié

II.2.1 Présentation du module « bande transporteuse MCS-700/EV »

Le module mod. MCS-700/EV a été conçu pour le transport de pièces, dans les deux directions, le long d'un axe linéaire. La bande transporteuse est mise en mouvement par un moteur CC commandé par un contacteur de marche directe/inverse pour mouvement de la bande.

Un capteur à fibre optique permet la détection du passage des pièces sur la bande transporteuse.

L'ensemble du module est monté sur une structure profilée en aluminium pouvant être facilement transportée à l'intérieur du laboratoire.

Le contrôle complet du module est effectué par un API relié au système par le biais d'un box d'interface E/S.

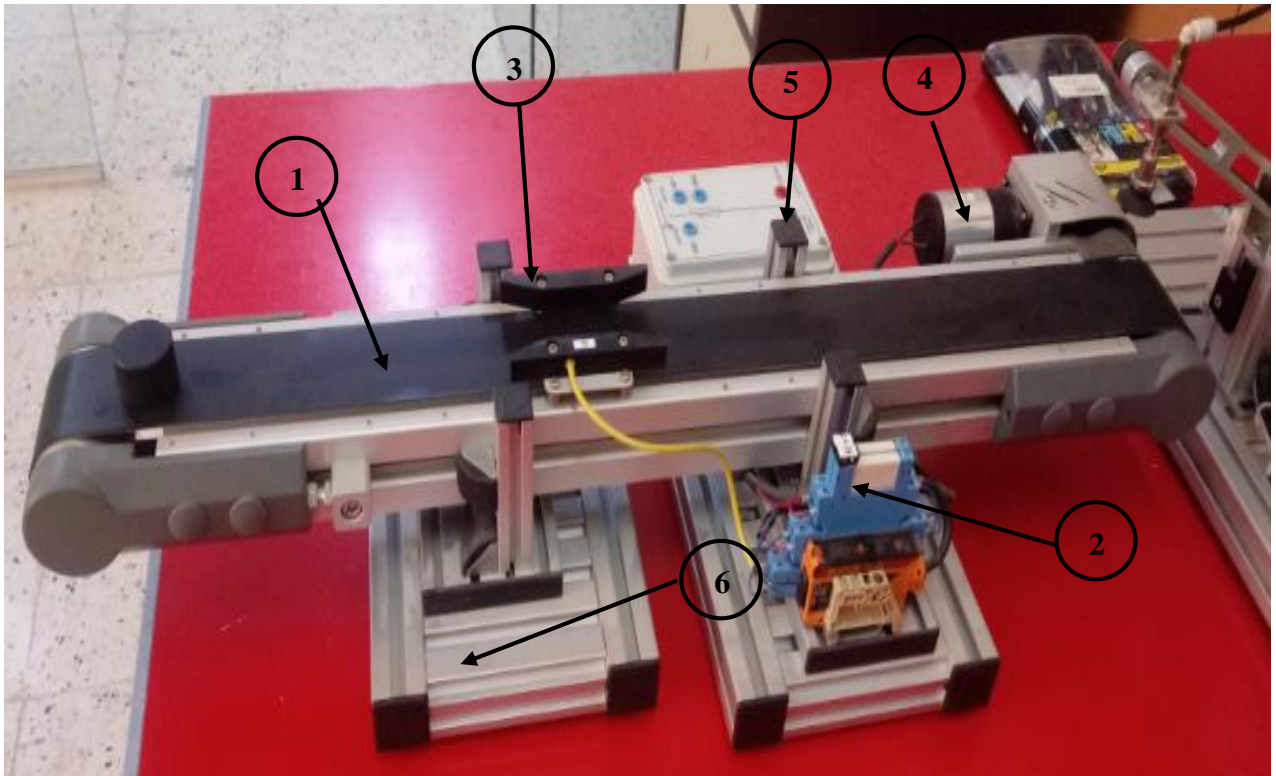


Figure II. 1 : Vue général du module Mod.700/EV.

1. Tapis roulant,
2. Contacteur de marche directe/inverse,
3. Capteur fibre optique,
4. Moteur à courant continu,
5. Box d'interface contenant les entrées/sorties du module,
6. Profilé en aluminium.

II.2.2 Présentation du module manipulateur tournant MC S-510/EV

Le module Mod. MCS-510/EV réalise la fonction de transport des pièces d'une station à l'autre. Il est constitué par un cylindre pneumatique rotatif réglable à l'intérieur d'un angle de 180°. La fonction de transport des pièces a lieu grâce à une ventouse reliée à un système de génération de vide commandé par une électrovanne 5/2 voies.

Les butées de fin de course du cylindre sont du type REED et le mouvement est contrôlé par une électrovanne à 5/3 voies, de sorte que l'on puisse arrêter le mouvement du moteur pneumatique en n'importe quelle position à l'intérieur de l'angle de rotation.

Lors de la phase de prélèvement et de déchargement, la vitesse de rotation peut être réglée au moyen des deux soupapes de décharge A et B.

Le contrôle complet du module est effectué par un API (en option) relié au système par le biais d'un box d'interface E/S.

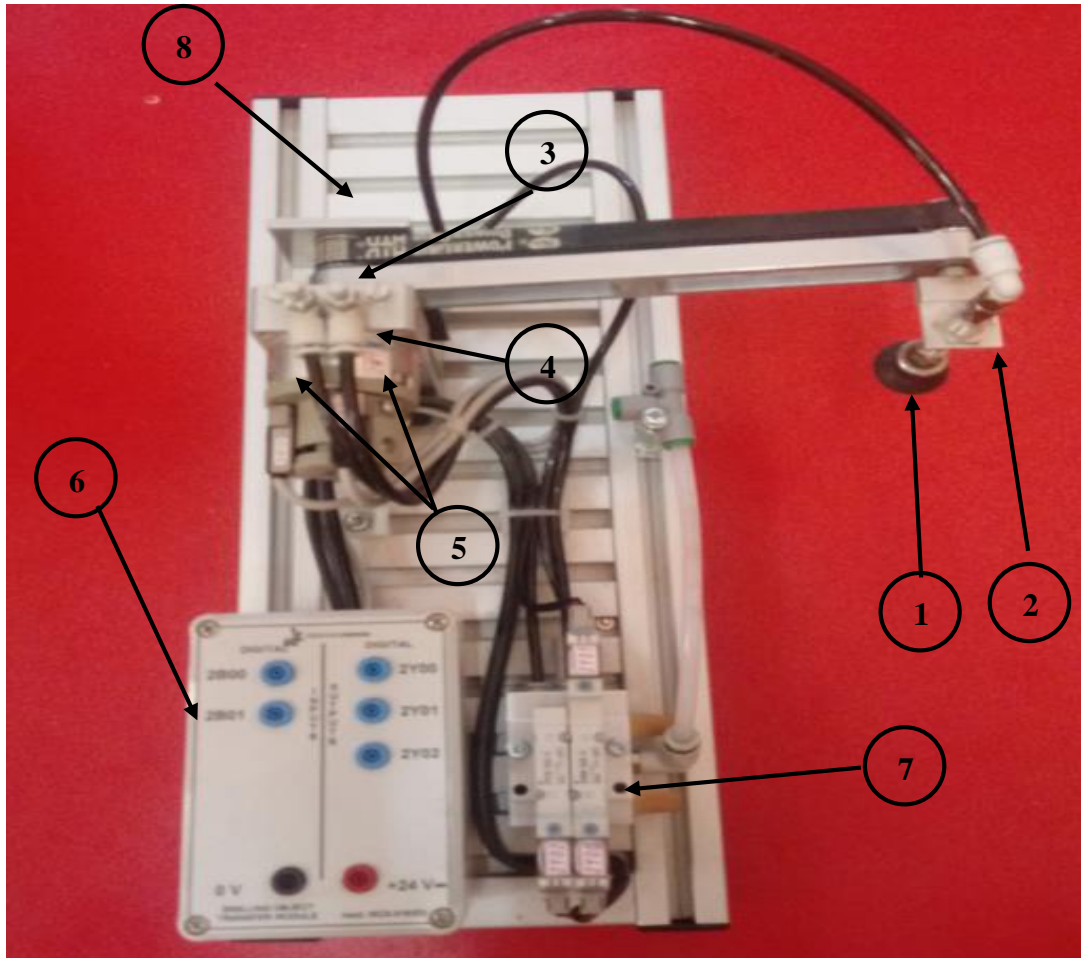


Figure II. 2 : Vue générale du module mod.MCS-510.

1. Soupape d'aspiration,
2. Bras tournant,
3. Vanne de réglage d'air (soupape de décharge A et B),
4. Moteur pneumatique,
5. Capteur de fin de course,
6. Box d'interface,
7. Groupe des vannes,
8. Profilé en aluminium.

II.2.3 Présentation du module de contrôlées des pièces MCS-570/EV

Le module de contrôle de pièces mod. MCS-570/EV est constitué par une table tournante où sont disposées les pièces à examiner. Un groupe de trois capteurs : capacitif, inductif et optique, servant pour détecter trois types de pièces présentes, noir ou blanc et métallique. Le résultat peut être observé sur l'afficheur.

La position de la table tournante est contrôlée par un capteur de proximité inductive tandis que la rotation est gérée par un petit moteur à courant continu.

Chapitre II : description des systèmes et logiciels utilisés

Les positions libres pour le prélèvement et le dépôt des pièces sont au nombre de deux.

Le capteur capacitif est placé dans le point de relâché des pièces par la station précédente et aussi peuvent être changée la position de capteur selon le module qui dépose la pièce sur la table

L'ensemble du module est monté sur une structure profilée en aluminium pouvant être facilement transportée à l'intérieur du laboratoire.

Le contrôle complet du module est effectué par un API (en option) relié au système par le biais d'un box d'interface E/S.

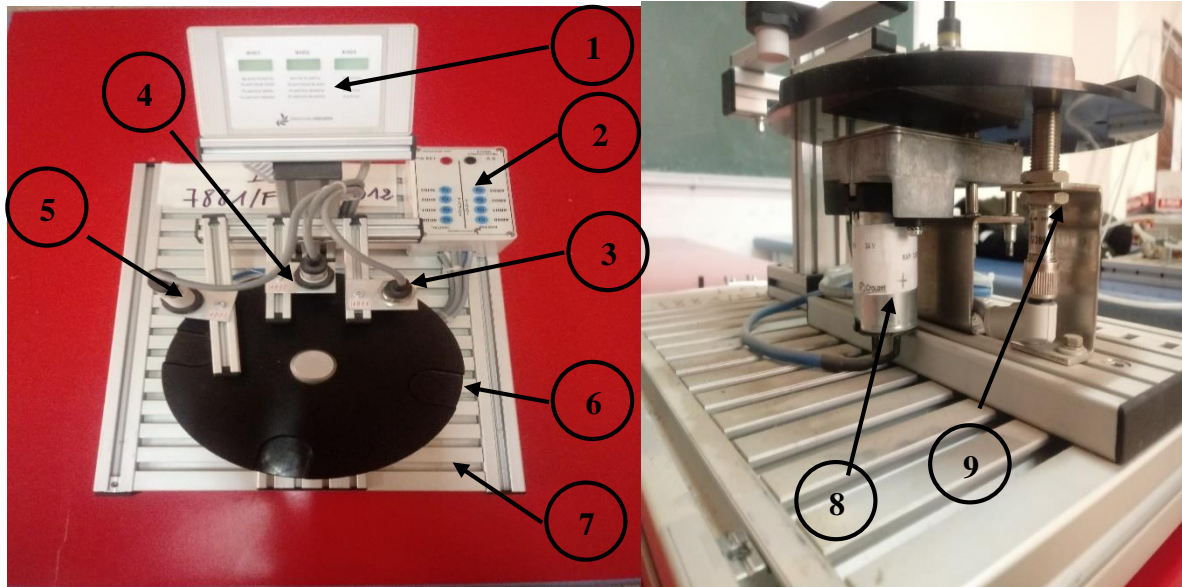
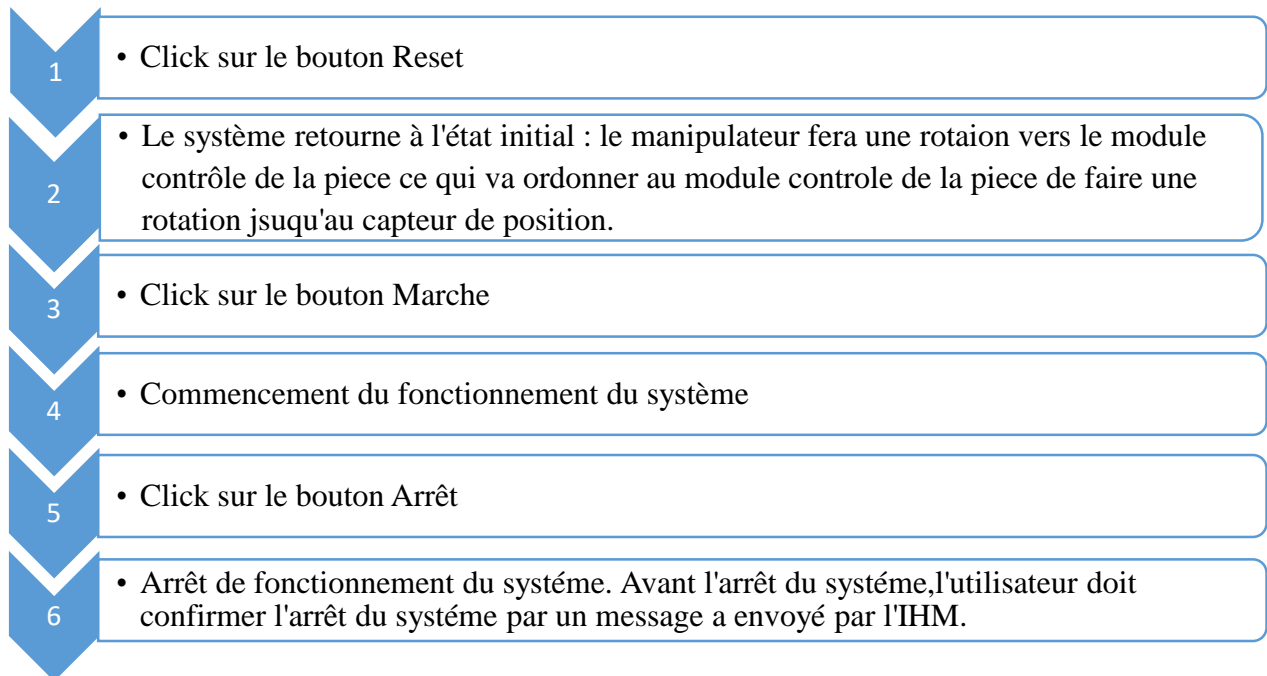


Figure II. 3 : Vue générale du module mod.MCS-570/EV.

1. Afficheur
2. Box d'interface,
3. Capteur inductif,
4. Capteur optique,
5. Capteur capacitif,
6. Table tournante,
7. Profile en aluminium,
8. Moteur à CC
9. Capteur inductif de proximité.

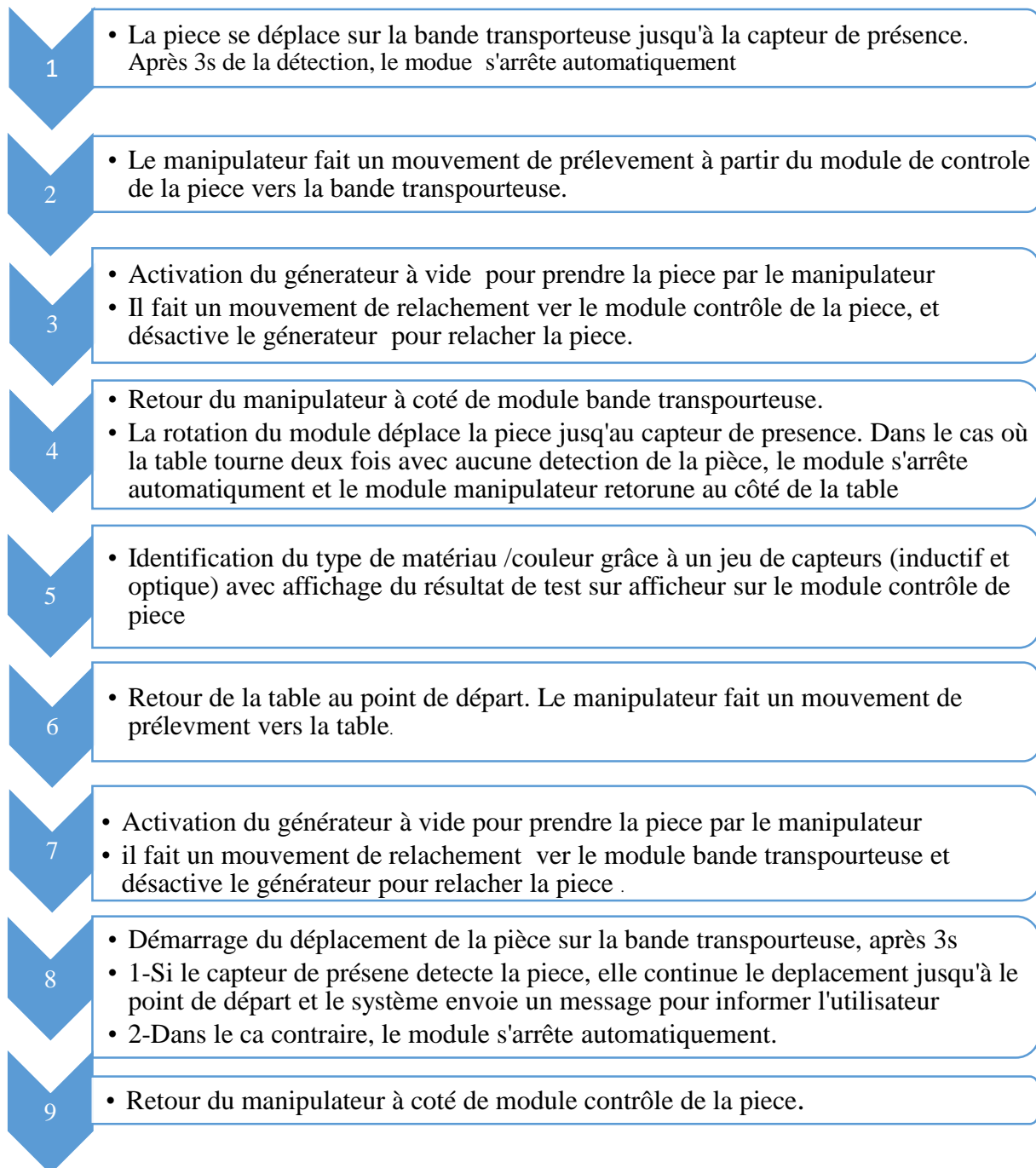
II.2.4 Fonctionnement du système

Pour commander le système, on a jugé judicieux de l'expliquer au biais de l'organigramme suivant.



Chapitre II : description des systèmes et logiciels utilisés

Le fonctionnement du système est comme suit :



II.3 Automate programmable (API)

II.3.1 Définition

Un automate programmable industriel (API) est une machine électronique, il est simplement un ordinateur qui va assurer le dialogue entre une application et une autre, spéciale dans la conduite, la surveillance et le contrôle en temps réel pour les tâches de commande dans l'environnement industriel telles que le pétrole, l'acide ou l'automobile. Chaque API est composé de trois parties principales : entrée/sortie,

Chapitre II : description des systèmes et logiciels utilisés

CPU et communication. La nature des informations peut être traitée par API de type TOR (tout ou rien) ou numérique et analogique.

Voici ce qui caractérise les API :

- Connexions directe grâce à ses entrées et sorties.
- Fabriqué spécifiquement pour résister aux changements climatiques et aux vibrations.
- Programmation et leur langage facile à comprendre.

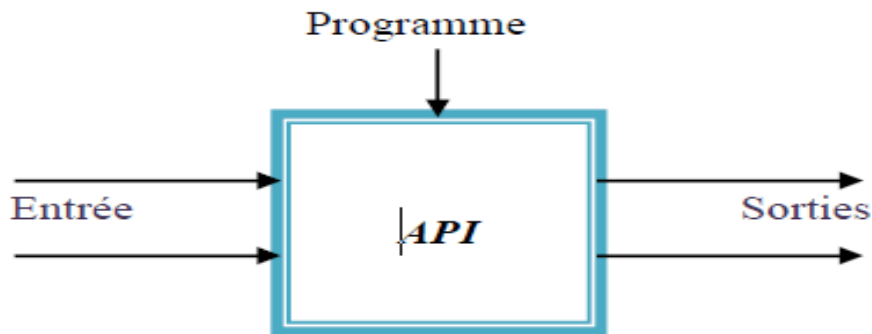


Figure II. 4 : Automate programmable industriel.

II.3.2 Type de conception d'API

Il existe deux normes mécaniques pour la conception d'API :

Boîte unique : Les caractéristiques de ce type sont résumées dans les points suivants :

- Tous les composants hardware dans un seul appareil, sauf le dispositif de programmation,
- Le nombre maximal d'entrée/sortie contenu est de 40,
- Sa mémoire est petite,
- Utilisé dans la petite application.

Modulaire et rack : Les caractéristiques de ce type sont résumées dans les points suivants :

- Toutes les unités d'API sont séparées,
- Le nombre d'entrée/sortie contenu est plus de 50,
- La possibilité d'extension de module d'entrée/sortie et de la mémoire grâce à un rack.



Figure II. 5 : Présentation d'API de type boîte unique.



Figure II. 6 : Présentation d'API de type modulaire et rack.

II.3.3 Présentation du S7-300

L'automate utilisé dans notre projet appartient à la gamme SIMATIC S7 de SIEMENS.

Le S7-300 « CPU 314-2 PN/DP », qui est un automate modulaire pour les applications d'entrée et de milieu de gamme, avec possibilité d'extensions jusqu'à 32 modules, et une mise en réseau par l'interface multipoint (MPI), Profibus et industrial Ethernet. [7]

Les caractéristiques principales des automates programmables industriels (API) Siemens S7- 300 sont les suivantes :

Il s'agit d'un matériel multiprocesseur :

- Processeur logique (bit processor),
- Processeur pour les opérations arithmétiques (word processor),
- Processeur dédié à la régulation de type PID,
- Processeur dédié à la gestion des communications.



Figure II. 7 : Présentation du S7-300.

II.3.4 Architecture Général de l'API

En général, le S7-300 contient 5 types de module principal comme les autre API.

1. **Une alimentation (PS) :** joue un rôle très important dans le fonctionnement, car elle fournit la tension nécessaire au fonctionnement du composant électronique à partir de 220V/50Hz ou dans certains cas à partir de 24V en fournissant les tensions continues $+/- 5V$, $+/-12V$ ou $+/-15V$.
Il surveille également la tension et les organise de manière à éviter toute pénurie ou augmentation de tension, et convertit également le 24v en 5v ou 12V, car certains éléments fonctionnent sous 5v ou 24v par exemple, le microprocesseur et chaque entrée/sortie à sa propre tension d'alimentation.
2. **Unité de entrée/sortie :** permet de se connecter aux dispositifs d'entrée comme les capteurs, et des dispositifs de sorties comme les moteurs. Il est considéré comme une interface entre la CPU et le processus. Ils sont définis et généralement abrégés en IOs. Ces unités peuvent être analogiques ou numériques.
3. **Moyens de communication :** ils permettent la communication de l'ensemble des blocs de l'automate et des éventuelles Extensions. Les liaisons s'effectuent :
 - Avec l'extérieur par des borniers sur lesquels arrivent des câbles transportant les signaux Électriques.
 - Avec l'intérieur par des bus reliant divers éléments, afin d'échanger des données, des Etats et des adresses.
4. **CPU :** est le cerveau principal de l'API et sa fonction est de surveiller l'état des dispositifs d'entrée et de les comparer au programme, puis de les convertir en signaux de sortie conformément au programme écrit.
5. **Mémoire :** son rôle est de recevoir, stocker et gérer toutes les unités. Ils sont divisés en trois parties principales :
 - **La mémoire morte :** (ROM : Read Only Memory) contient des systèmes d'exploitation et ne peut pas être modifié.
 - **La mémoire vive :** (RAM Random Access Memory) est une mémoire utilisée pour lire et stocker des informations et les donnée pouvant être

Chapitre II : description des systèmes et logiciels utilisés

modifiées et surveillées. Mais il est perdu en cas de manque de tension. Elles nécessitent par conséquent une sauvegarde par batterie.

- **La mémoire morte reprogrammable (EPROM : Erasable and Programmable Read Only Memory)** utilisé pour stocker le programme à exécuter.

II.3.5 Cycle d'un API [9]

Tous les automates fonctionnent selon le même mode opératoire :

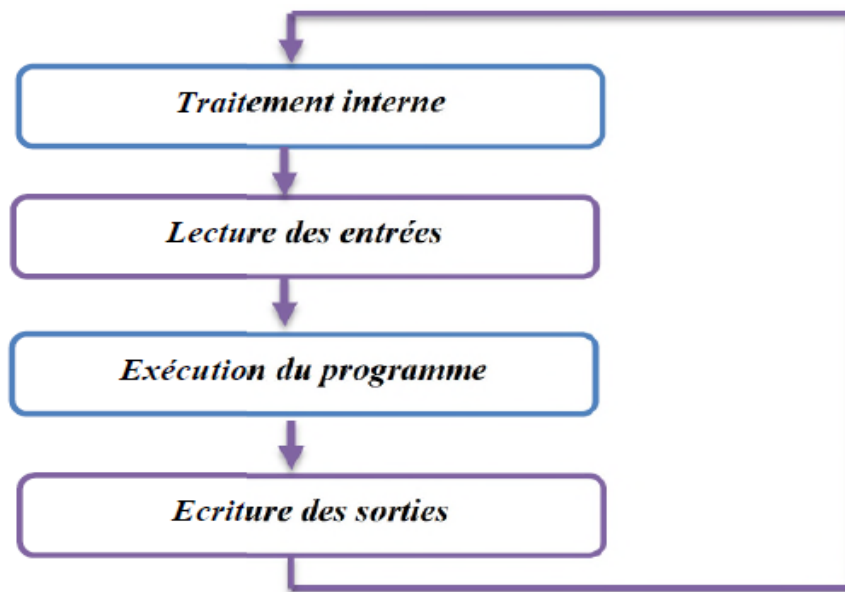


Figure II. 8 : Cycle d'un API.

- **Traitement interne** : L'automate effectue des opérations de contrôle et met à jour certains paramètres systèmes (détection des passages en RUN / STOP,...).
- **Lecture des entrées** : L'automate lit les entrées (de façon synchrone) et les recopie dans la Mémoire image des entrées.
- **Exécution du programme** : L'automate exécute le programme instruction par instruction et écrit les sorties dans la mémoire image des sorties.
- **Ecriture des sorties** : L'automate bascule les différentes sorties (de façon synchrone) aux Positions définies dans la mémoire image des sorties.

Ces quatre opérations sont effectuées continuellement par l'automate (fonctionnement cyclique).

II.4 Logiciels de développement

II.4.1 STEP 7

STEP7 SIMATIC Manager est un logiciel de base pour la configuration, la programmation, la vérification et le diagnostic des systèmes automatisés de famille

Chapitre II : description des systèmes et logiciels utilisés

SIEMENS. Elle représente une interface graphique pour programmer facilement et commodément les systèmes automatisés SIMATIC S7-300, S7-400. Ce logiciel permet de :

- Configurer et paramétrer le matériel,
- Spécifier la communication,
- Tester et mettre en service,
- Documenter et archiver,
- Diagnostiquer le fonctionnement.

STEP7 permet l'accès aux bases de données des automates SIEMENS, il permet de les programmer. Il prend également en compte le réseau des automates ce qui permet d'accéder à tout automate du réseau et permet aux automates d'envoyer les messages entre eux. [10]

II.4.2 Langages de programmation

Le logiciel Step7 permet de programmer avec plusieurs langages différents, qui peuvent être même mélangés dans un même programme.. La programmation peut être effectuée soit en utilisant le langage Grafcet directement, soit en créant une séquence d'exécution avec un Bloc d'Organisation.

Pour tous les automates, on utilise les langages suivants :

- Texte structuré (SCL),
- Schéma à contacts (CONT),
- Logigramme (LOG).

Pour notre travail, on utilise les langages pour programmer l'automate S7-300 suivants :

- Liste d'instructions (LIST)
- Programmation de chaînes séquentielles (Grafcet)

Dans notre cas, nous utiliserons le langage de programmation Grafcet pour programmer le système étudié.

II.4.3 Grafcet

Grafcet (Graphe Fonctionnel de Commande par Etape et Transition) est un graphe ou outil de description du cahier de charge de système automatisé, il est utilisé pour décrire les opérations séquentielles de la partie contrôle associée aux différentes étapes du processus du système automatisé au cours du développement de la production tout en respectant des conditions de transition d'une étape à l'autre, donc on peut dire qu'une série d'étapes temporelles consécutives liées entre elle par la transition. Ses avantages peuvent être résumés comme suit :

Chapitre II : description des systèmes et logiciels utilisés

- Il est indépendant de la matérialisation technologique,
- Il traduit de façon cohérente le cahier de charge,
- Il est bien adapté aux systèmes automatisés.

On peut modéliser Le Grafcet par les étapes suivantes :

- Il est composé de plusieurs étapes représentées par des carrés et numérotées à l'intérieur, sauf la première étape qui représente par un double carré appelé étape initiale,
- Devant chaque étape, on décrit la tâche correspondante, c'est-à-dire que fait le système dans cette étape,
- Entre deux étapes successives, on dessine un trait horizontal qui représente la transition d'une étape à la suivante,
- Devant la transition, on écrit la condition de fin de la tâche précédente appelée réceptivité,
- Relier les étapes de transition.

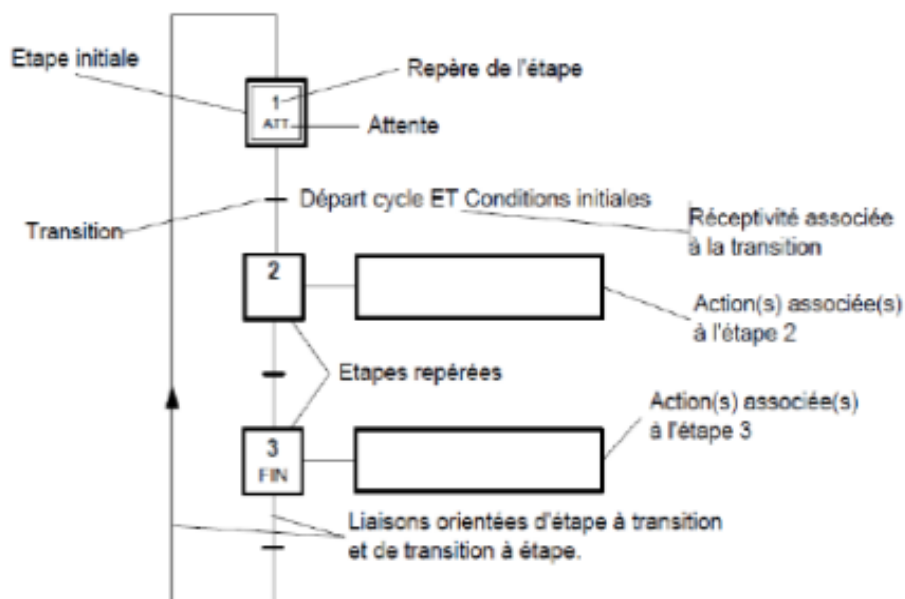


Figure II. 9 : Structure et interprétation du Grafcet.

II.4.3.1 Structure de grafcet

Le Grafcet est défini par un ensemble constitué d'éléments graphiques de base.

- Etape initiale

C'est l'étape qui est activée au démarrage du système, en générale, elle n'est pas une action associée.



Figure II. 10 : Etape initiale.

- **Etape normale**

Cette étape peut-être associé à l'activation d'une action, en générale, ce dernier représente les sorties dans le système automatisé. Ils sont l'actionneur pour exécuter le travail et les pré-actionneurs servent d'intermédiaire entre les actionneurs et les l'automatisme quand celui ne peut pas commander directement.

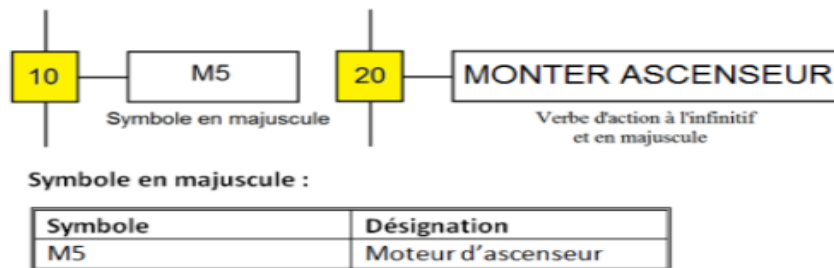


Figure II. 11 : Exemple de deux critères d'actions.

Toujours, on trouve deux cas pour une étape, active ou inactive. Si une étape est active, toutes les actions associées à celle-ci seront exécutées. Si une étape est inactive, les actions qui lui sont associées ne peuvent en aucun cas être lancées.

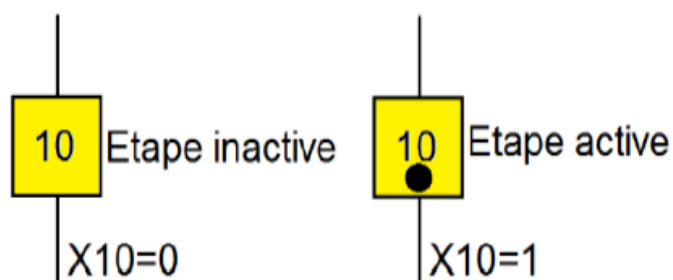


Figure II. 12 : Exemple de deux types d'étapes.

- **Transition**

Pour passer d'une étape à une autre, il faut valider la condition de la réception. On peut dire que la transition représente une porte entre deux étapes et la réception c'est la clé pour ouvrir la porte et assurer la transition vers la prochaine étape.

Chapitre II : description des systèmes et logiciels utilisés

En générale, la réception représente une entrée c'est-à-dire les capteurs. On distingue deux types de capteurs

- Le capteur actionné manuellement par exemple bouton poussoir.
- Le capteur actionné automatiquement par exemple capteur fin de course.

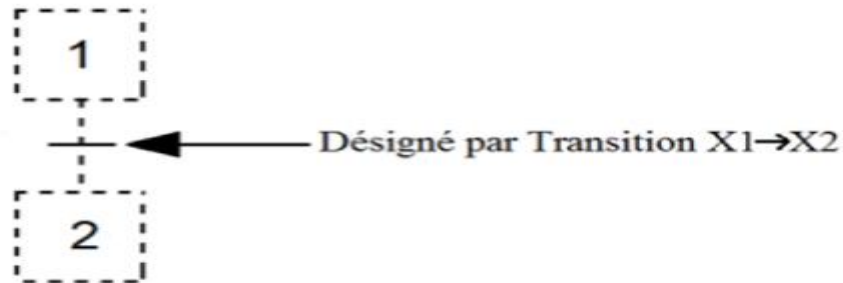


Figure II. 13 : Transitions.

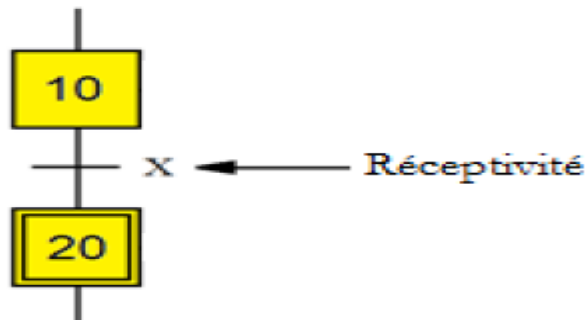


Figure II. 14 : Réceptivité.

- Liaisons orientées

Les liaisons relient les étapes aux transitions et les transitions aux étapes. Elles sont indiquées par le sens d'évolution et qui opèrent le rebouchage.

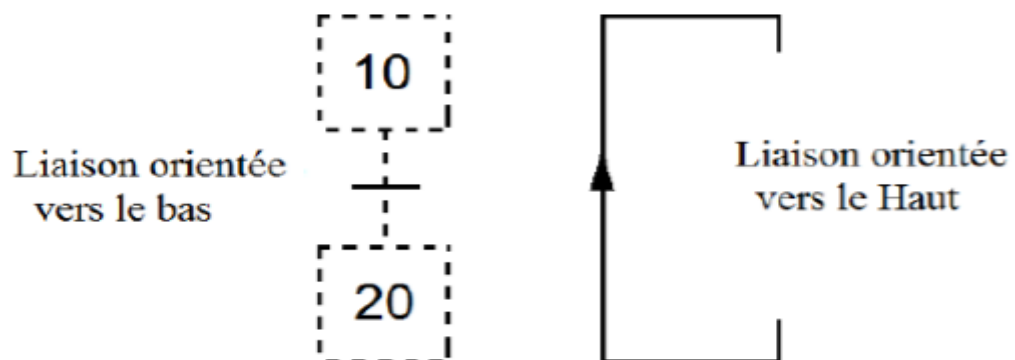


Figure II. 15 : Liaison orientée.

II.4.3.2 Règles du grafcet

1. L'étape initiale correspond à un comportement de repos du système.

Chapitre II : description des systèmes et logiciels utilisés

Principe

Au Runtime, l'opérateur peut réaliser le contrôle-commande du processus, les tâches suivantes sont alors exécutées :

- Communication avec les automates,
- Affichage des vues à l'écran,
- Commande du processus, par exemple, spécification de consignes ou ouverture et fermeture de vannes,
- Archivage des données de Runtime actuelles, des valeurs processus et événements d'alarme.

Variable

On distingue deux types de variables, les variables externes et les variables internes

- Les variables externes permettent de communiquer et d'échanger des données entre les composants d'un processus automatisé, entre un pupitre opérateur et un automate.
- Les variables internes ne possèdent aucun lien avec l'automate, elles sont enregistrées dans la mémoire du pupitre.

II.6 Conclusion

L'objectif de ce chapitre était tout d'abord de présenter le système étudié pour comprendre son rôle et son fonctionnement. Notre système étudié est géré par un API. Cette étude est focalisée sur l'automate programmable S7-300, en mettant en avant ses caractéristiques techniques. Dans ce chapitre, on a cité les logiciels utilisés pour la programmation et la supervision qui feront l'objet du chapitre III.

Quelques remarques importantes sont notées pour l'installation des deux logiciels, cela peut vous éviter la probabilité de tomber dans le cas des erreurs, donc il faut éviter :

- L'installation de WinCC avant le STEP7,
- L'installation de STEP7 v5.5.0 SP1,
- Toutes les versions des Windows sauf la version professionnelle.

Chapitre III

Développement, modélisation et conception d'IHM des modules de chaîne de stockage/déstockage en utilisant S7-300 et WinCC flexible : Cas réel au sien de laboratoire MELT.

III.1 Introduction

Comme tout système à microprocesseur, les API fonctionnent sur une base spécifique au programme qui définit les tâches à exécuter.

Pour contrôler le circuit, nous réaliserons le programme d'intégration de l'API. Pour ce faire, nous utiliserons le programme de conception de logiciel pour les systèmes d'automatisation SIMATIC STEP7.

Dans ce chapitre, nous allons décrire du programme d'automatisation développé à partir d'une analyse fonctionnelle, en plus de la supervision.

III.2 Programmation sur Step7

III.2.1 Principe

Après une compréhension d'un cahier de charge bien élaboré, pour assurer une bonne programmation de l'automatisation d'un dispositif, il est recommandé de suivre les étapes suivantes :

- Création du projet,
- Configuration, le paramétrage du matériel et de la communication réseau,
- Création de la table mnémonique,
- Élaboration du programme,
- Chargement du programme,
- Test du programme sur le simulateur PLCISIM,
- Traitement et le diagnostic des erreurs dans le cas de perturbation.

III.2.2 Démarrage de STEP 7

Pour démarrer logiciel step7, il faut cliquer deux fois sur l'icône « Simatic Manager » Figure 3.1, ce qui permet d'ouvrir la fenêtre qui représente dans la Figure 3.2



Figure III. 1 : Fenêtre SIMATIC Manager.

III.2.3 Création d'un nouveau projet

On peut créer un nouveau projet en cliquant sur suivant dans la fenêtre de démarrage (assistance de STEP7 illustré dans la Figure III.2) tout en choisissant le type de CPU sur laquelle vous travaillez (c'est-à-dire de choisir une CPU adéquate à notre automate réel). Il faut annuler la fenêtre éclaircie sur la Figure III.2 via l'appui sur Annuler. Car après une recherche sur le menu de la Figure III.3. A titre d'essai, on a appuyé sur suivant avant d'annuler, par conséquent, nous n'avons pas trouvé notre CPU adéquate.

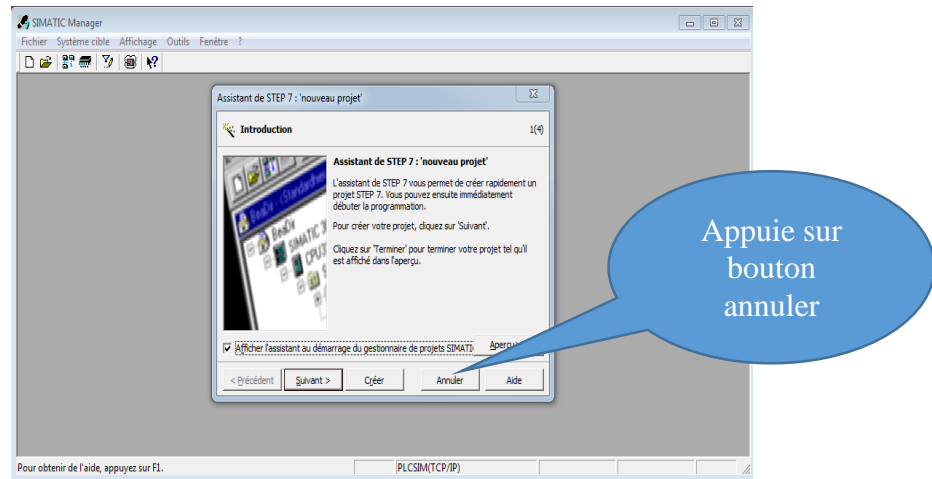


Figure III. 2 : Assistance de STEP7.

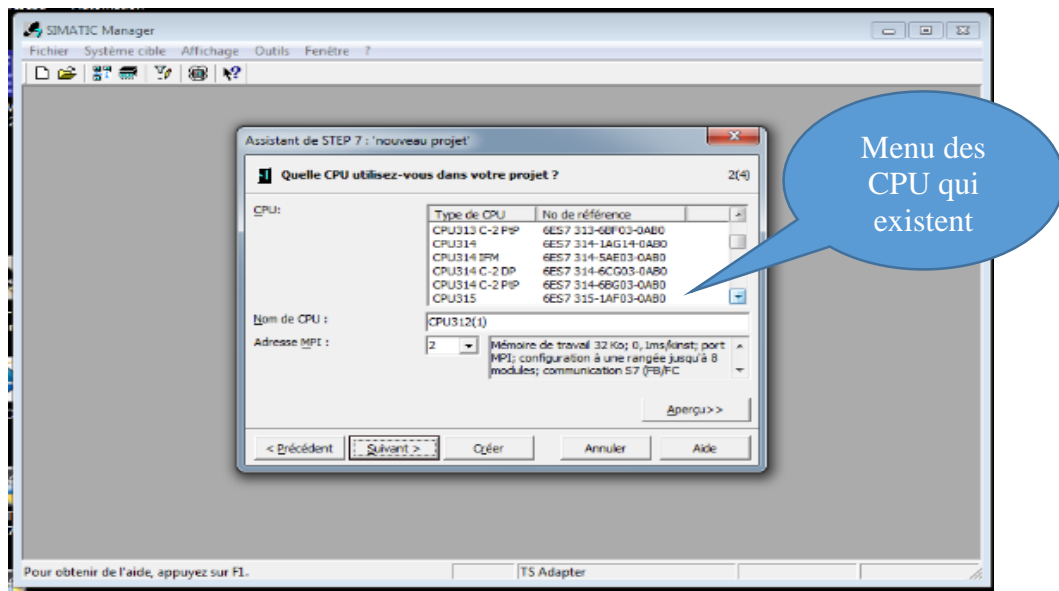


Figure III. 3 : Non existence de CPU 314 PN/DP.

Pour cela, il faut annuler l'assistant de projet et cliquer sur l'icône « fichier » dans la barre de menu. Après cette action, cliquez sur nouveau et choisir le nom de projet par exemple « projet_final ». En fin, cliquez sur ok. Cette méthode ne permet pas de choisir le type de CPU, donc il faut ajouter la configuration matérielle.

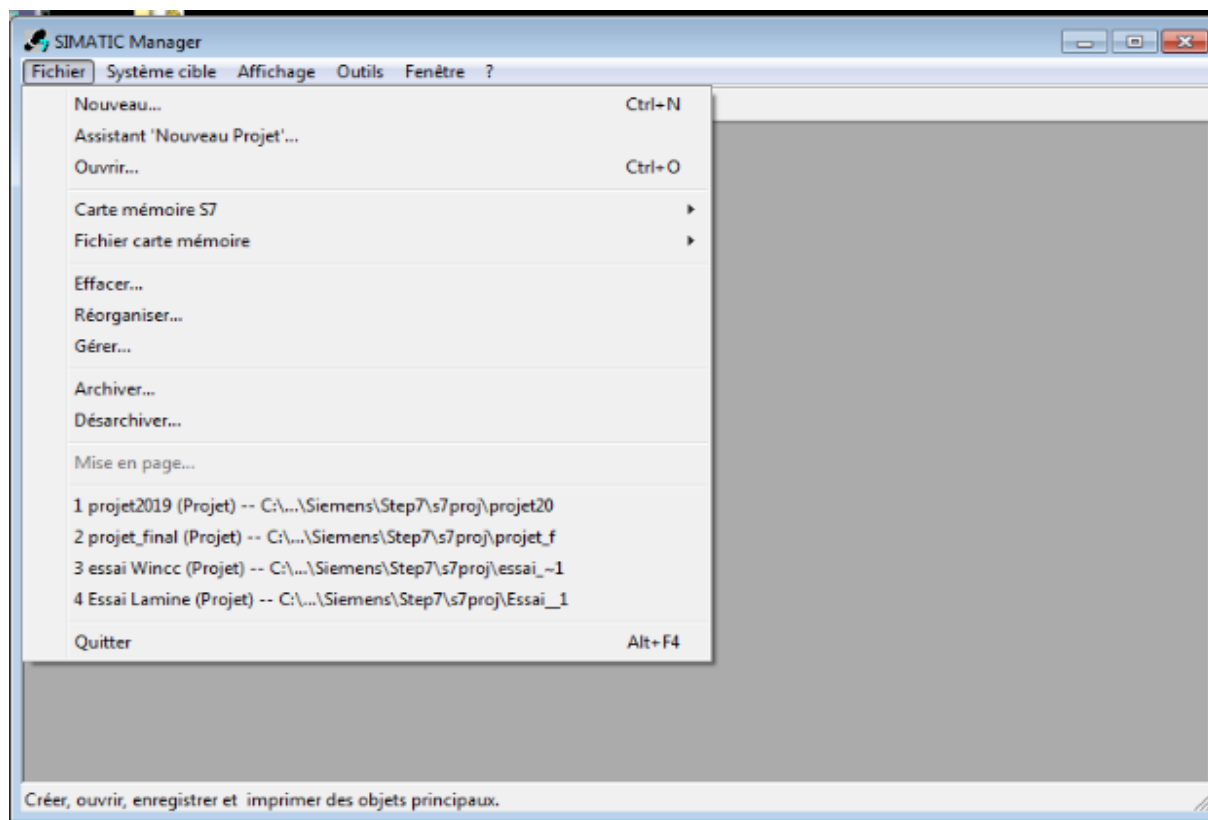


Figure III. 4 : Fenêtre pour un nouveau projet.

III.2.4 Configuration matérielle

Dans STEP7, chaque fois un nouveau projet a été créé, il faut configurer la station de travail. C'est une étape très importante qui consiste à l'agencement des châssis des modules.

Toujours, un projet contient deux grandes parties :

- Une description de matérielle,
- Description de fonction ou configuration matérielle.

La configuration matérielle est utilisée pour :

- Configurer et paramétrer le support matériel,
- Configurer la liaison entre la CPU et le console de programmation.

Pour la bonne configuration matérielle du module, vous devez suivre les étapes suivantes :

II.2.4.1 Ajout d'une nouvelle station

Pour cette étape, la Figure III.5 décrit les 03 phases indispensables.

1. Cliquer sur le bouton droit de la souris,
2. Cliquer sur « insérer un nouvel objet »,
3. Choisir « Station SIMATIC 300 » la station utilisé dans notre cas.

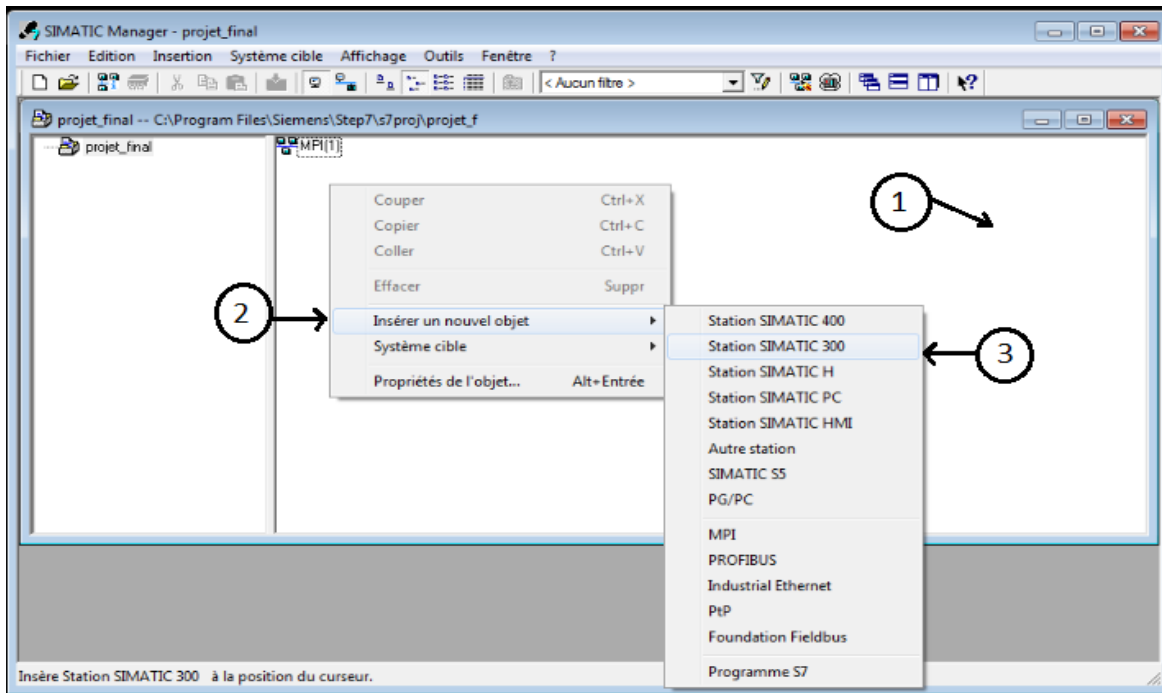


Figure III. 5 : Ajout d'une nouvelle station.

III.2.4.2 Ouverture de la fenêtre de configuration matérielle (HW Config)

Pour cette étape, la Figure III.6 montre les 03 phases à suivre.

1. Cliquer sur « SIMATIC 300 » la station déjà ajoutée,
2. Cliquer sur « Matériels »,
3. La fenêtre de configuration matérielle « le châssis » s'ouvre directement.

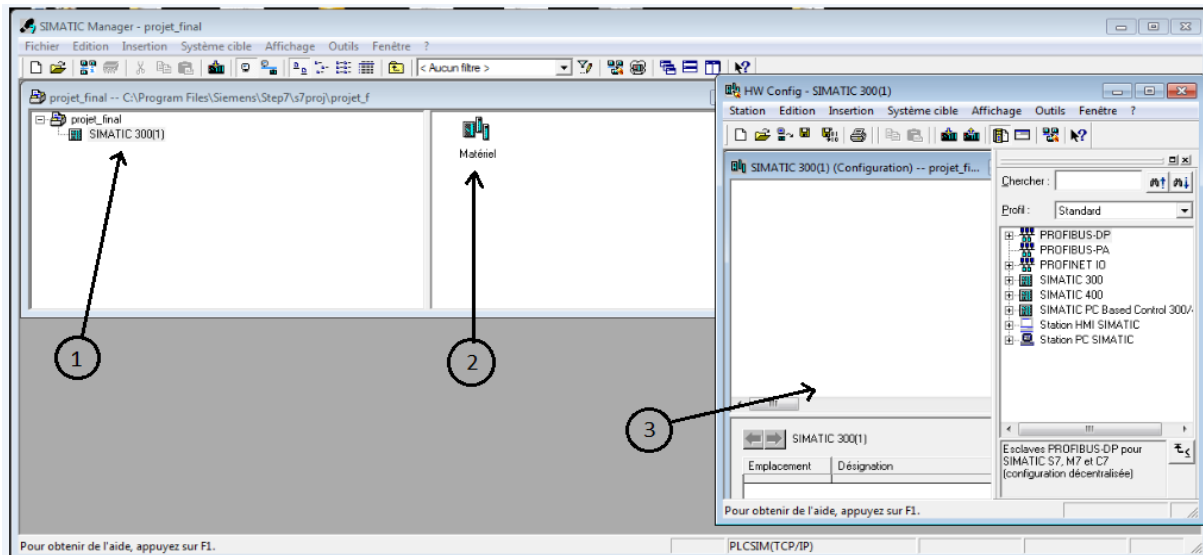


Figure III. 6 : Ouverture de la fenêtre de configuration matérielle (HW Config).

Pour configurer le matériel, on commence par le choix des châssis, le choix doit être adéquat aux caractéristiques de l'API qu'on a au sien de notre laboratoire de recherche (Alimentation, type de CPU, moyen de communication).

A : Ajout du Rack

Pour cette étape, la Figure III.7 cite les 04 phases à suivre.

- 1- Cliquer sur « SIMATIC 300 »,
- 2- Choisir « RACK-300 »,
- 3- Glisser « profilé support » vers le châssis,
- 4- Choisir le « RACK-300 ».

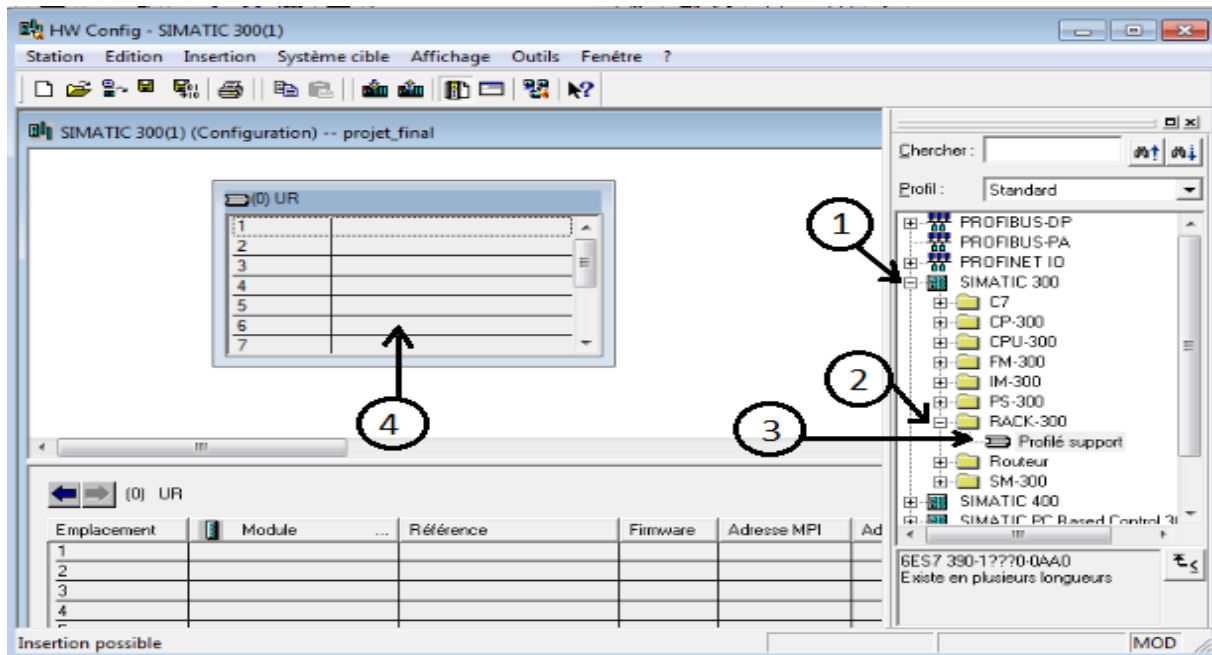


Figure III. 7 : Comment ajouter le rack.

B : Ajout d'une alimentation

Dans cette étape, la Figure III.8 décrit les 03 phases indispensables à suivre.

1. Cliquer sur «PS-300»,
2. Choisir «PS-307 5A»,
3. Glisser «PS-307 5A» dans le «rack».

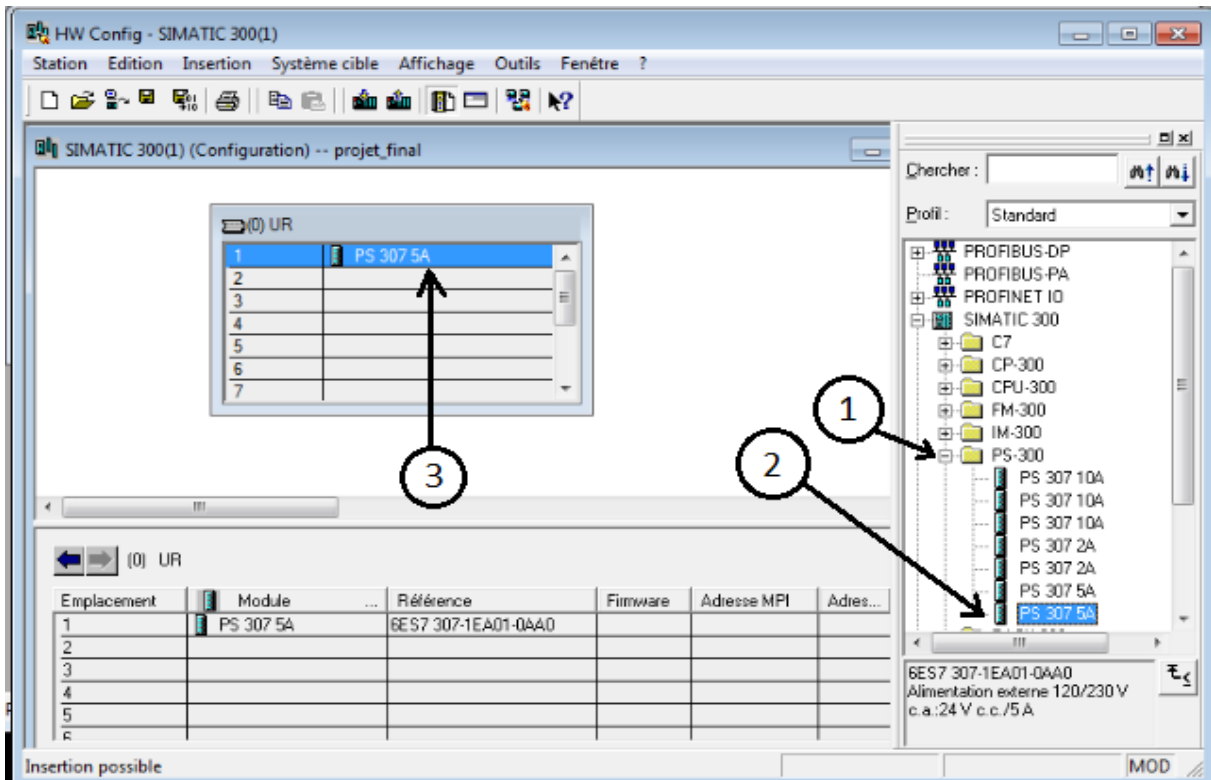


Figure III. 8 : Comment ajouter alimentation.

C : Ajout du CPU 314 PN/DP

Pour cette partie, toute la procédure est décrite dans la Figure 3.9

1. Cliquer sur «CPU-300»,
2. Choisir le type de CPU de travail «CPU 314C-2 PN/DP»,
3. Glisser «V3.3» dans le «rack» (la zone verte).

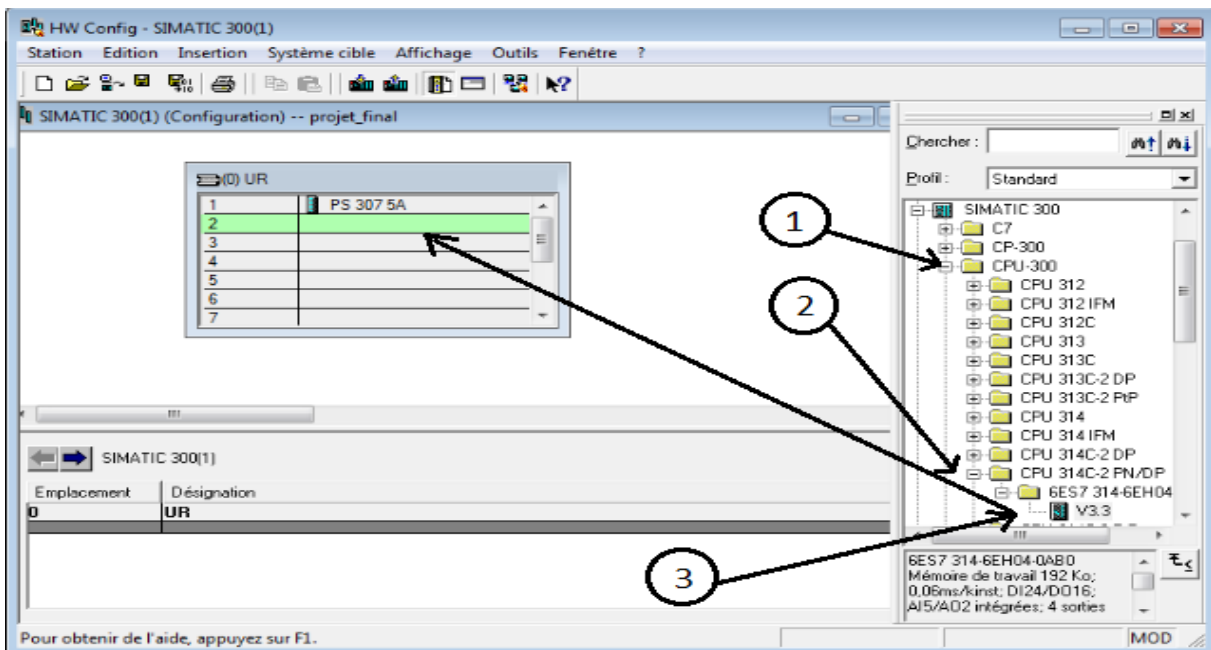


Figure III. 9 : Comment ajouter notre CPU.

D : Choix de communication par câble Ethernet

la Figure III.10 montre la meilleure procédure pour cette étape.

4. Cliquer sur nouveau,
5. Cliquer sur ok pour assurer le moyen de communication par câblé Ethernet,
6. En cliquant « ok », la CPU sera ajoutée.

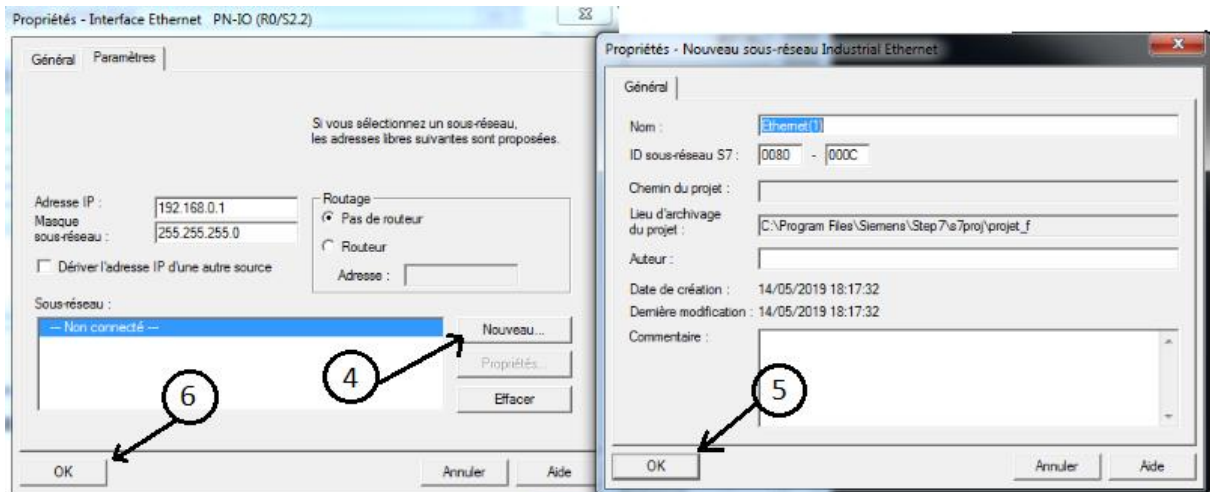


Figure III. 10 : Confirmation de communication par câble Ethernet.

D : Configuration du réseau

Dans cette étape, on va vous montrer, à travers la Figure III.11, les phases à suivre.

1. Cliquer sur «Charger dans le module»,
2. Cliquer sur «Simulation du module» pour ouvrir «PLCISIM» parce que cette étape est très importante pour que le logiciel donne la possibilité d'ouvrir le fenêtre de configuration du réseau mère,
3. Cliquer sur « NETPRO »,
4. Glisser «connexion au réseau de couleur rouge» dans le «MPI», (une ligne en rouge), cela va ajouter la possibilité de communiquer par câblé MPI,
5. Cliquer sur «enregistrer et compiler».

Remarque : la ligne verte représente la communication par le réseau Ethernet qu'on a ajoutée.

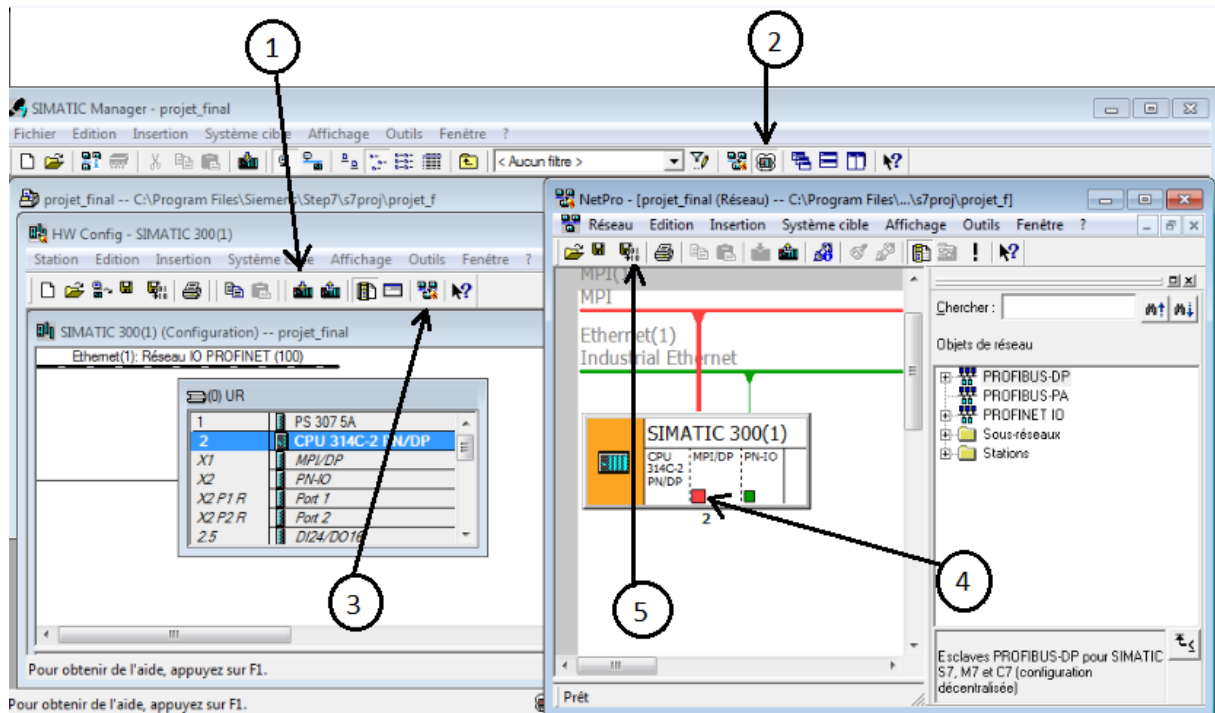


Figure III. 11 : Configuration réseau dans STEP 7.

Quand la configuration matérielle étant terminée, un dossier « programme STEP7 » est automatiquement inséré dans le projet, comme indiqué sur la figure ci-dessous.

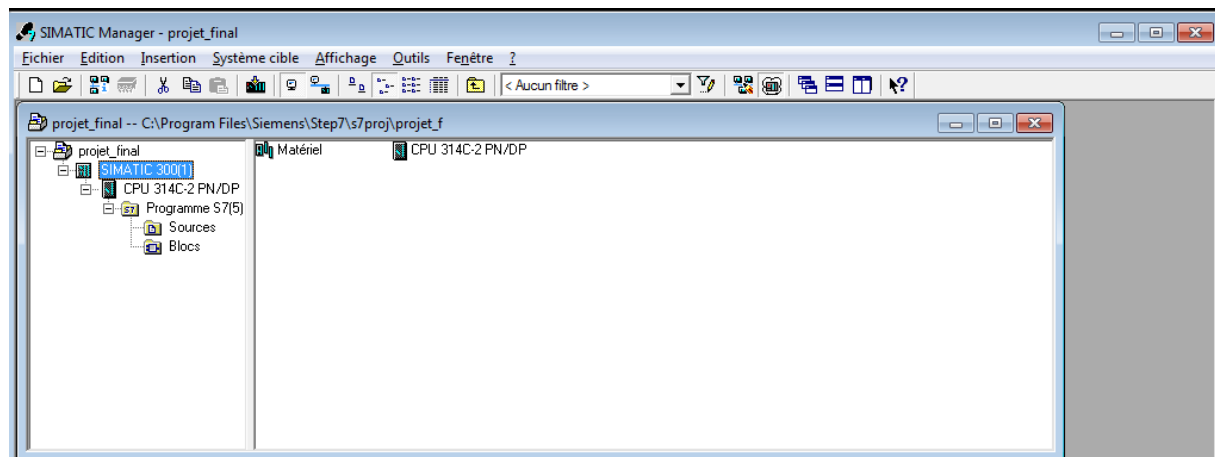


Figure III. 12 : Hiérarchie du programme STEP7.

III.2.5 Création de la table des Mnémoniques

Lorsqu'on programme dans step7, il faut qu'on définisse les variables entrées/sorties dans une table tirée de notre cahier de charge. Cette déclaration a comme but de trouver l'ensemble des variables citées dans le cahier de charge prédéfini au sein du programme.

La table mnémotique nous permet d'insérer les différents détails, le nom de la variable, son adresse adéquate, son type et en fin des commentaires.

Dans la table de mnémonique, nous introduirons différentes variables d'entrée/sortie (capteurs et actionneur et pré-actionneur) situées dans chaque module avec adresse qui nous avons utilisé pour la simulation par logiciel WinCC. Chaque variable est adressée comme suit :

- Pour les entrées (Capteurs) : $E_{n,m}$ Ou bien $I_{n,m}$
- Pour les Sorties (Actionneurs et pré-actionneur) : $A_{n,m}$ Ou bien $Q_{n,m}$

Remarque : N : signifie l'Octet (0 à 255) et M : signifier le Bit (0 à 7)

Dans ce qui se suit on va vous présenter comment insérer la table mnémonique.

- 1- Cliquer sur «Programme »,
- 2- Cliquer sur «Mnémonique».

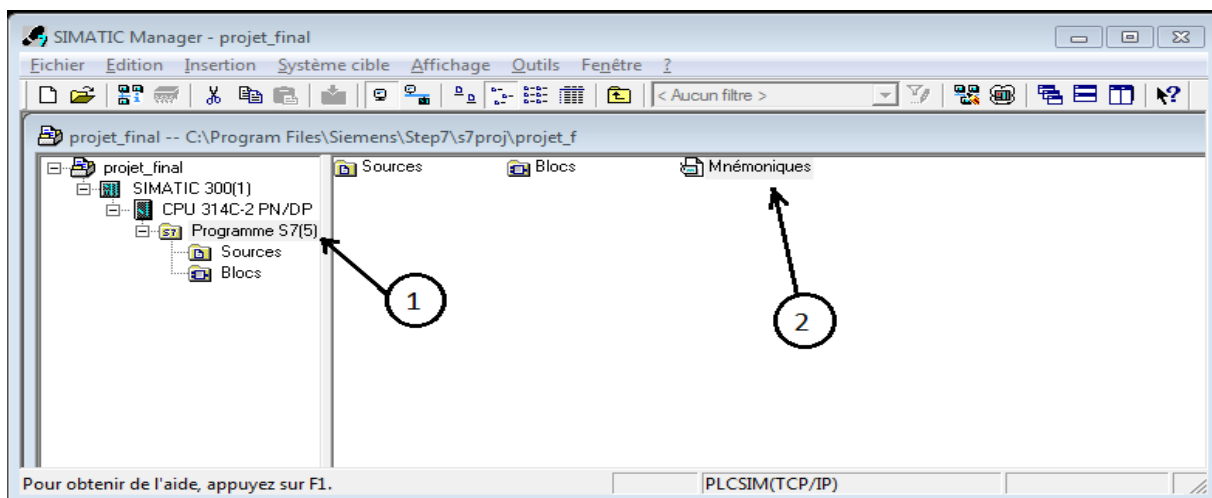


Figure III. 13 : Comment créer une table mnémonique.

Les Figures III.14, III.15 et III.16 représentent les variables utilisées dans notre programme.

The screenshot shows the 'Editeur de mnémoniques' window for 'Programme S7(1) (Mnémoniques)'. The table below lists the variables defined in the program, including their state, mnemonic name, operand, data type, and a descriptive comment.

	Etat	Mnémonique	Opérande	Type de d	Commentaire
1		2Moteur_G_Vide2	M 2.4	BOOL	moteur de générateur manipulateur
2		2Moteur1	M 2.3	BOOL	moteur CC de manipulateur
3		2Y00	Q 124.1	BOOL	préactionneur de prélèvement et relachement Droite
4		2Y01	Q 124.2	BOOL	préactionneur de prélèvement et relachement Gauche
5		2Y02	Q 124.7	BOOL	préActionneur de Générateur a vide
6		affichage de piéc...	M 5.2	BOOL	affichage de pièce en retour dans wincc
7		affichage de table	M 5.1	BOOL	variable pour l'affichage de table
8		affiche de pièce	M 5.0	BOOL	affichage de pièce dans wicc dans tapis
9		ALARM_S	SFC 18	SFC 18	Generate Permanently Acknowledged Block-Related Messages
1		ALARM_SQ	SFC 17	SFC 17	Generate Block-Related Messages with Acknowledgment
1		arrêt_2	Q 125.2	BOOL	pour l'utilisation dans wincc
1		arrêt_interface	M 0.1	BOOL	botton arrêt dans Wincc
1		arrêt_réel	I 124.1	BOOL	botton arrêt dans réel
1		B1	M 3.0	BOOL	position de BRAS sans pièce dans IHM
1		B2	M 3.1	BOOL	position de BRAS sans pièce dans IHM
1		B3	M 3.2	BOOL	position de BRAS sans pièce dans IHM
1		B4	M 3.3	BOOL	position de BRAS sans pièce dans IHM
1		B5	M 3.4	BOOL	position de BRAS sans pièce dans IHM
1		B6	M 3.5	BOOL	position de BRAS sans pièce dans IHM
2		blanc_4H02	Q 124.5	BOOL	commande DEL pièce de couleur blanc
2		C3_table	C 3	COUNTER	calcul de rotation de table
2		C4_manipulateur	C 4	COUNTER	
2		C5_manipulateur	C 5	COUNTER	
2		cap_2B00_fin de C	I 125.1	BOOL	capteur de fin course coté droite
2		cap_2B01_fin de C	I 125.2	BOOL	capteur de fin course coté Gauche
2		cap_4B00_proxi	I 125.3	BOOL	capteur de proximité:position table tournante

Figure III. 14 : Variables utilisées dans notre programme.

Chapitre III : DEVELOPEMENT, Modélisation et conception D'IHM : en cas réel

Etat	Mnémonique	Opérande	Type de d	Commentaire
2	cap_4B01_prés	I 125.4	BOOL	capteur capacitif:Présence pièce
2	cap_4B02_noir	I 125.5	BOOL	capteur optique:couleur noir de la pièce
2	cap_4B03_métal	I 125.6	BOOL	capteur inductif:pièce en métal
3	cap_tapis_19B1	I 125.0	BOOL	capteur a fibre optique dans tapis
3	couleur_pièce métal	M 80.0	BOOL	pièce métal
3	couleur_pièce noir	M 80.4	BOOL	pièce noir
3	compteur_dépl_R	C 1	COUNTER	compteur_dépl_Retour
3	couleur_pièce bl...	M 80.2	BOOL	pièce blanc
3	dépl	C 0	COUNTER	compteur de déplacement
3	dépl_décr	MW 10	WORD	déplacement de retour
3	dépl_des_pièce	MW 20	WORD	
3	déplacement_1	FC 2	FC 2	bloc FC de déplacement
3	déplacement_2	FC 1	FC 1	bloc FC pour apple des autre blocs
4	G7_STD_3	FC 72	FC 72	
4	intielisation de _C	M 1.7	BOOL	initialisation de compteur de déplacement
4	M	M 200.0	BOOL	
4	m100	M 100.1	BOOL	pour l'a fffichage de la pièce dans wincc
4	m90_B	M 90.5	BOOL	pour l'a fffichage de la pièce blanc dans wincc
4	m90_M	M 90.3	BOOL	pour l'a fffichage de la pièce Métal dans wincc
4	m90_Mé	M 90.4	BOOL	pour l'a fffichage de la pièce Métal dans wincc
4	m90_N	M 90.0	BOOL	pour l'a fffichage de la pièce noir dans wincc
4	m90_NN	M 90.1	BOOL	pour l'a fffichage de la pièce noir dans wincc
4	manipulateur ba...	FB 2	FB 2	
5	march_interface	M 0.0	BOOL	botton marche dans Wincc
5	marche_réel	I 124.0	BOOL	botton marche dans réel
5	métal_4H03	Q 124.6	BOOL	commande DEL pièce en métal

Figure III. 15 : Suite des variables utilisées dans notre programme.

Etat	Mnémonique	Opérande	Type de d	Commentaire
5	MMM	MB 40	BYTE	
5	moteur_table_4K00	Q 124.3	BOOL	commande de moteur CC table
5	Moteur_tapis	M 0.3	BOOL	moteur CC de bande transporteuse a deux sens
5	msg de validation	Q 124.0	BOOL	message de validation que la pièce est contrôlée
5	N° des Pièce	C 2	COUNTER	compteur de calcule des pièce
5	noir_4H01	Q 124.4	BOOL	commande DEL pièce de couleur noir
5	P1	M 1.0	BOOL	position de pièce et manipulateur dans wincc
6	P2	M 1.1	BOOL	position de pièce et manipulateur dans wincc
6	P3	M 1.2	BOOL	position de pièce et manipulateur dans wincc
6	P4	M 1.3	BOOL	position de pièce et manipulateur dans wincc
6	P5	M 1.4	BOOL	position de pièce et manipulateur dans wincc
6	P6	M 1.5	BOOL	position de pièce et manipulateur dans wincc
6	Piece color	MB 80	BYTE	couleur des pièces dans wincc
6	pièce_vers_D	Q 125.0	BOOL	contacteur de sens direct en tapis 19k1
6	pièce_vers_G	Q 125.1	BOOL	contacteur de sens direct en tapis 19K2
6	remise_à0_C0	M 1.6	BOOL	remis a zéro le compteur de déplacement
6	reset_lamp	Q 125.3	BOOL	rest lamp dans wincc
7	reset_system	M 0.2	BOOL	restar le systeme
7	reset_système	I 124.2	BOOL	boutton restar le système
7	table_controle	FB 3	FB 3	
7	tapis_G>D	FB 1	FB 1	bloc FB1 de tapis sens direct
7	TIME_TCK	SFC 64	SFC 64	Read the System Time
7	vue_1	M 40.1	BOOL	affichge de vue_1
7	vue_2	M 40.2	BOOL	affichge de vue_2
7	vue_3	M 40.3	BOOL	affichge de vue_3
7	vue_principale	M 40.0	BOOL	affichage de vue _principal

Figure III. 16 : Suite des variables utilisées dans notre programme.

III.2.6 Elaboration du programme

La programmation dans STEP7 utilise plusieurs types de blocs dans le cas où le programme utilisateur et les données correspondantes peuvent être mémorisées. Selon l'exigence de processus, on crée le programme par rapport aux modules utilisés dans notre projet, après l'étude du cahier de charge de chaque module, on peut structurer le programme en différents blocs.

Dans notre programme, on a créé des blocs qui sont réparties comme suit :

FC 1 : le déplacement de la pièce droite vers la gauche.

FC 2 : le déplacement de la pièce gauche vers la droite.

FB 1 : la programmation de module bande transporteuse, contient 3 graphes :

- Graphe1 : l'aller de la pièce
- Graphe2 : graph de retour de la pièce
- Graphe 3 : graph principale de FB1, graphe de commande (marche, arrête, reset)

FB 2 : la programmation du module manipulateur tournant, contient 4 graphes :

- Graphe1 : fonctionnement de bras dans étape l'aller
- Graphe2 : fonctionnement de bras dans étape retour
- Graphe 3 : responsable de l'affichage du bras.
- Graphe 4 : graph principale de fb2, dédiée à la commande (marche, arrête, reset)

FB 3 : la programmation du module table de contrôle de la pièce, contient 3 graphes :

- Graphe1 : contrôle de la pièce.
- Graphe2 : rotation de la table.
- Graphe 3 : graph principale dans FB 3, dédiée à la commande (marche, arrête, reset)

Remarque : Tout le système contient le même bouton de marche, arrêt, reset.

On peut faire un seul programme dans un seul bloc FB, mais on a choisi d'utiliser 3 programmes dans 3 modules pour faciliter la compréhension et pour permettre aux générations futures de développer le programme.

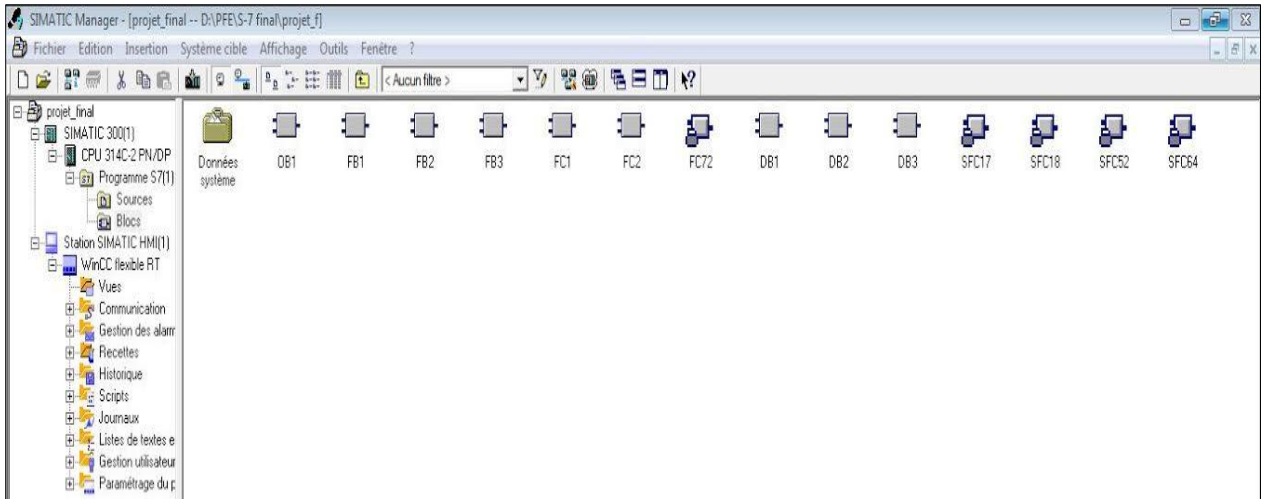


Figure III. 17 : Différents blocs de programmation utilisés.

III.2.6.1 Bloc de programmation

A : Création du bloc Fonction FC

Le rôle de (FC) assure une fonctionnalité spécifique du programme. Ils peuvent être paramétrables. Dans ce cas, des paramètres sont transmis à la fonction lorsqu'elle est appelée. Les FC peuvent être utilisés pour la programmation de fonctionnalités récurrentes et complexes, par exemple pour effectuer des calculs.

On a utilisé les blocs FC juste pour assurer le déplacement de la pièce dans WinCC, ils sont établis comme suit :

1. Cliquer sur «insertion»,
2. Cliquer sur «BLOC 7»,
3. Cliquer sur «BLOC 7».

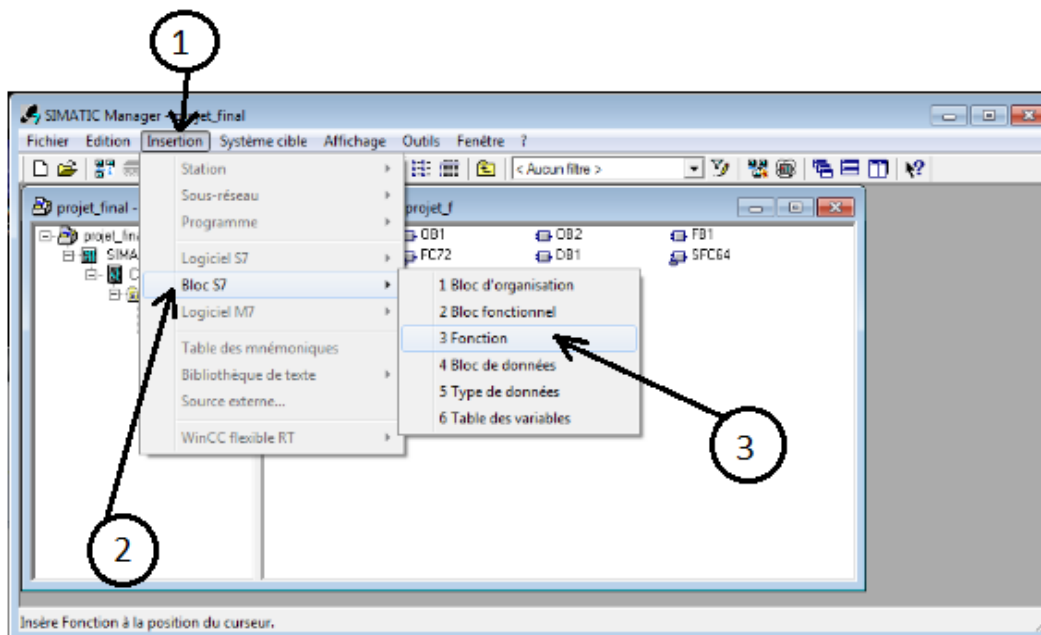


Figure III. 18 : Comment crée un bloc FC.

Les figures suivantes représentent un aperçu sur le programme dans le bloc FC.

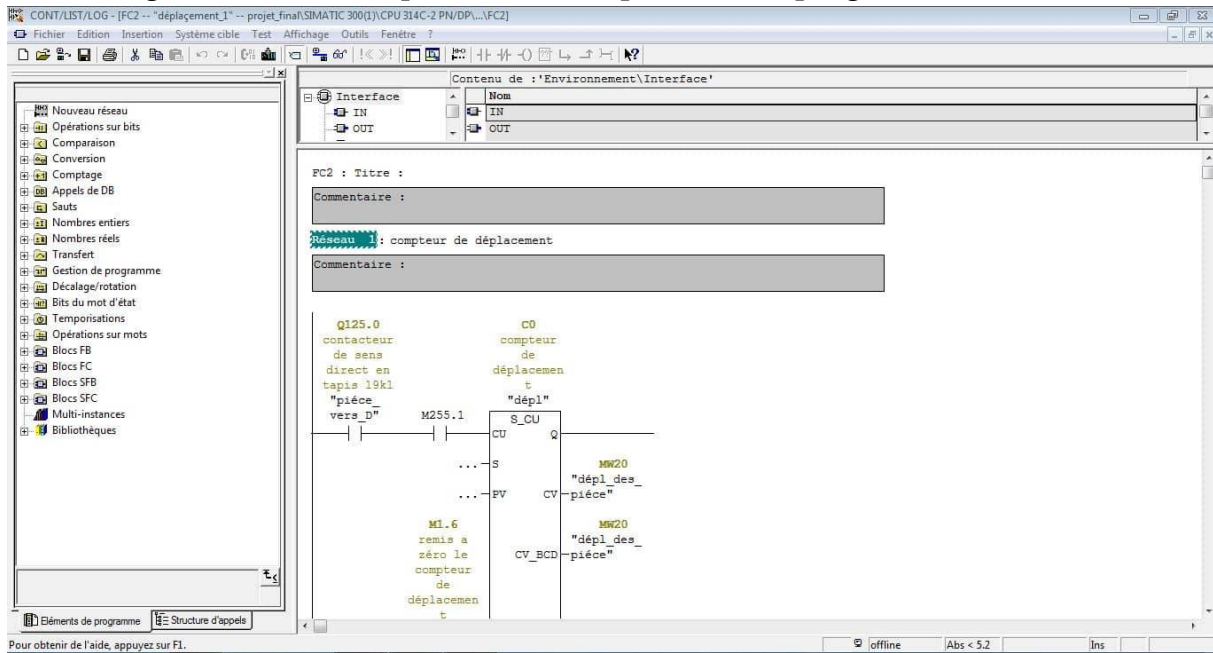


Figure III. 19 : Déplacement de la pièce droite à gauche.

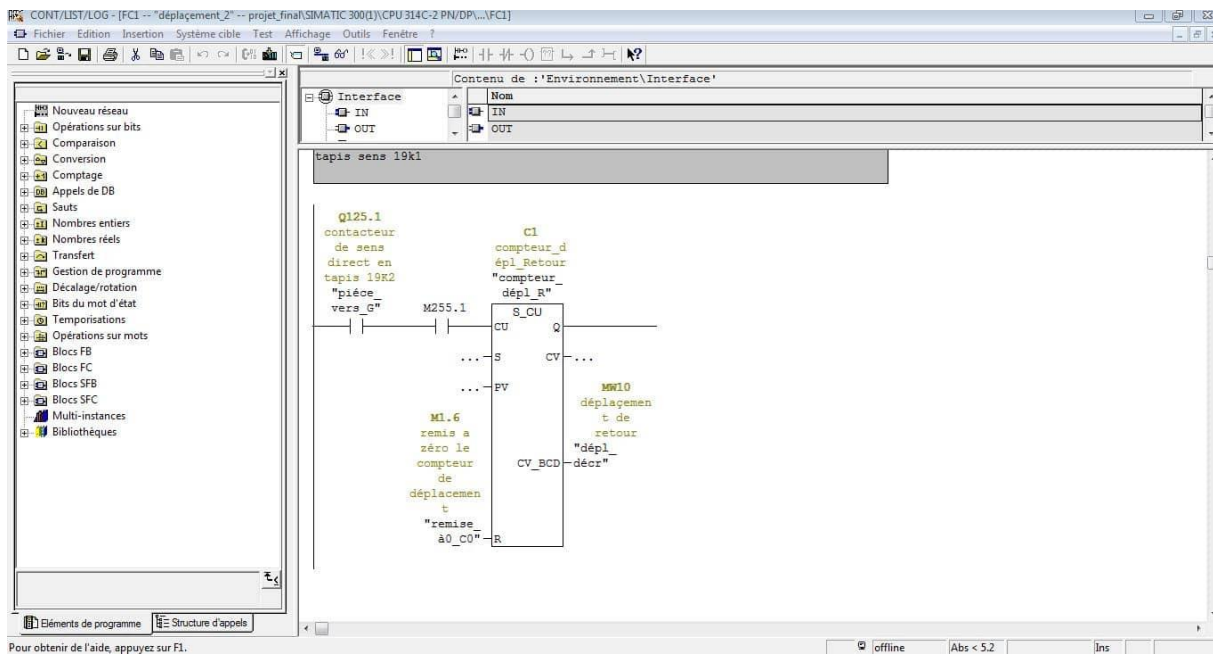


Figure III. 20 : Déplacement de la pièce gauche à droite.

B : Utilisations le memento de cadence

Dans cette fonction, nous allons utiliser 'memento de cadence' pour assurer le déplacement de la pièce dans WinCC par un compteur incrémenté, l'incrémentation peut être géré par un contact d'impulsion (memento de cadence).

Le memento de cadence est un octet, chacun des bits de cet octet change d'état de vitesse de la pièce dans WinCC. La Figure 3.21 suivant représente la référence entre les bites.

Bit	7	6	5	4	3	2	1	0
Durée de période (S)	2	1.6	1	0.8	0.5	0.4	0.2	0.1
Fréquence(Hz)	0.5	0.625	1	1.25	2	2.5	5	10

Tableau 1 : Différence entre les bits.

Remarque : Chaque bit a une fréquence prédéfinie. Plus le numéro de bit augmente, plus la cadence diminue. En d'autres termes, la fréquence est inversement proportionnelle au numéro de bit. Par conséquent, le bit 0 fournit une cadence la plus élevée.

Sa configuration est faite comme suite :

1. Double cliquer sur «CPU 314 » dans le rack,
2. Choisir «Mémento de cadence »,
3. Cocher dans la barre «Mémento de cadence »,
4. Choisir une adresse de l'octet. Dans notre travail, on a choisi l'adresse c'est 255.

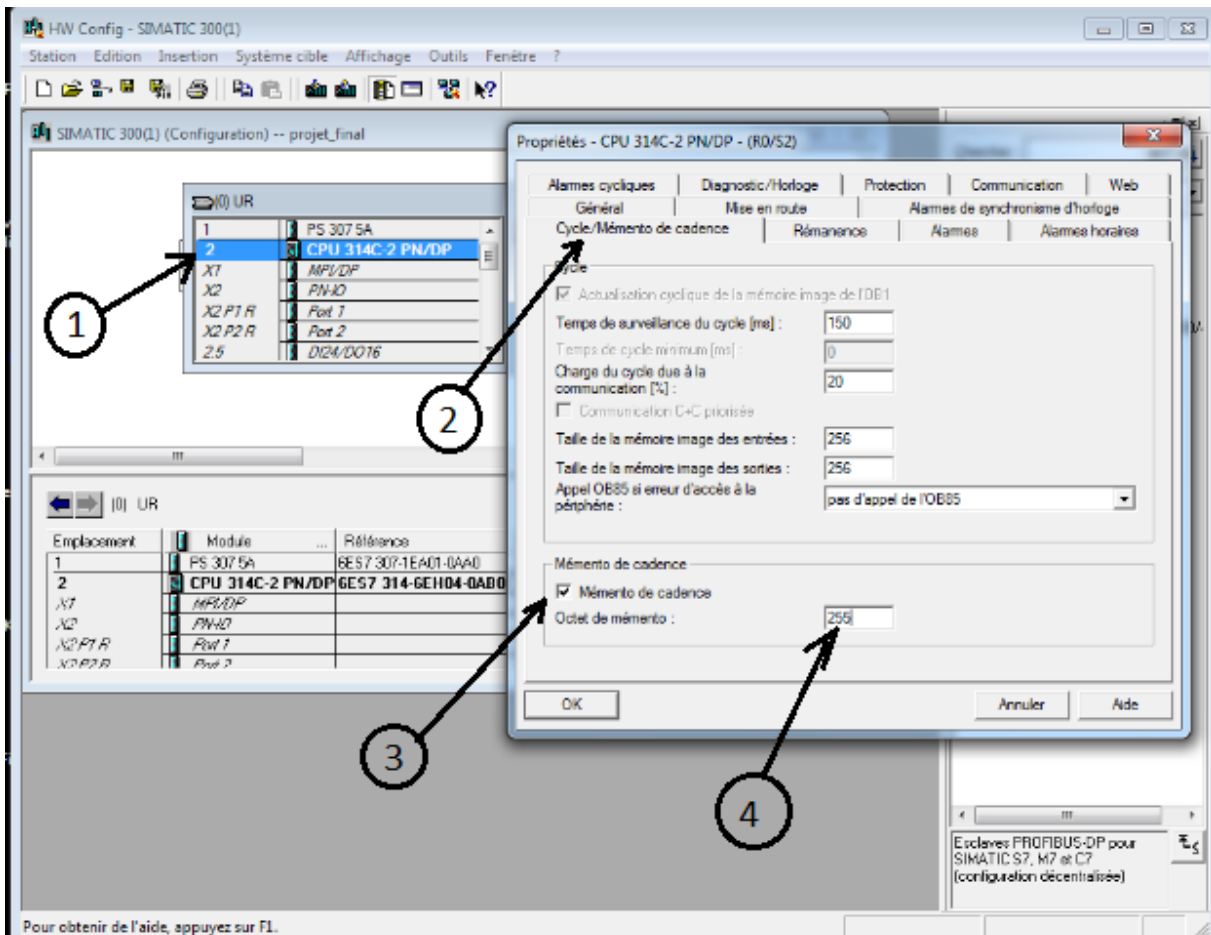


Figure III. 21 : Comment ajouter un Mémento de cadence.

C : Créations du bloc FB

Le bloc FB utilise le langage de programmation grafcet, et ce dernier a été utilisé pour programmer et exécuter le système étudié dans notre travail. La figure suivante représente comment créer un bloc FB

1. Cliquer sur «insertion»,
2. Cliquer sur « Bloc7 »,
3. Choisir « Bloc fonctionnel ».

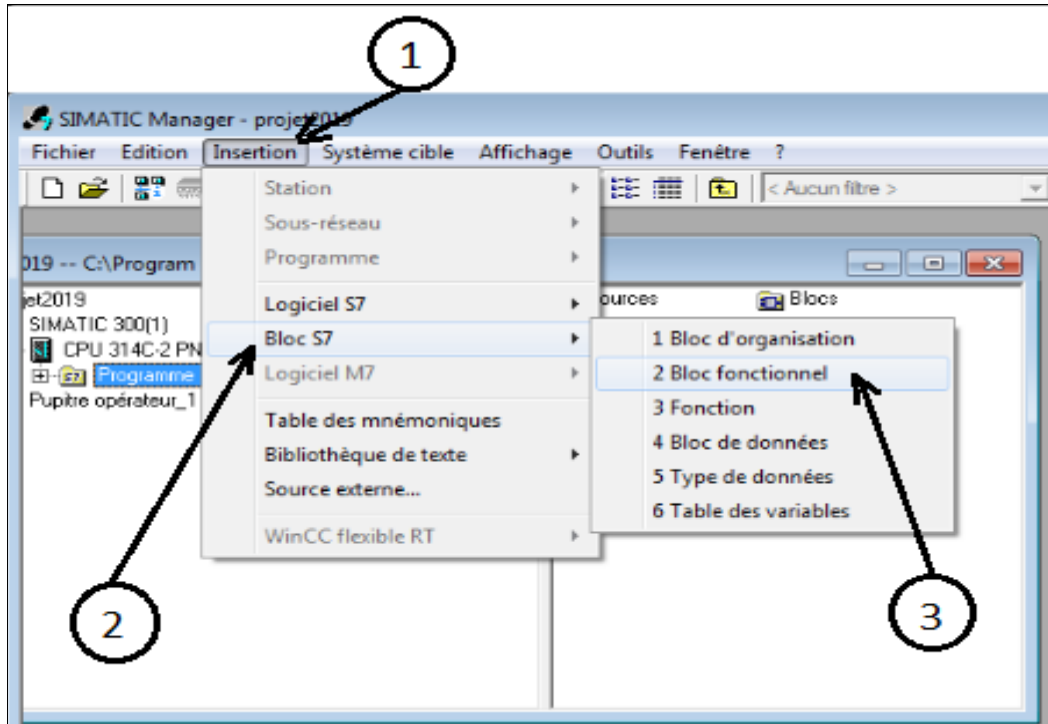


Figure III. 22 : Comment crée un bloc fonctionnel.

Les figures suivantes représentent le programme dans le bloc FC.

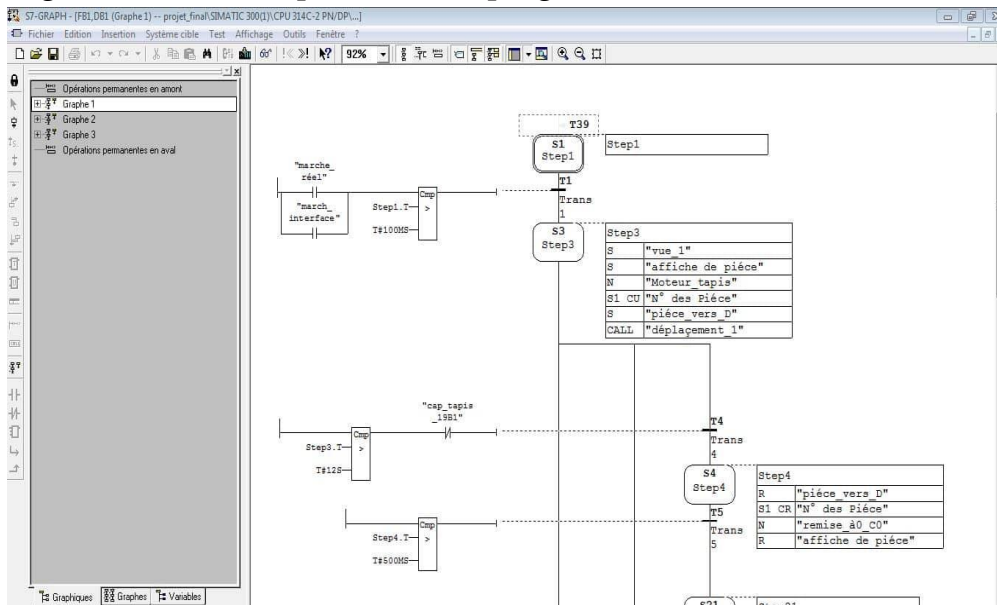


Figure III. 23 : Aperçu sur le programme de la bande transporteuse.

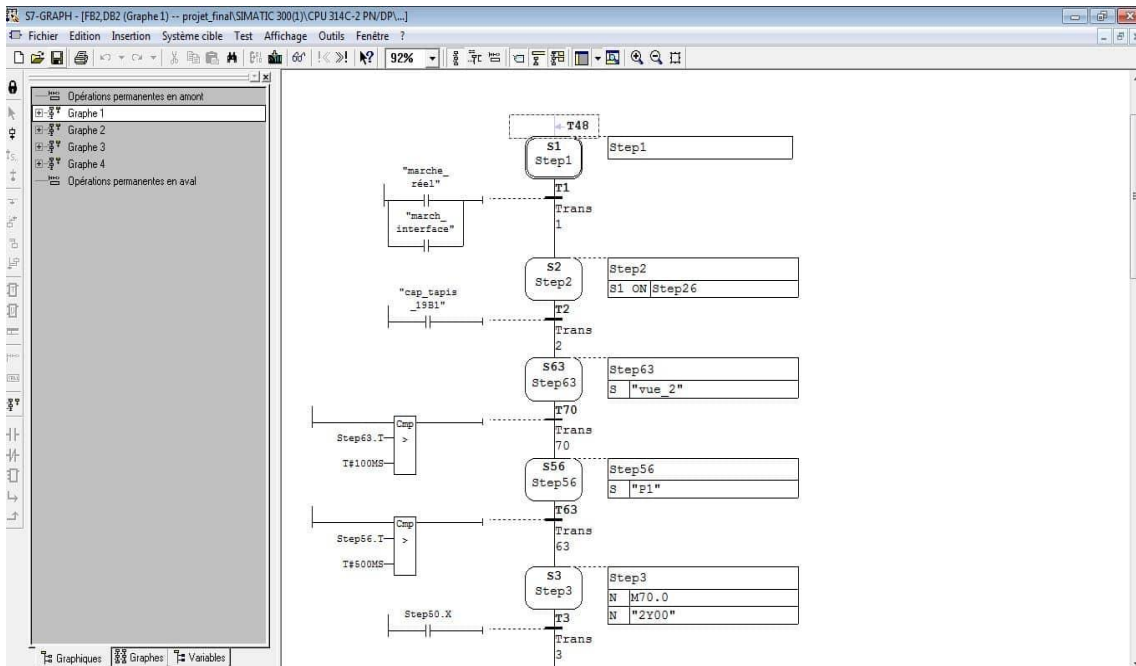


Figure III. 24 : Aperçu sur le programme du module manipulateur.

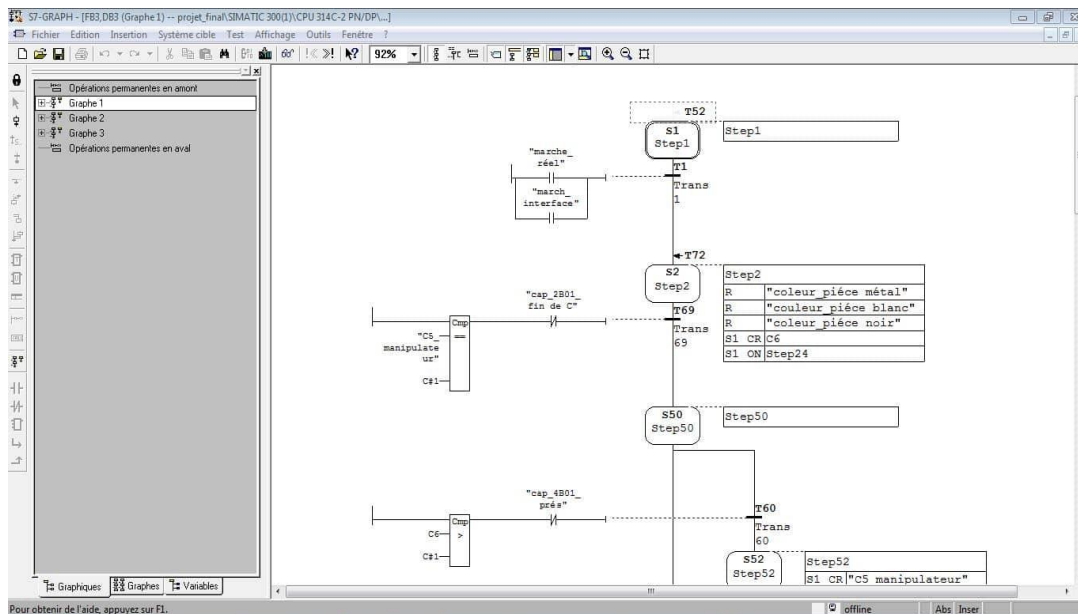


Figure III. 25 : Aperçu sur le programme du module contrôle de pièce.

III.2.6.2. Bloc d'exécution et sauvegardé des données

A : Blocs OB

Le rôle de bloc OB est de faciliter l'exécution du programme crée dans d'autres blocs. Ce processus se fait par l'appel des autres blocs de programmation dans OB (voir les Figure III.27 et III.28).

Le bloc OB peut être créé directement après configuration de matérielle, comme peut créer manuellement comme dans la Figure III.27.

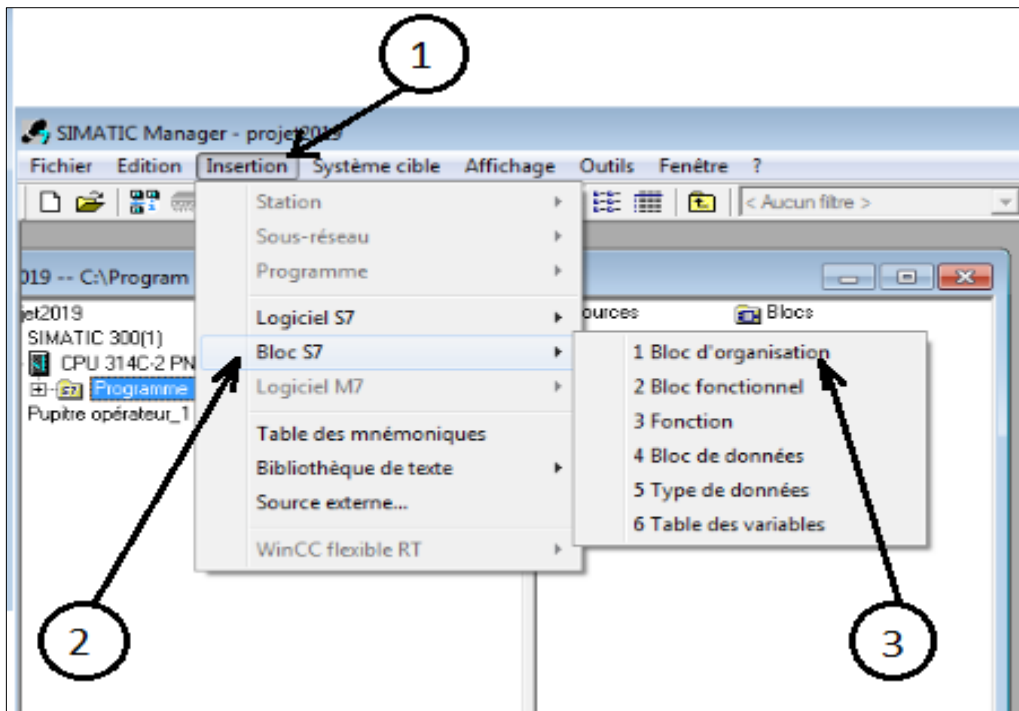


Figure III. 26 : Comment créer un bloc d'organisation manuellement.

Pour pouvoir faire la liaison entre les blocs et OB il faut procéder selon les étapes suivantes.

1. Choisir «bloc FB» et «bloc FC»,
2. Glisser «FC déplacement» dans ce réseau, par exemple,
3. Glisser «FB convoyeur» dans ce réseau, par exemple. Il est toujours nommé DB avec un numéro, exemple DB1.

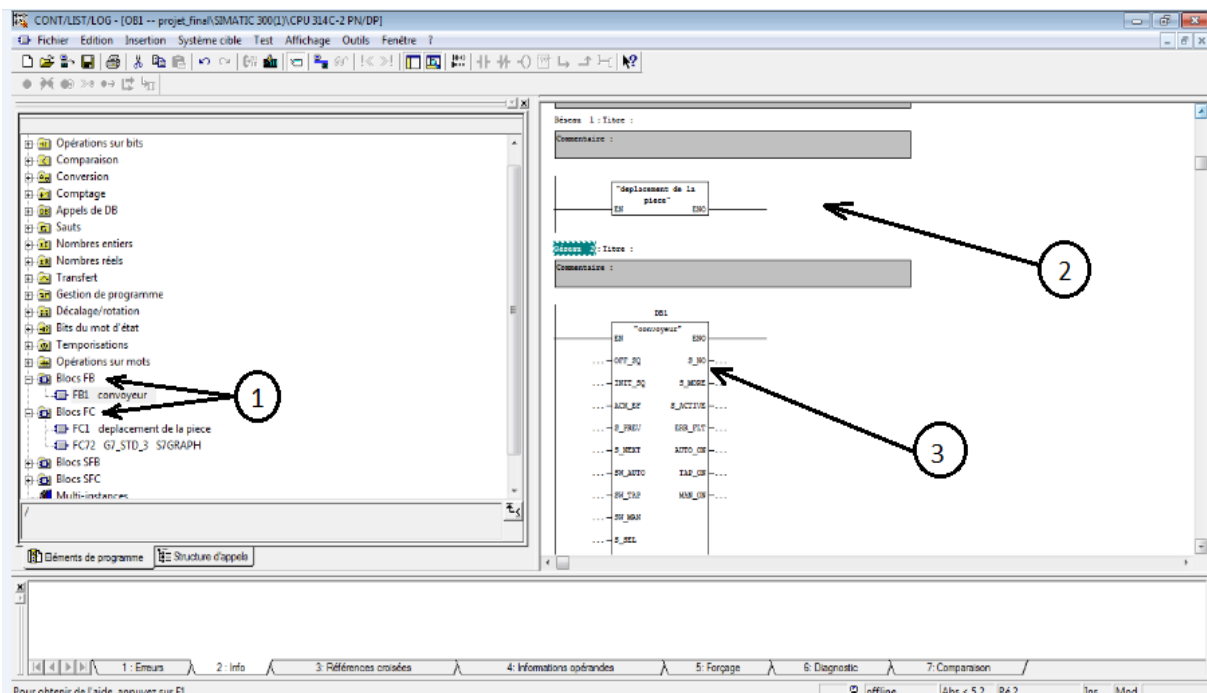


Figure III. 27 : Exemple comment ajouter les autres blocs dans bloc OB.

III.3 Programmation sur WinCC

III.3.1 Création et configuration du projet

III.3.1.1. Insertion d'une station HMI dans SIMATIC manager

Pour insérer une station HMI, il faut Suivez ces étapes.

1. Cliquer sur « projet »,
2. Cliquer sur « insertion »,
3. Cliquer sur « Station »,
4. Choisir « Station SIMATIC HMI ».

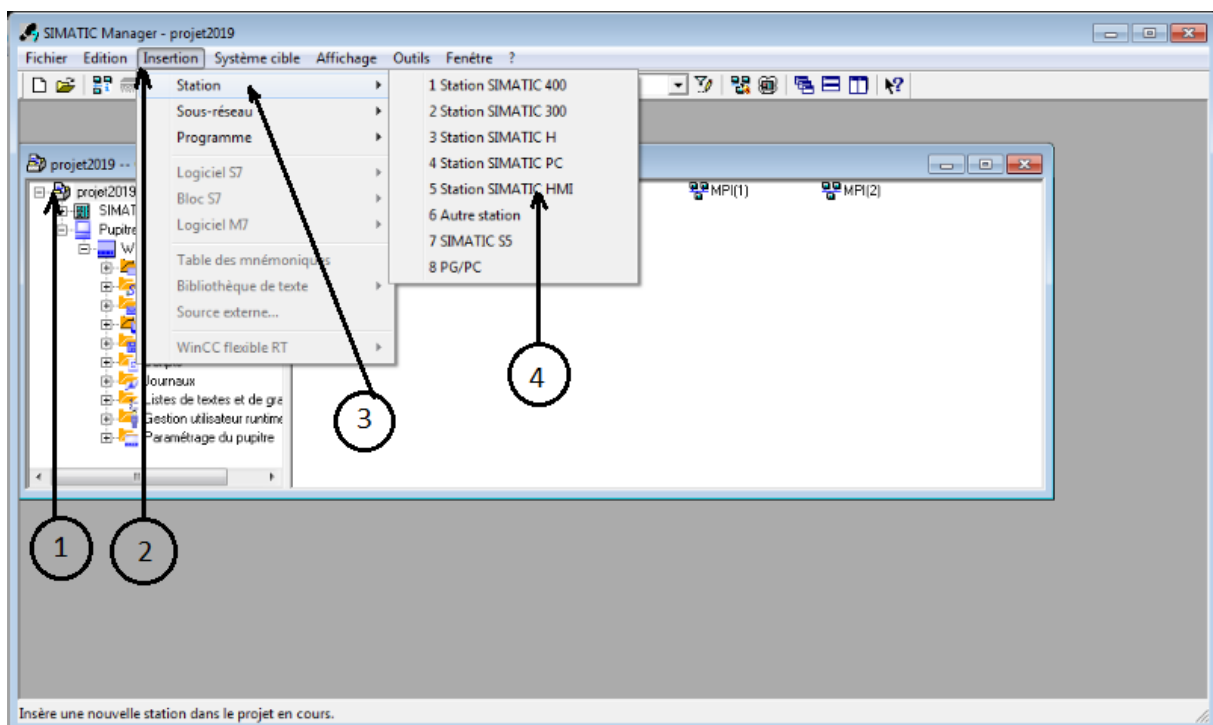


Figure III. 28 : Insertion d'un nouveau Station.

III.3.1.2 Choix de la station (pupitre)

Dans notre cas, on a utilisé WinCC flexible Runtime pour commander le système directement par le PC. Pour cela, vous devez :

1. Cliquer sur « PC »,
2. Choisir « WinCC Runtime »,
3. Cliquer sur « ok » pour confirmer votre choix.

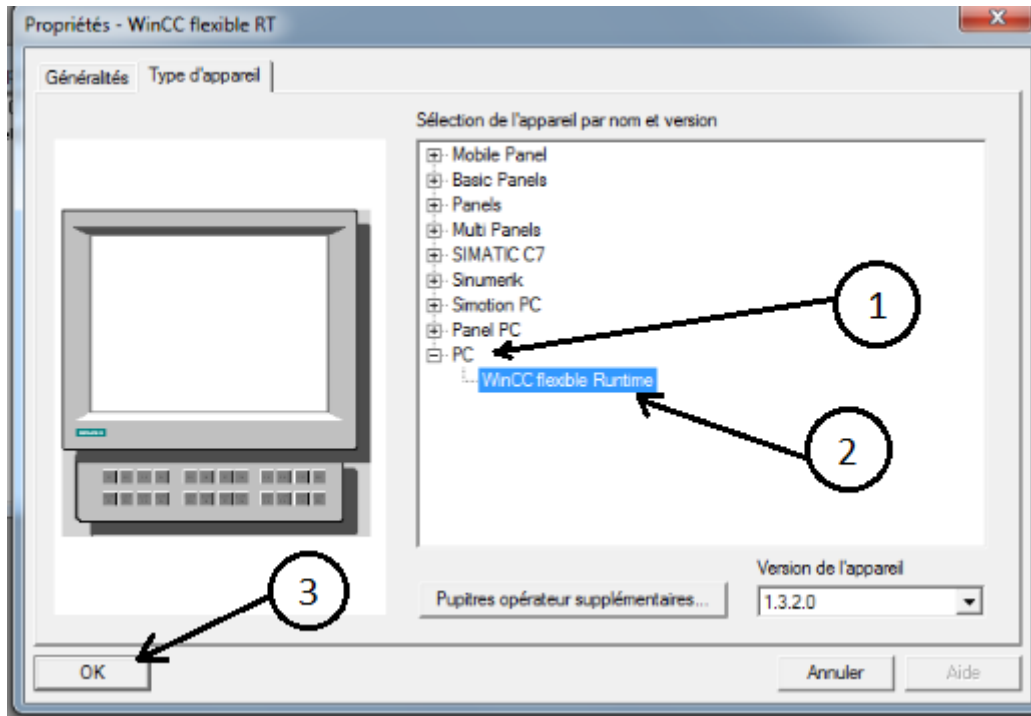


Figure III. 29 : Choix du pupitre.

III.3.1.3 Configuration du réseau

Pour la configuration des réseaux de WinCC flexible, vous devez suivre les étapes suivantes (03 étapes montrées dans la Figure III.30 et 02 dans la Figure III.31).

- 1- Cliquer sur «Pupitre opérateur 1»,
- 2- Cliquer sur «Configuration»,
- 3- Cliquer sur «NETPRO»,

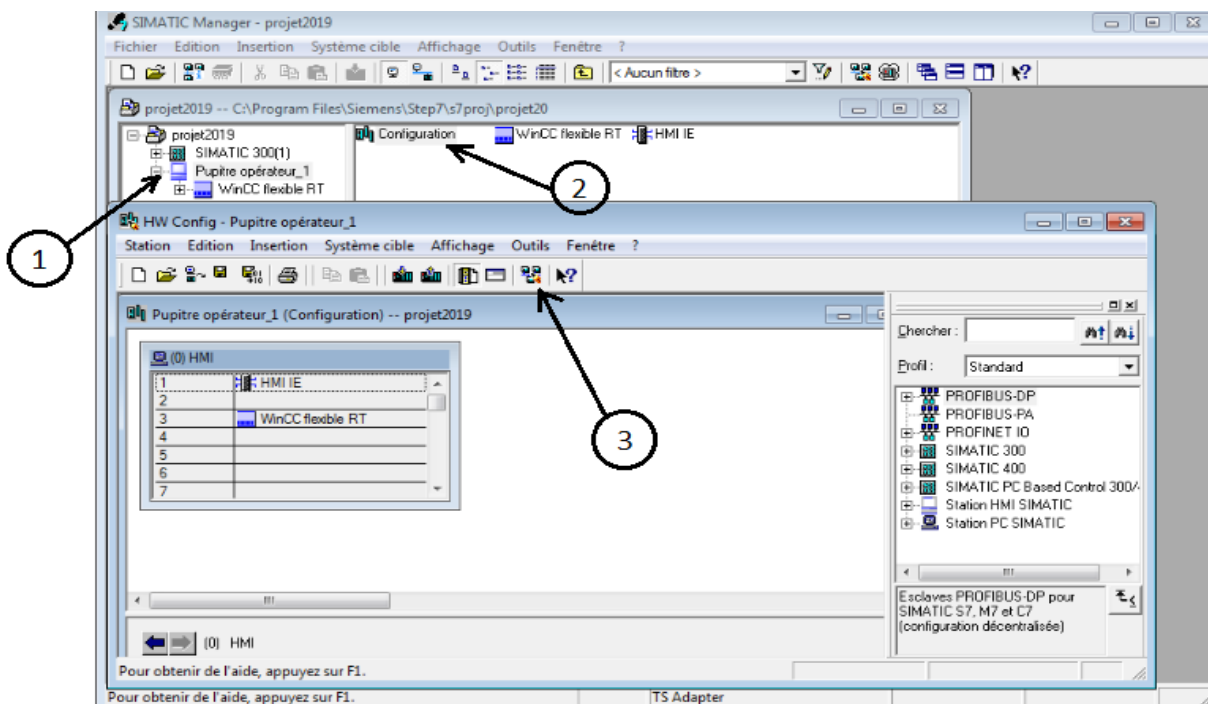


Figure III. 30 : Comment ouvrir la fenêtre de configuration du réseau.

4. Cliquer sur le carré vert, ensuite glisser vers la ligne verte,
5. Enfin enregistrer et compiler.

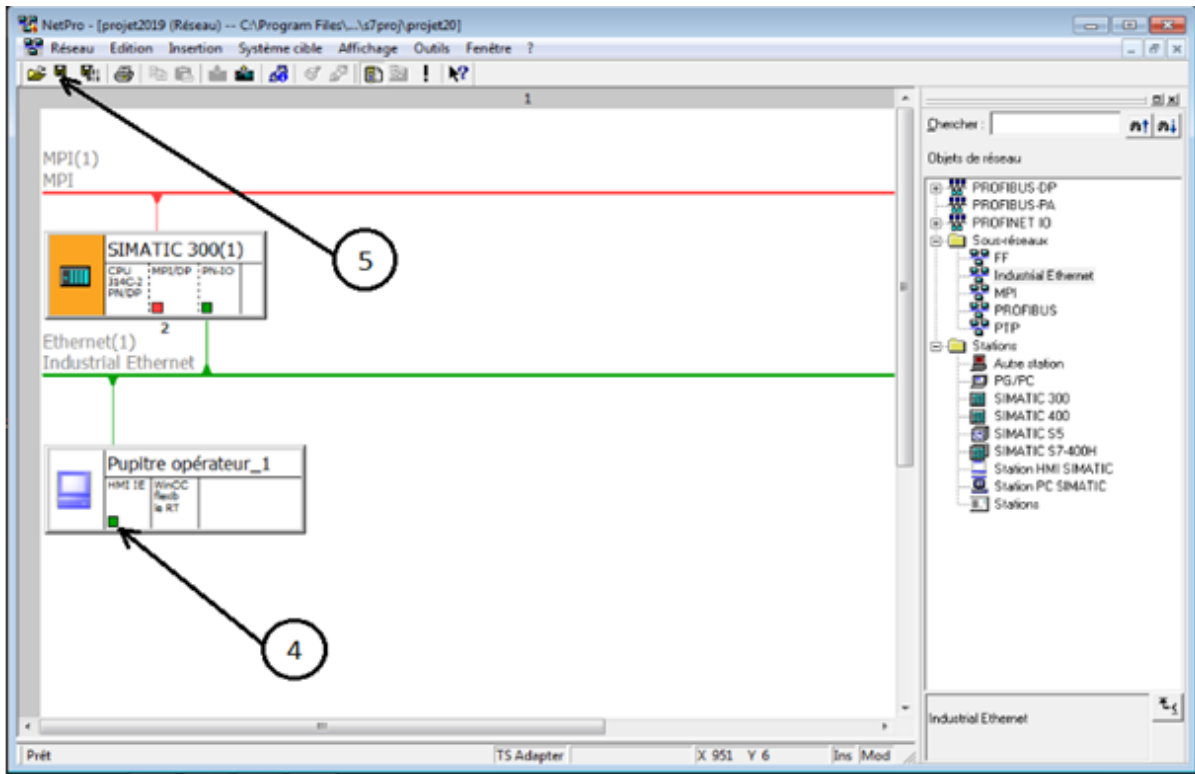


Figure III. 31 : Réseau Ethernet après configuration.

III.3.1.4 Activation de la liaison

Pour activer la liaison entre WinCC et la station de travail (API), la fenêtre de WinCC Flexible doit être ouverte. Les étapes sont montrées à partir des deux figures (Figure III.32 et III.33).

1. Cliquer sur « WinCC flexible »,
2. Cliquer sur « Vues »,
3. Cliquer sur « Modèle » ou sur « Vue_1 ».

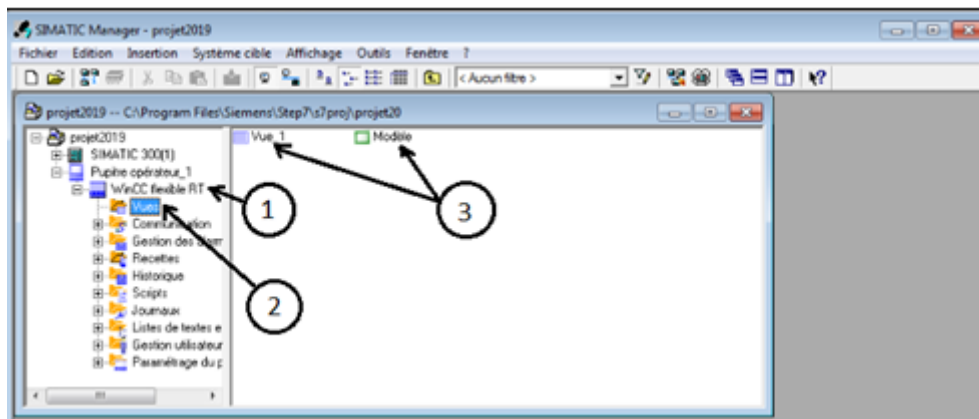


Figure III. 32 : Comment ouvrez WinCC à travers Step7.

4. Cliquer sur « communication »,
5. Cliquer sur « liaison »,
6. Double cliquer sur le carré pour ajoute « liaison_1 »,
7. Choisir interface « Ethernet ».

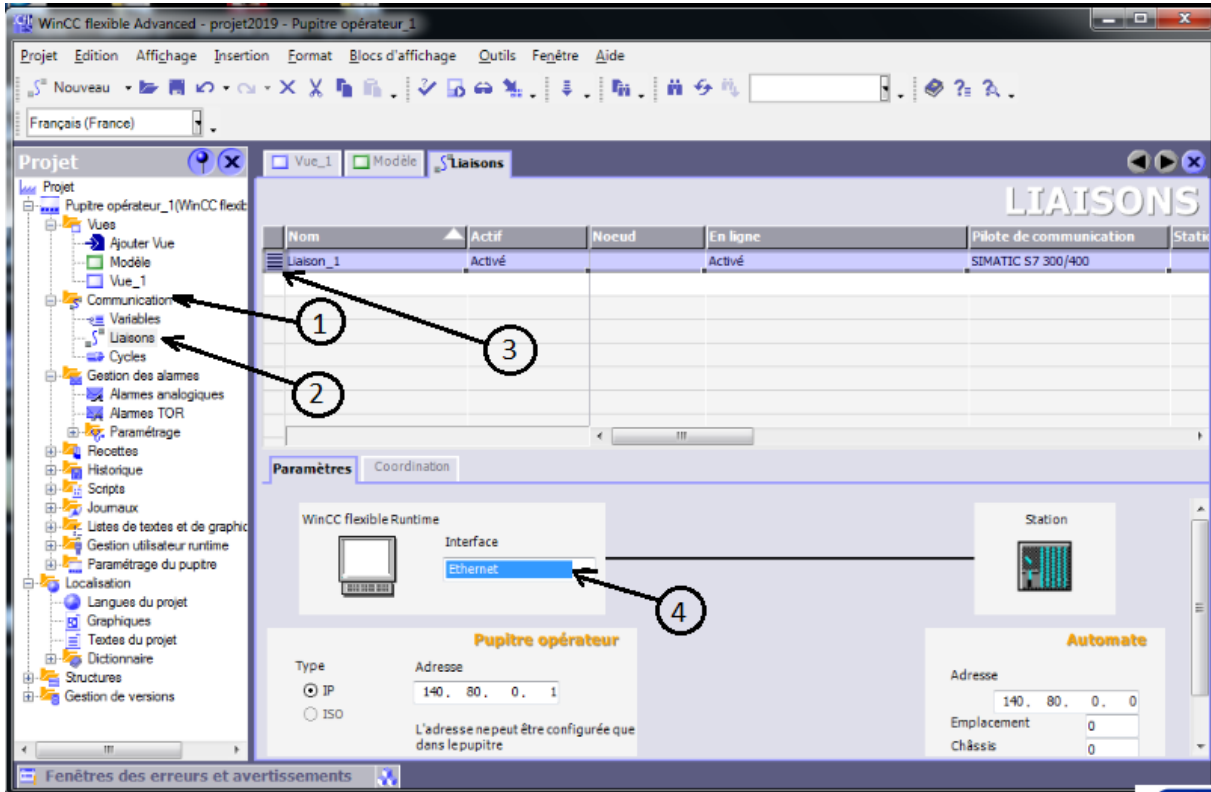


Figure III. 33 : Configuration de réseau pupitre vers l'automate.

III.3.2 Configuration des vues

Dans cette partie, on va vous donner juste à titre d'exemple une configuration sur la vue de la bande transporteuse et le bras de translation.

A : Déplacement de la pièce dans le convoyeur

La figure suivante représente la configuration pour le déplacement de la pièce de droite à gauche dans le module. Les mêmes étapes sont suivies pour le déplacement inverse. Une fois les objets seront présentés dans la vue, on procède comme suit :

1. Cliquer sur l'objet à transporter,
2. Cliquer sur « Animation »,
3. Choisir la direction de déplacement, « déplacement horizontale », puis on choisit le sens,
4. Cliquer sur « active »,
5. Choisir le « variable »,
6. Ajouter la plage de déplacement 0 jusque 95. La plage est choisie par rapport au temps de déplacement de la pièce en temps réel dans le module.

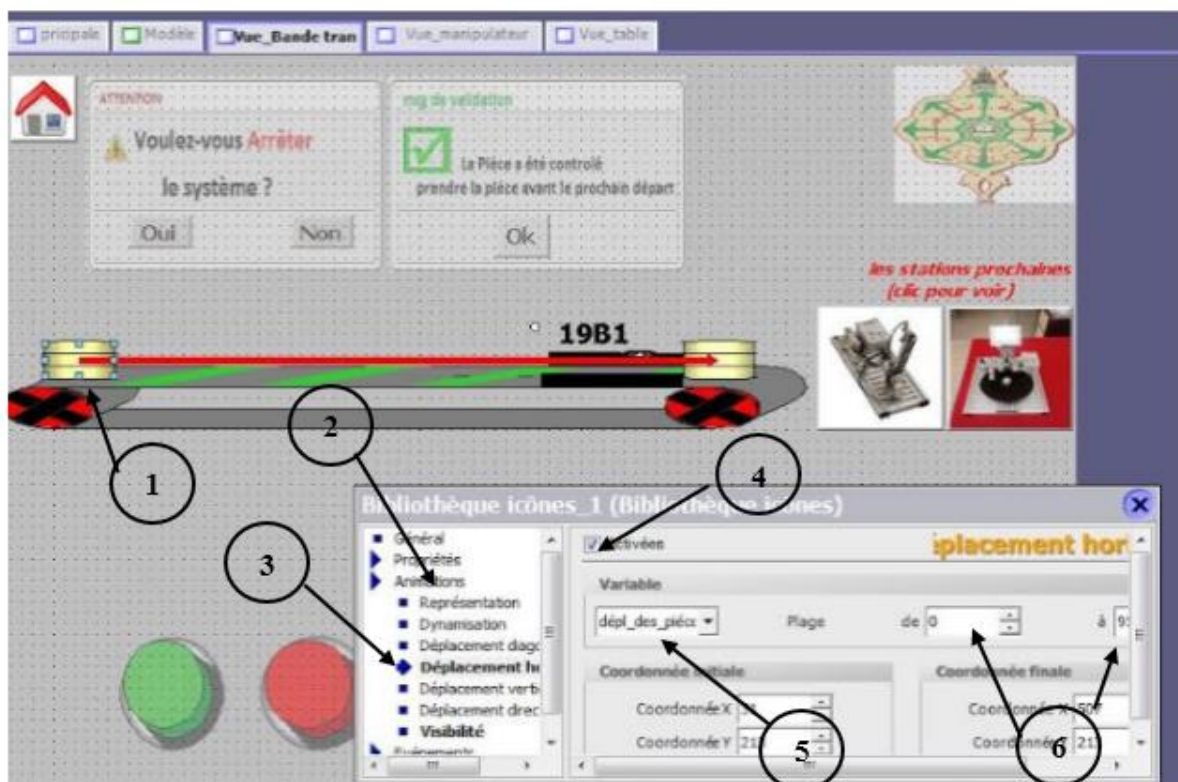


Figure III. 34 : Déplacement de la pièce droite à la gauche.

B : Activation des couleurs d'objets

Pour le changement de couleur, reportez-vous au test à effectuer par a rapport au module de contrôle des pièces. Nous avons effectué un changement de couleur directement selon ce test. Le logiciel WinCC nous donne cette modification directement en utilisant un seul objet sans utiliser la propriété de Visibilité qui fonctionne en utilisant plusieurs objets les unes au-dessus des autres.

- 1- Cliquer sur l'objet puis on clique sur animation,
- 2- Cliquer sur « représentation »,
- 3- Cliquer sur « active », puis cliquer sur la flèche pour ajouter la variable,
- 4- Choisir « binaire »,
- 5- Entrer l'intervalle de chaque pièce,
- 6- Choisir la couleur par rapport l'intervalle.
 - o Pièce métallique : couleur grise
 - o Pièce blanc : couleur blanc
 - o Pièce noire : couleur noir

Remarque importante : pour chaque couleur est assignée un bit. Pour éliminer toute conflit entre les couleurs, il faut mieux espacer entre les bits choisis. Dans notre projet, on a assigné : **bit M80.0** (pour couleur métallique, grise), **bit M80.2** (pour couleur blanche) et **bit 80.4** (pour la couleur noire). Cela va nous augmenter

l'intervalle utilisé pour le changement de couleur dans WinCC ce qui nous a facilité la tâche.

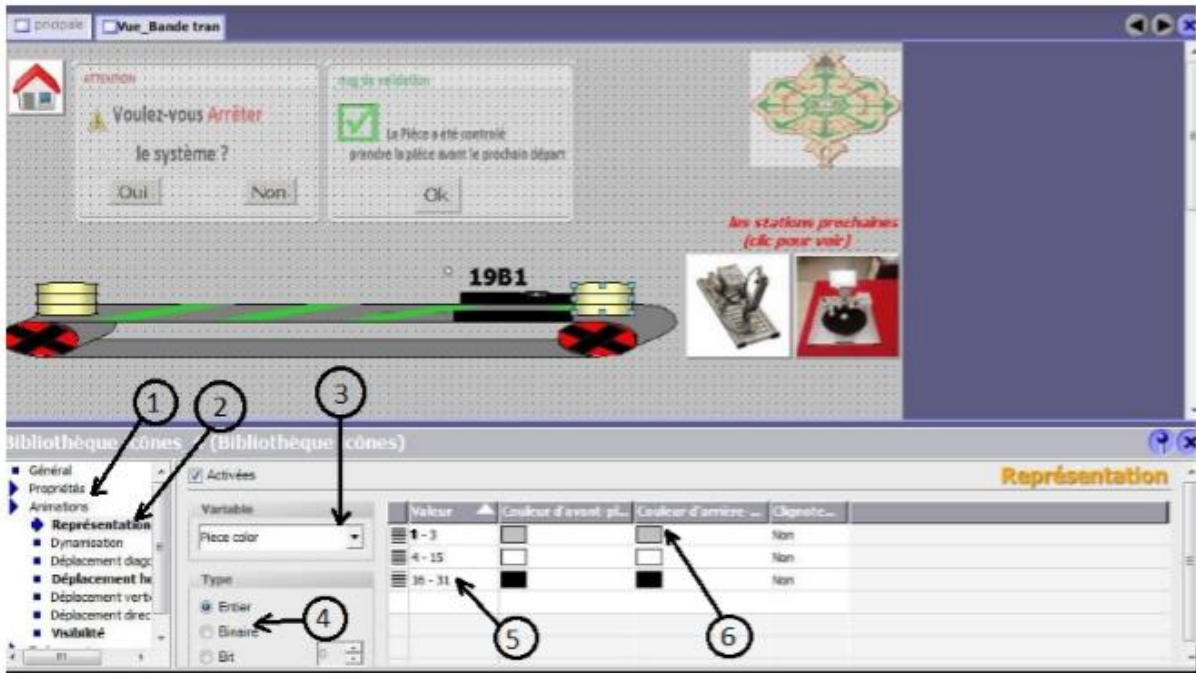


Figure III. 35 : Comment changer la couleur automatiquement dans le WinCC.

Pour expliquer le calcul d'intervalle d'une couleur, on prendra comme exemple la couleur **grise** de la pièce métallique.

Comme l'adresse **M80.0** est assigné à cette couleur, et vu que le bit **M80.1** n'est pas utilisé, donc l'intervalle va commencer par 1 (détection de la pièce métallique donc le bit $M80.0 = '1'$) jusqu'à la valeur 3 (les bits $M80.0$ et $M80.1$ sont supposés égales à '1' ; aucune influence sur la couleur).

Le même raisonnement est appliqué pour les autres couleurs.

Couleur de pièces	Métallique	Blanc	Noir
Adresse de type de pièces	M 80.0	M 80.2	M 80.4
Calcule en entière	[1-3]	[4-15]	[16-31]

Tableau 2 : Calcul d'intervalle de changement de couleur des 03 pièces.

Bits	0 bit assigné	1 bit non utilisé	2	3	4	5	6	7
Valeur min (=1)	1	0	0	0	0	0	0	0
Valeur max (=3)	1	1	0	0	0	0	0	0

Tableau 3 : Exemple de calcul d'intervalle pour la pièce métallique.

C : Visibilité

Nous avons utilisé la visibilité dans le WinCC pour afficher les objets à un moment où une variable est active et disparaît lorsque la variable s'arrête au Runtime. C'est-à-dire la modification des objets a effectué de manière dynamique. Lorsque la variable prend une valeur donnée ou lorsque sa valeur se trouve dans un intervalle donné, l'objet est affiché.

Par exemple, la figure ci-dessous explique comment utiliser cette propriété pour faire pivoter les manipulateurs : les images sont numérotées de 1 à 6. Les images apparaissent lorsque leur variable est activée, ce qui les fait apparaître comme si elles bougeaient.

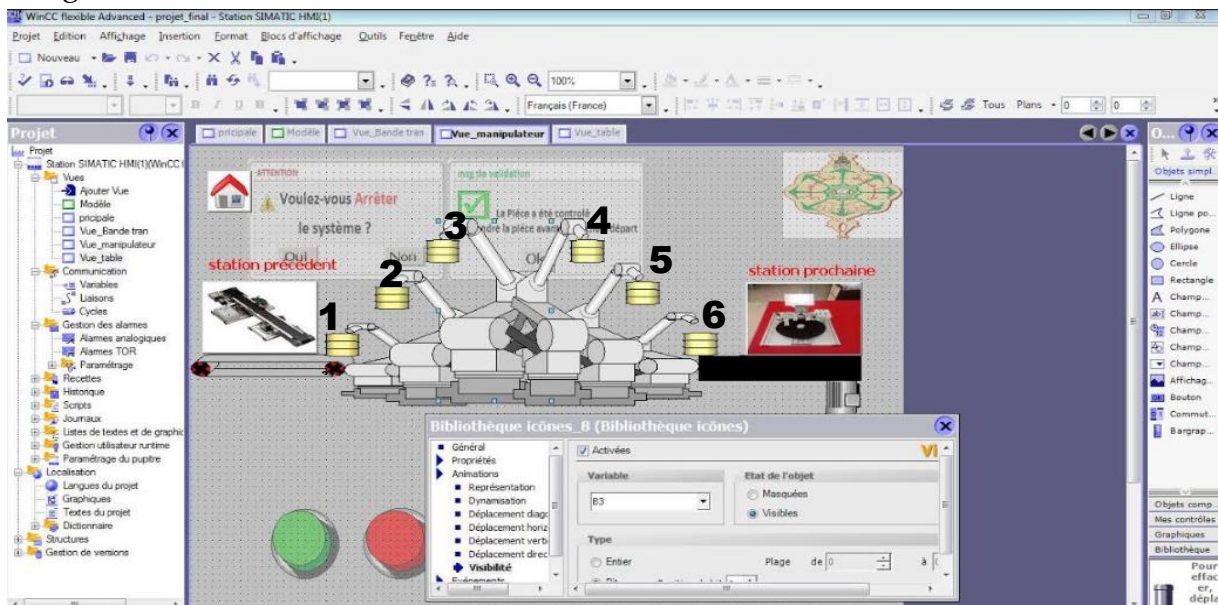


Figure III. 36 : Comment utiliser la visibilité pour montrer le mouvement du manipulateur.

La visibilité est faite selon la procédure suivante :

1. Cliquer sur « Animation »,
2. Cliquer sur « Visibilité »,
3. Cliquer sur « active »,
4. Cliquer sur flèche, puis ajouter la variable (apparition et disparition en fonction de l'état de la variable sélectionnée),
5. Choisir visibilité.

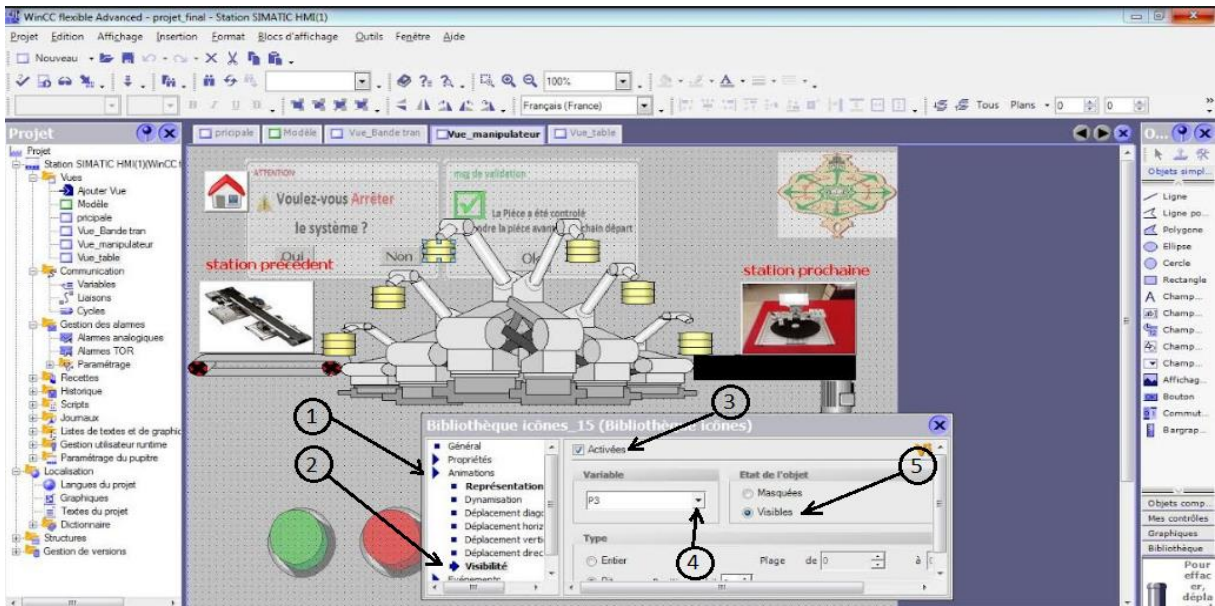


Figure III. 37 : Utilisation de la visibilité.

D : Configuration du bouton

Les étapes suivantes représentent comment configurer le bouton de l'IHM (Arrêt, Marche, Reset). Référez vous aux deux figures Figure III.39 et III.40.

- 1- Double cliquer sur le bouton,
- 2- Cliquer sur « Evènement »,
- 3- Cliquer sur « presser »,
- 4- Cliquer sur la flèche et choisit « positionner bits »,
- 5- Entrer le nom de la variable,

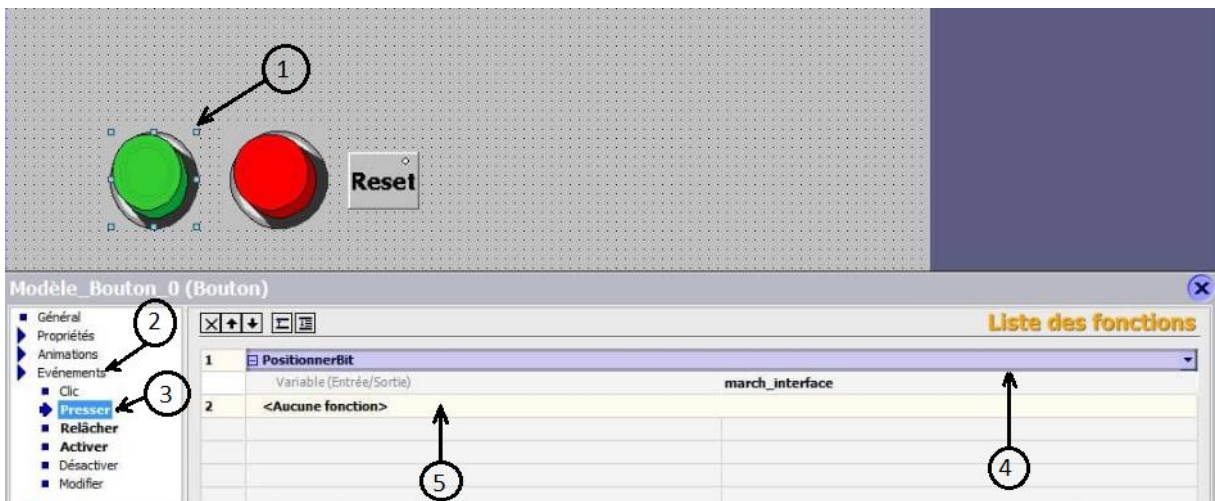


Figure III. 38 : Partie1 : configuration du bouton.

- 6- Choisir « relâcher »,
- 7- Cliquer sur cette flèche et choisir « RAZbit »
- 8- Entrer le nom de la variable.

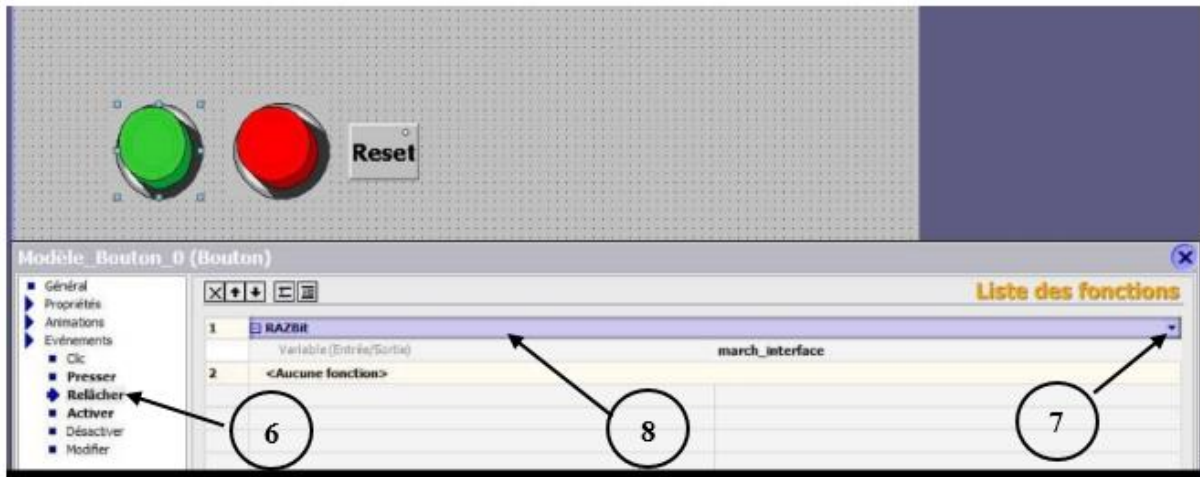


Figure III. 39 : Partie2 : configuration le bouton.

III.3.3 Représentation de l'IHM de chaque module

Dans cette partie, on va montrer les réalisations des vues de nos IHM.

III.3.3.1 IHM Vue de module bande transporteuse

La figure ce dessous représente l'IHM de module bande transporteuse.

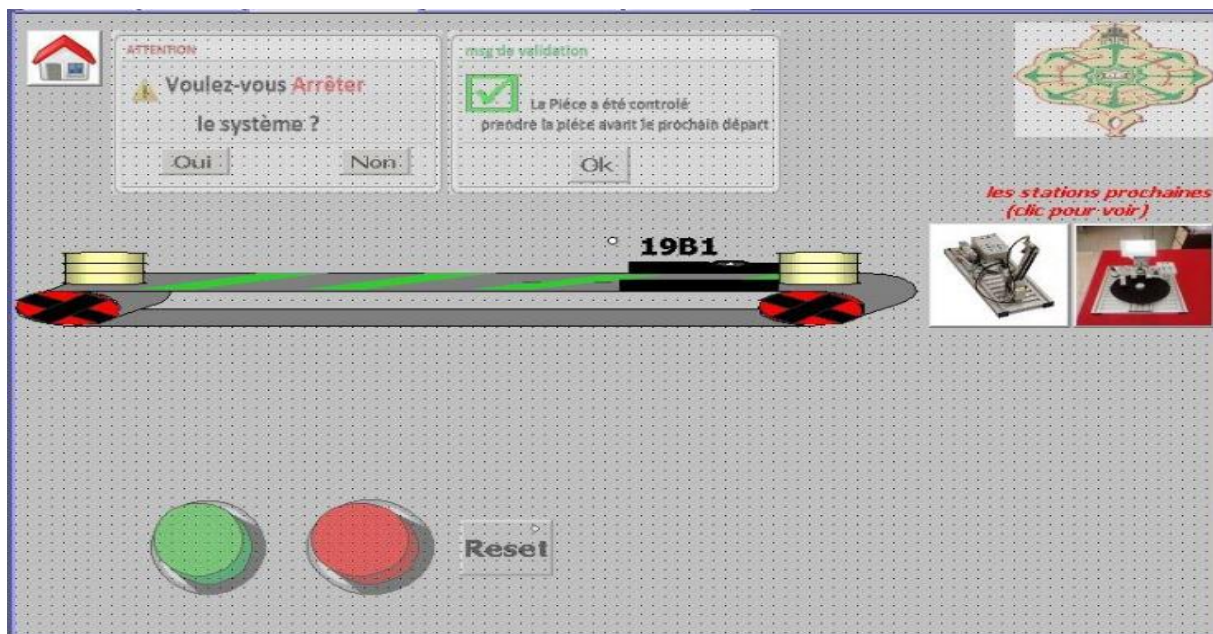


Figure III. 40 : IHM du module bande transporteuse.

III.3.3.2 IHM de module manipulateur

La figure ci-dessous représente l'IHM de module manipulateur.

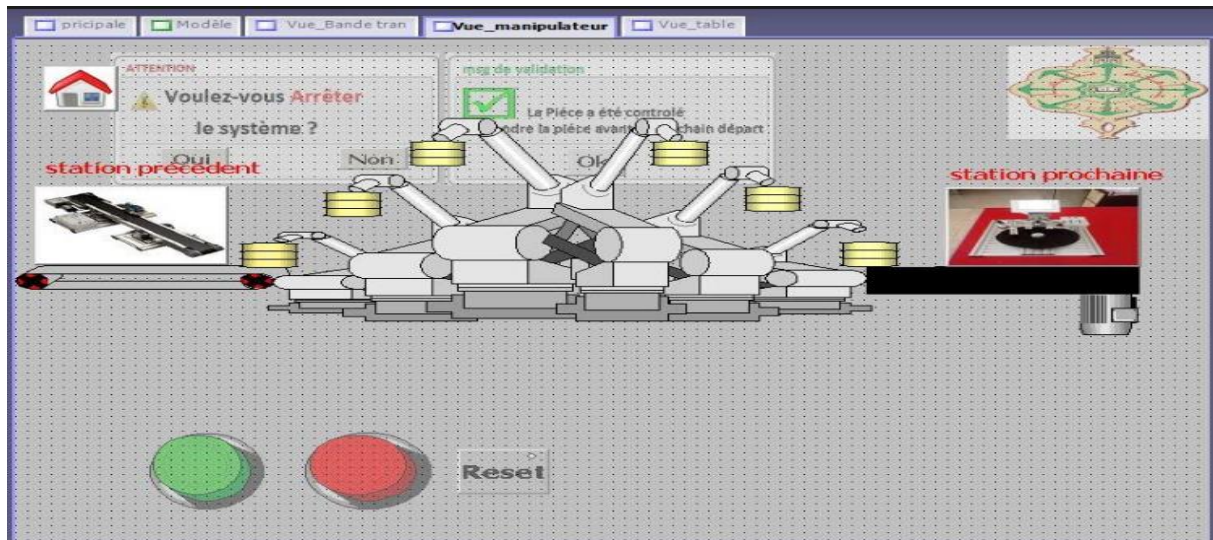


Figure III. 41: IHM du module manipulateur.

III.3.3.3 IHM du module contrôle de pièce

La figure ce dessous représente l'IHM de module contrôle de pièce.

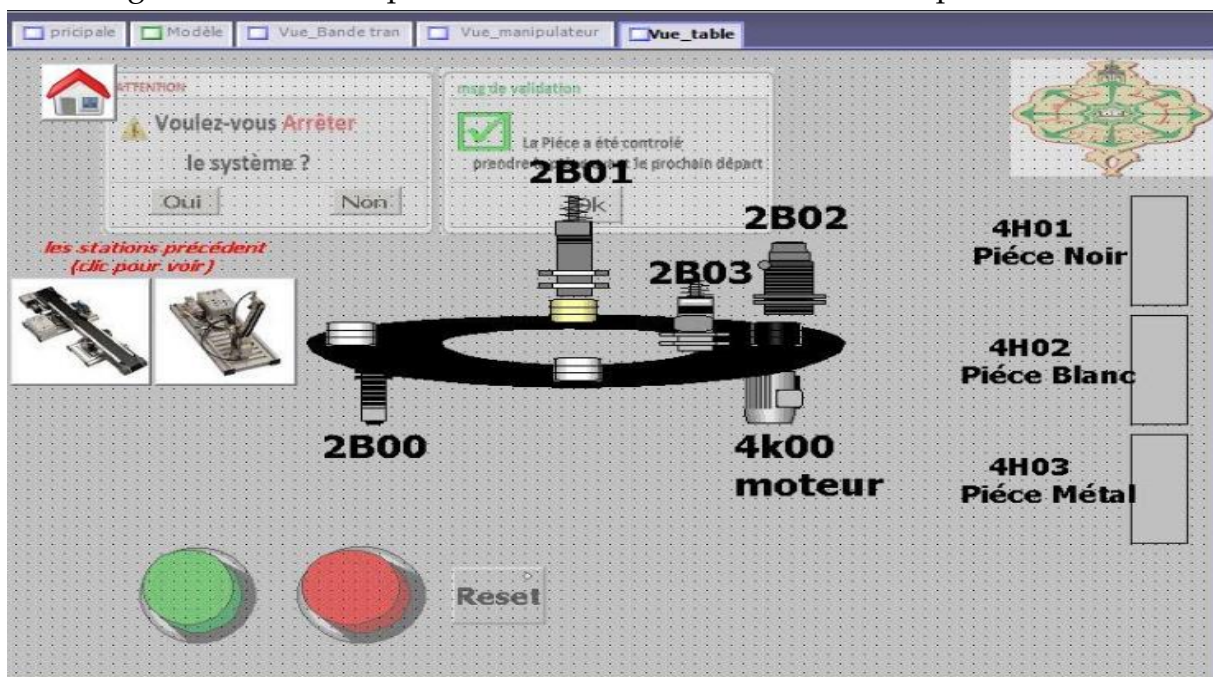


Figure III. 42 : IHM du module contrôle de pièce.

III.3.3.4 IHM principale

La figure ci-dessous représente l'IHM principale. Son rôle est de gérer la visibilité des différentes vues en cliquant sur la figure qu'on veut visualiser. (c'est-à-dire une vue Modèle qui nous aidera pour basculer d'une vue de module à une autre.)

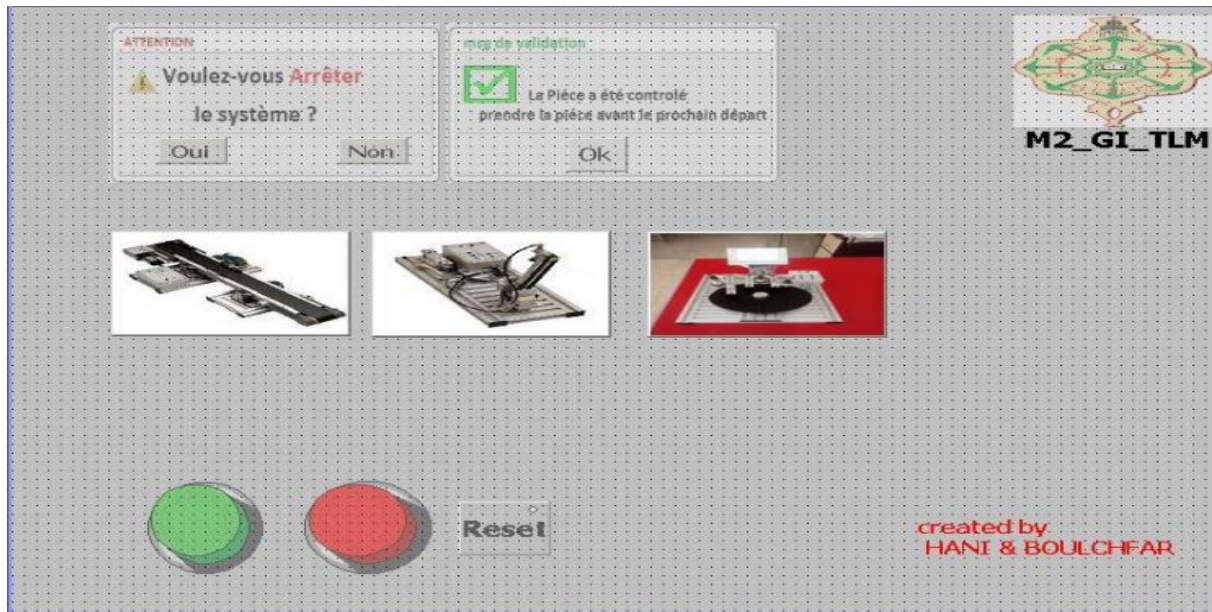


Figure III. 43 : IHM principale.

III.3.3.5 Message du système

La figure de ce dessous représente les messages envoyés par le système

Message 1 : pour valider l'arrêt du système,

Message2 : devient visible dans le cas où le cycle de travail est achevé.

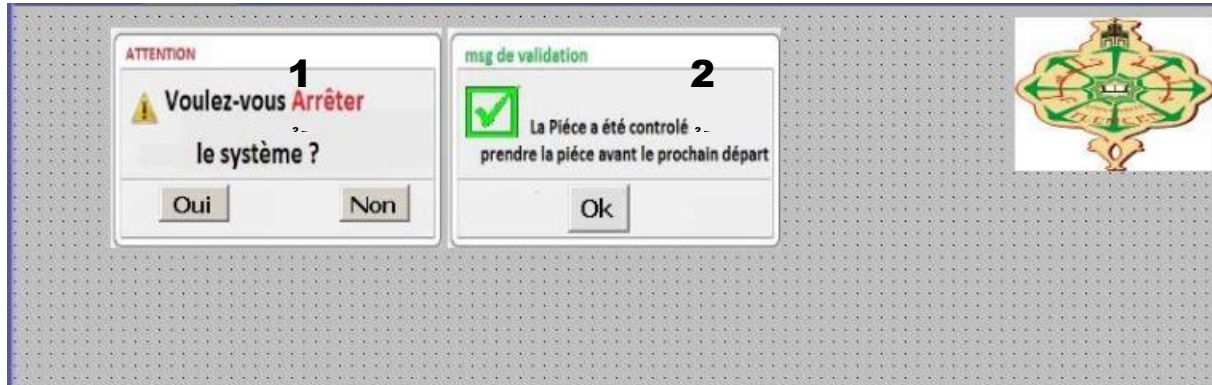


Figure III. 44 : Message envoyé par le système.

III.3.4 Importation des donner du système

Le logiciel WinCC nous permet de transférer et d'importer des informations de système à l'utilisateur. Ce qui nous aide à suivre l'état de fonctionnement du système en format Excel.

Pour importer des textes de WinCC flexible vers fichier format Excel, veuillez procéder de la façon suivante :

1. Cliquer sur « projet »,
2. Choisir « importation/Exportation »,
3. Choisir « importation CSV »,

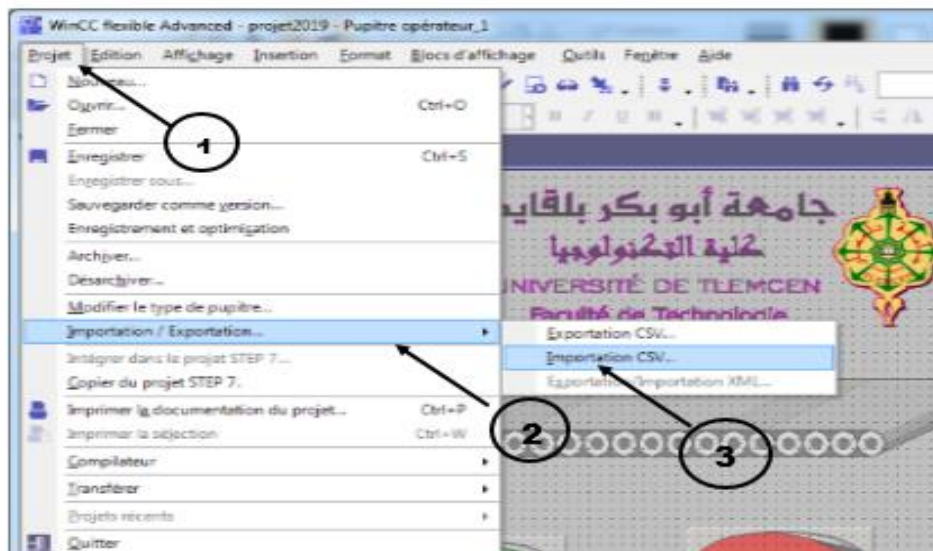


Figure III. 45 : Ouverture de la fenêtre d'importation.

4. Sélectionner les textes devant être importés de WinCC,
5. Cliquer sur « importation ».

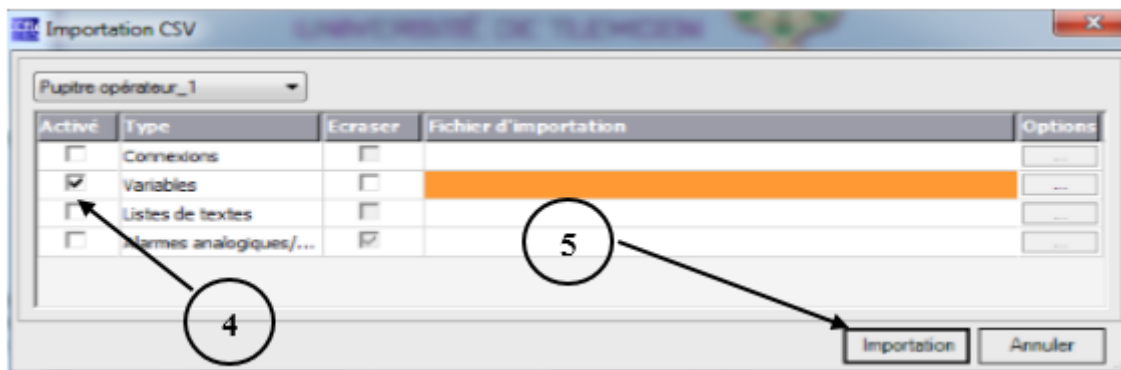


Figure III. 46 : Fenêtre de choix des données.

Variable	Type	Valeur	Description
#Colonnes A	nom de variable (obligatoire - sans apostrophe)		
#Colonnes B	connexion (si vide, interne)		
#Colonnes C	adresse (dépend de la connexion)		
#Colonnes D	type de données (dépend de la connexion)		
#Colonnes E	longueur (nécessaire si le type de données est Sting)		
#Colonnes F	éléments du tableau		
#Colonnes G	mode d'acquisition (1 = Sur événement 2 = Cyclique sur utilisation (par défaut) 3 = cyclique en continu)		
#Colonnes H	cycle d'acquisition, par ex. 1s		
#Colonnes I	limite supérieure (nombre à virgule flottante)		
#Colonnes J	limite supérieure supplémentaire (nombre à virgule flottante)		
#Colonnes K	limite inférieure supplémentaire (nombre à virgule flottante)		
#Colonnes L	limite inférieure (nombre à virgule flottante)		
#Colonnes M	Échelonnement linéaire (0 ou false, 1 ou true)		
#Colonnes N	valeur d'échelle supérieure AS (nombre à virgule flottante)		
#Colonnes O	valeur d'échelle inférieure AS (nombre à virgule flottante)		
#Colonnes P	valeur d'échelle supérieure IHM (nombre à virgule flottante)		
#Colonnes Q	valeur d'échelle inférieure IHM (nombre à virgule flottante)		
#Colonnes R	valeur initiale (le type dépend du type de données)		
#Colonnes S	ID de mise à jour		
#Colonnes T	commentaire (500 caractères max.)		

Variable	M5.2	M5.1	M5.0	Q125.2	M0.1	M3.0	M3.1	M3.2	M3.3	M3.4	M3.5	Q124.5	Description
affichage d Liason_1	Bool	1	2 100 ms										0 affichage de pièce en retour dans wincc
affichage d Liason_1	M5.1	Bool	1	2 100 ms									0 variable pour l'affichage de table
affiche de Liason_1	M5.0	Bool	1	2 100 ms									0 affichage de pièce dans wincc dans tapis
arrêt_2	Liason_1	Q125.2	Bool	1	2 100 ms								0 pour utilisation dans wincc
arrêt_interf	Liason_1	M0.1	Bool	1	2 100 ms								0 bouton arrêt dans Wincc
B1	Liason_1	M3.0	Bool	1	2 100 ms								0 position de BRAS sans pièce dans IHM
B2	Liason_1	M3.1	Bool	1	2 100 ms								0 position de BRAS sans pièce dans IHM
B3	Liason_1	M3.2	Bool	1	2 100 ms								0 position de BRAS sans pièce dans IHM
B4	Liason_1	M3.3	Bool	1	2 100 ms								0 position de BRAS sans pièce dans IHM
B5	Liason_1	M3.4	Bool	1	2 100 ms								0 position de BRAS sans pièce dans IHM
B6	Liason_1	M3.5	Bool	1	2 100 ms								0 position de BRAS sans pièce dans IHM
blanc_4H0	Liason_1	Q124.5	Bool	1	2 100 ms								0 commande DEL pièce de couleur blanc

Figure III. 47 : Importation du texte de la variable.

III.4 Compilation et simulation du programme

III.4.1 Simulation sous Step7

III.4.1.1 Activation de la simulation

La figure suivante représente comment activer la simulation.

1- Sélectionner le « Blocs »,



2- Activé la simulateur

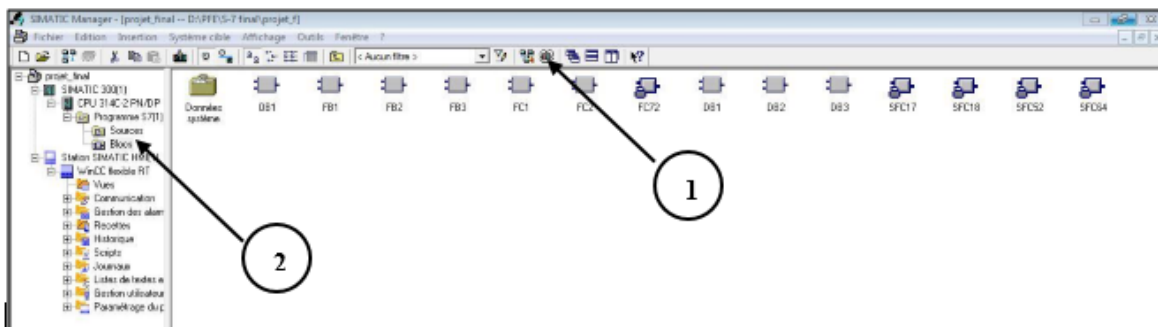



Figure III. 48 : Activation de la simulation.

III.4.1.2 Chargement du programme

Le chargement du programme se fait en appuyant sur . Cette action permet de charger le programme dans la CPU pour l'exécution.

1- Sélectionner le « Blocs »,

2- Charger le programme.

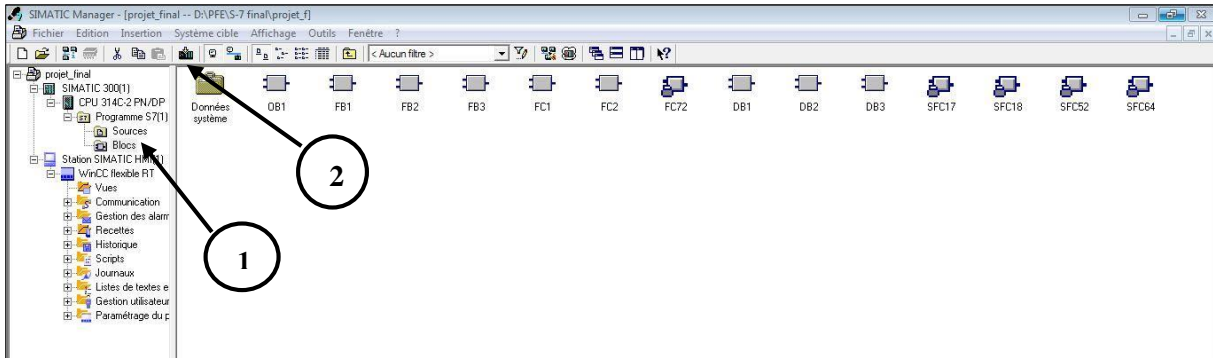


Figure III. 49 : Chargement du programme.

III.4.1.3 S7-PLCSIM

S7-PLCSIM c'est un simulateur de Step7 qui représente l'automate virtuel et nous permet de détecter des erreurs logiques de configuration, d'exécution et de test de notre programme qu'on va simuler sur l'ordinateur.

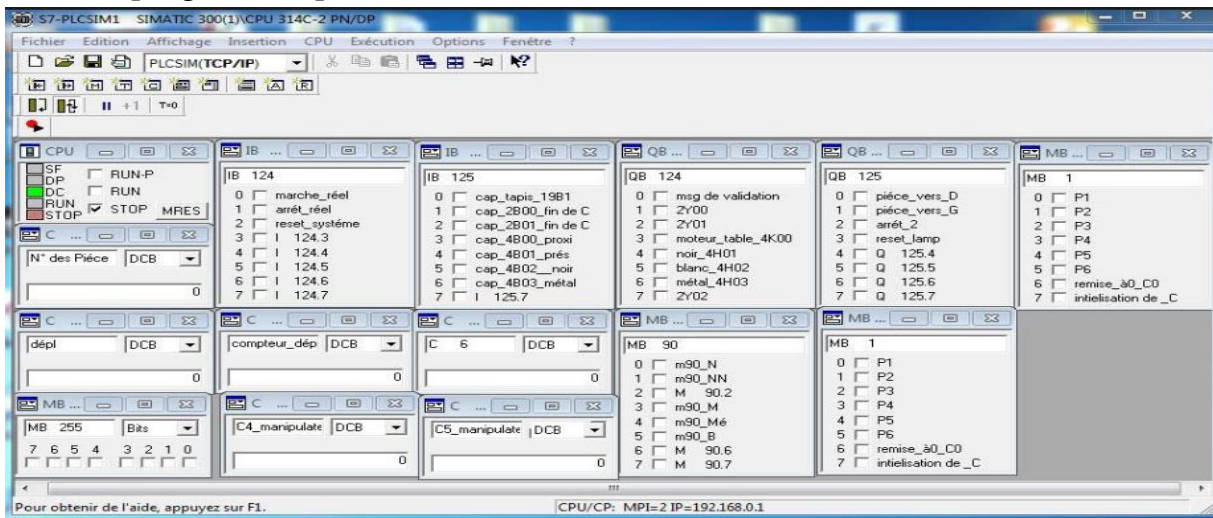


Figure III. 50 : Vue générale du logiciel S7-PLCSIM.

III.4.1.4 Lancement de la simulation

Cela se produit en appuyant sur « RUN-P », comme le montre la figure ci-dessous.

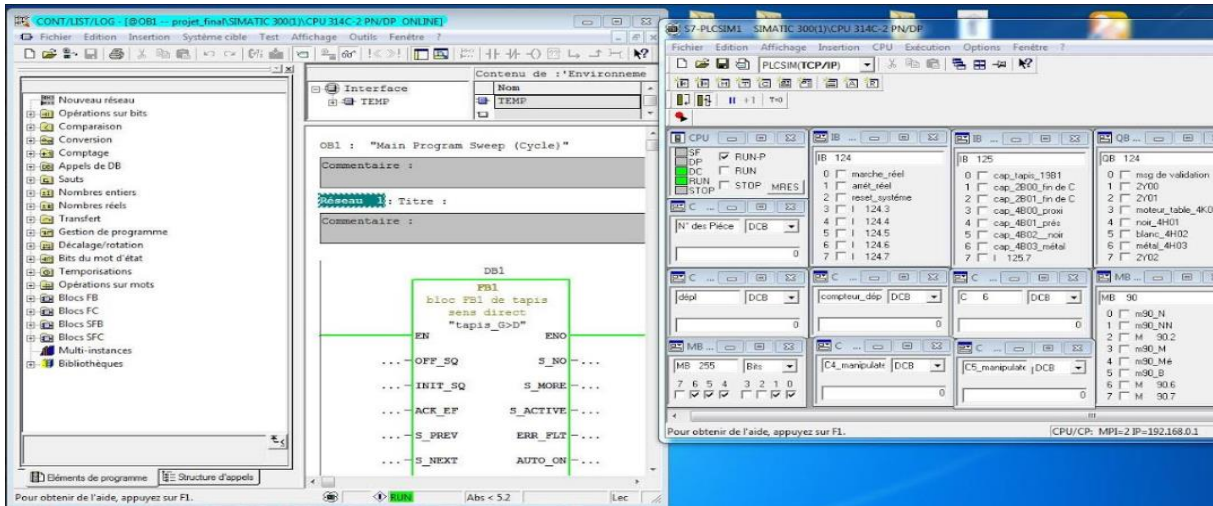


Figure III. 51 : Démarrage de la simulation.

III.4.2 Simulation avec WinCC

Une fois que la liaison est activée, tout ce qui nous reste à faire c’est de visualiser le processus sur le pupitre.

La simulation se fait avec l’activation de la commande « Runtime » qui se trouve dans la barre de menu sur la fenêtre WinCC flexible.

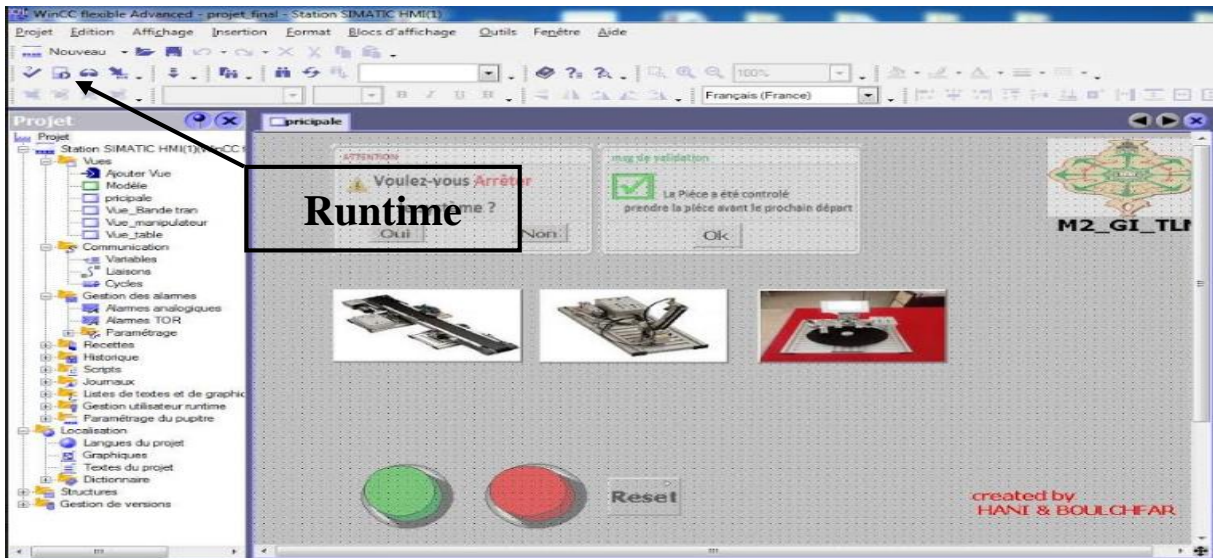


Figure III. 52 : Activation de la visualisation avec Runtime.

La fenêtre ci-dessous apparaît, juste après qu’on appuie sur l’icône « Runtime », donc on aura une page d’accueil qui représente l’ensemble des modules à traiter.

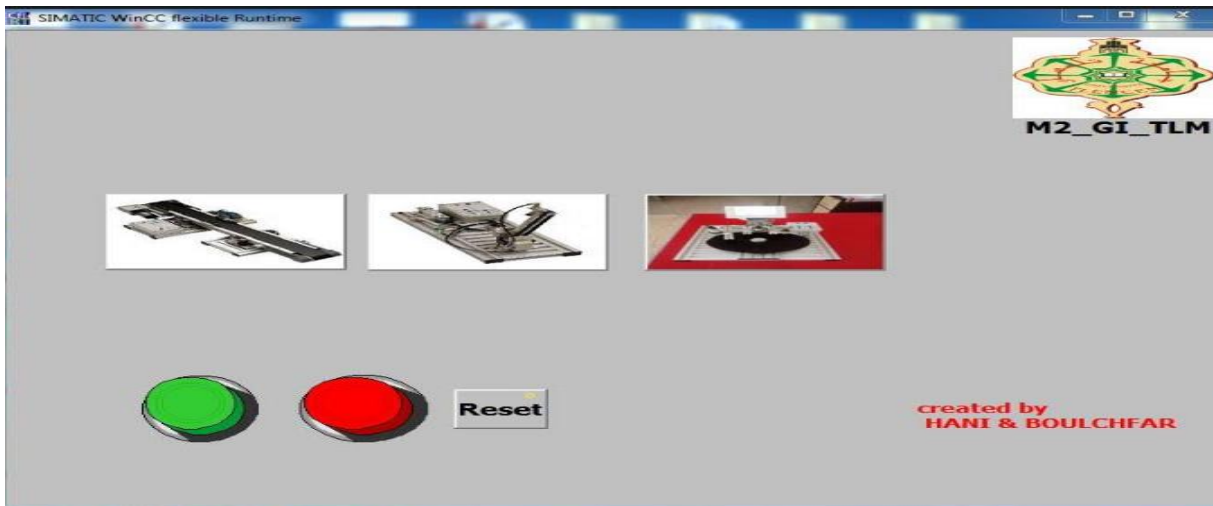


Figure III. 53 : Page d'accueil du système sur le pupitre.

III.4.3 Simulation de programme PLCSIM avec WinCC

Le PLCsim est un simulateur du logiciel Step7 de Simatic manger qui nous permet de simuler les programmes, il suffit qu'une liaison pc/automate soit établie. Notre interface est simple ; elle nous permettra de visualiser et de forcer les différents paramètres présents dans le programme.

Après l'intégration des deux projets WinCC et step7 entre eux, ainsi après l'activation des deux simulateurs PLCsim et Runtime, on peut observer une exécution du programme sur les deux logiciel **step7** et **WinCC** en parallèle.

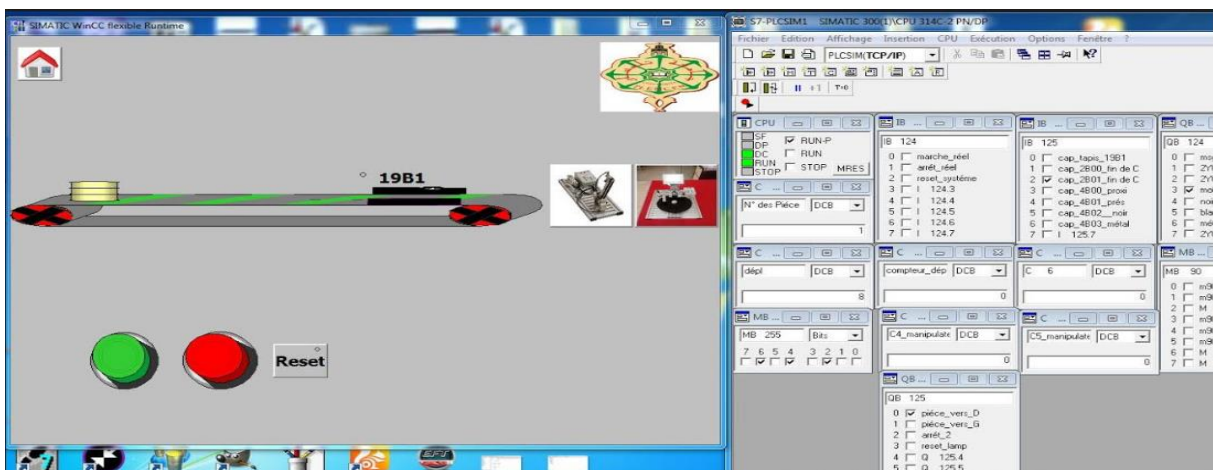


Figure III. 54 : Exécution de l'IHM via le simulateur PLCsim.

III.5 Injection du projet dans API réel S7-300

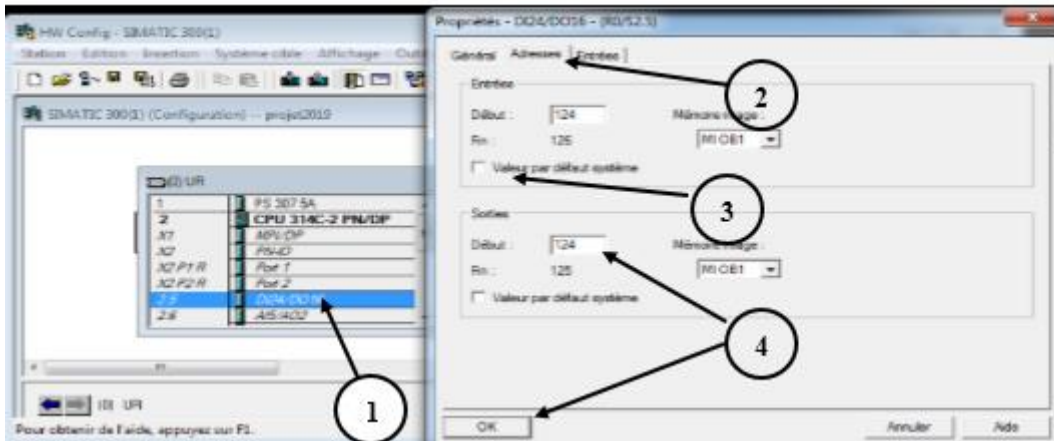
III.5.1 Configuration des Adresse selon le Hardware de notre API S7-300

Pour relier le système étudié et l'API, nous avons procédé comme suit :

- Nous avons configuré les adresses d'entrées/sorties de chaque module dans le programme, par rapport aux adresses d'API sur terrain. L'API accepte les

adresse 124.0 jusqu'à 125.7. Donc Nous avons changé toutes les adresses des entrées/sorties du programme à partir de 124, comme dans la figure suivante :

1. Double cliquer sur « DI24/DO16 »,
2. Cliquer sur « Adresses »,
3. Débaucher la fenêtre « valeur par défaut système »,
4. Entrer début de l'adresse et enfin cliquer sur « ok »,



Figure

III. 55 : Configuration des adresses DI/DO de chaque module.

- En fin, relier électriquement les DI/DO de chaque module avec le Box d'interface, avec l'adresse adéquate dans l'API. en suivant la table mnémorique.

III.5.2 Commande du système via IHM pupitre

III.5.2.1 Configuration réseau utilisé

A : Réseau MPI

La figure suivante représente la commande du système par le pupitre, nous avons utilisé un adaptateur USB/MPI. Pour assurer la connexion entre la CPU et PC, la Figure 3.56 ci-dessous montre liaison entre Pc/automate.



Figure III. 56 : Connexion PC avec l'automate Via MPI adaptateur.

Pour assurer la connexion entre PC et la CPU par adaptateur, vous devez suivre les étapes suivantes :

- Installer le driver après redémarrer ordinateur,
- Ouvrir SIMATIC manager Et suivez les étapes comme indiqué dans la figure suivante :

A. Cliquer sur « outils »,

B. Cliquez sur « paramétrage de l'interface PG/PC ».

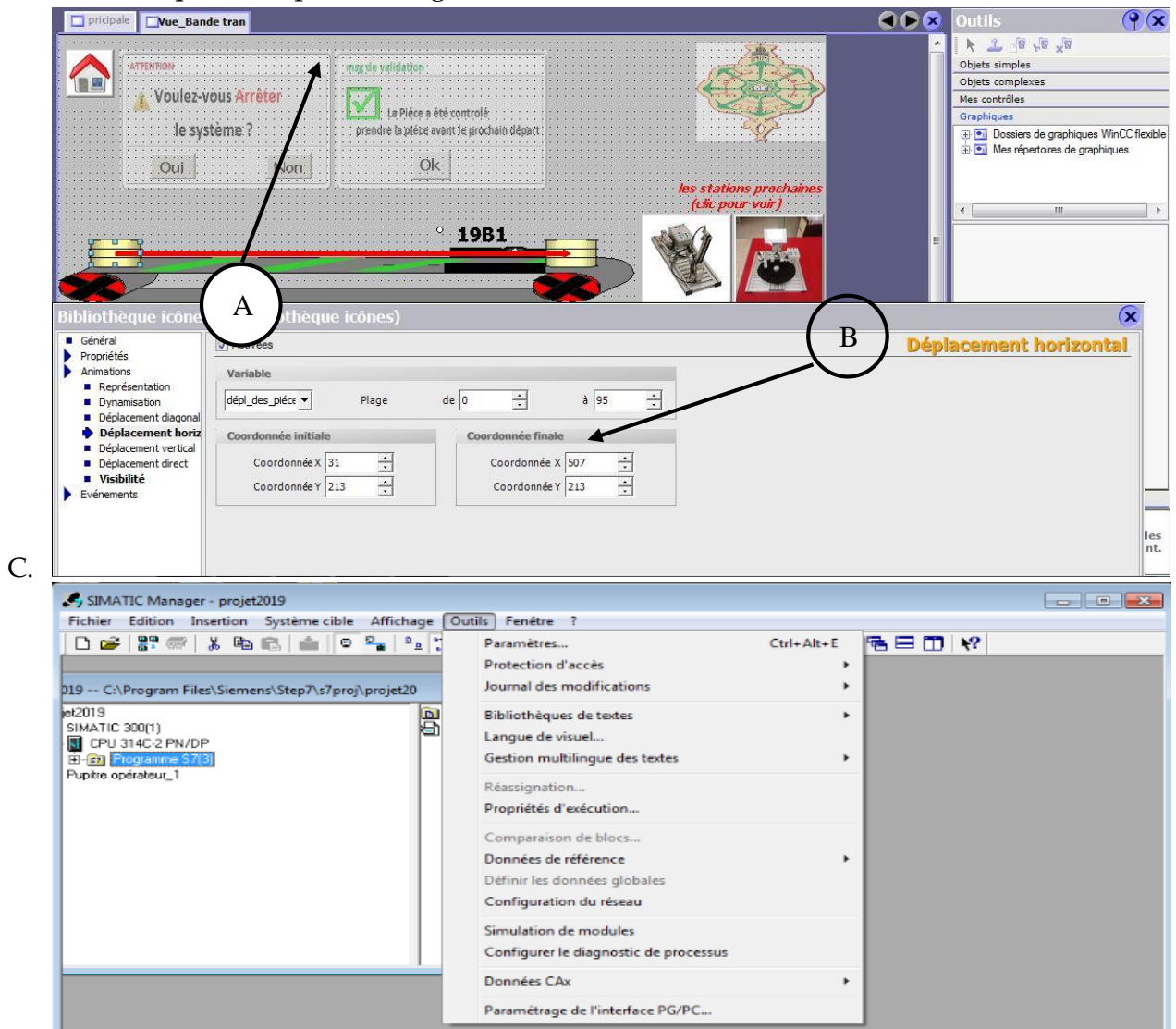


Figure III. 57 : Paramétrage de l'interface.

- Choisir le même nom d'adaptateur « PC Adapter USB A2 »



Figure III. 58 : Adaptateur MPI.

B : Réseau Ethernet

La figure suivante représente la commande de système par l'IHM, nous avons utilisé un câble Ethernet. Pour assurer la connexion entre l'automate et PC.



Figure III. 59 : Connexion PC avec l'automate Via câble Ethernet.

Pour assurer la connexion entre PC et l'automate via le moyen de communication Ethernet, on devez suivre les étapes suivantes :

- Ouvrir SIMATIC manager Et suivez les étapes comme indiqué dans la Figure 3.60 :
 - A. Cliquer sur « outils »,
 - B. Cliquer sur « paramétrage de l'interface PG/PC »,
 - C. Choisir « aucun » et clique sur « ok ».

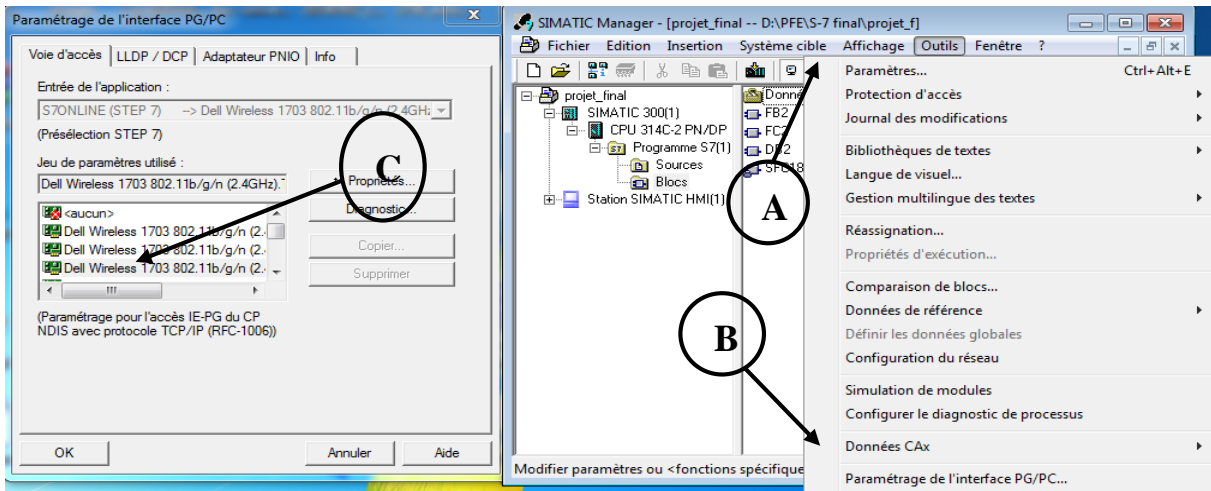



Figure III. 60 : Configuration réseau PC / automate Via câble Ethernet.

- Configurer l’adresse IP de l’ordinateur, il doit être différent par rapport à l’adresse IP de l’API doit être différent de l’adresse IP de l’automate OU step7 et le même pour l’Adres IP de winCC flexible. Les étapes suivantes représentent comment configurer le réseau de l’ordinateur :
- ✓ Cliquer sur le bouton droit de la souris sur cette icone ,
 - ✓ Choisir « ouvrir le centre Réseau partage ».
 - A. Cliquer sur « propriété»,
 - B. Cliquer sur «Protocol internet version 4 »,
 - C. Choisir « utiliser l’adresse IP »,
 - D. entrer l’adresse de l’ordinateur.

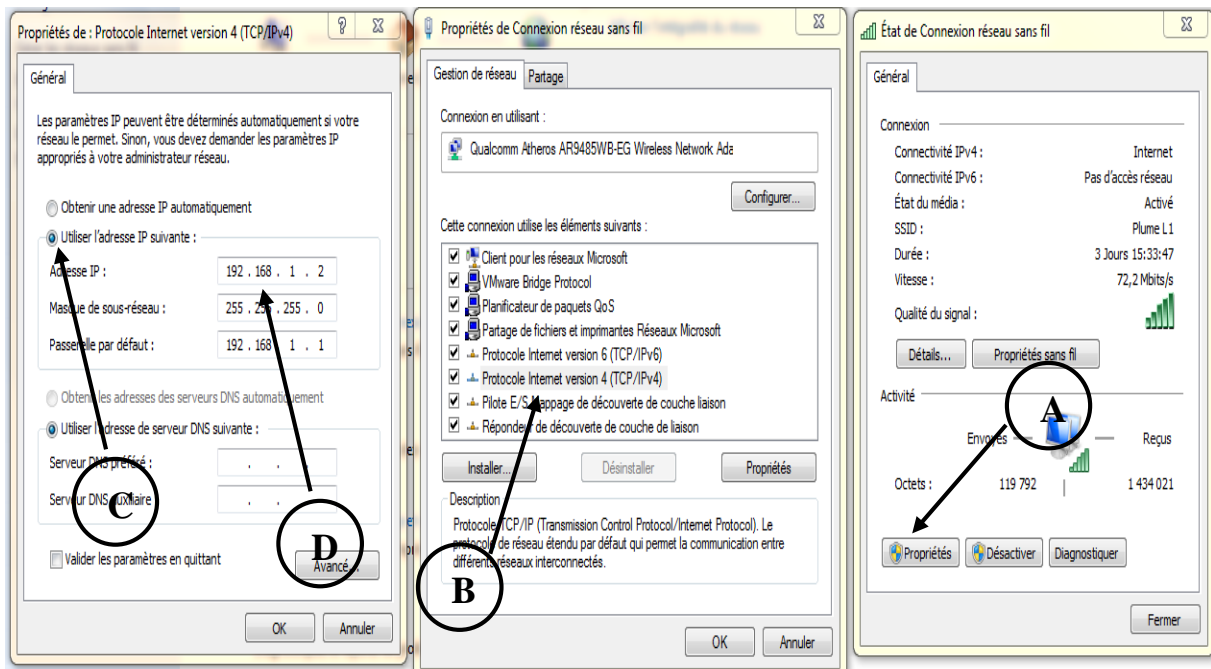


Figure III. 61 : Configuration d’adresse IP de l’ordinateur.

III.6 Conclusion

Step7 et WinCC nous ont permis de tester le programme qu'on a fait selon un cahier de charge bien éclairci, car ces outils sont fournis aux différentes fonctionnalités de test, telles que la simulation du programme, la visualisation ainsi la commande afin de remédier aux erreurs éventuelles et les modifications appropriées avant de procéder à la mise en œuvre dans l'automate.

Comme perspective, nous souhaitons prolonger notre travail, en nous focalisant plus particulièrement sur comment :

- Créer un programme complet de système Mécatronique G-1 en ajoutant les modules restant (Module manipulateur rotatif, Module d'emmagasinage à contrôle cartésien et Caisse contenant les pièces et outils),
- Faire un changement automatique des vues qui représente chaque module, lorsque le module qu'elle représente est fonctionnelle,
- Ajouter la propriété des alarmes dans WinCC.

Conclusion générale

Après la simulation et Intégration du projet en temps réel, aucune erreur n'a été détectée dans le programme. La création d'IHM exige une bonne connaissance et une bonne maîtrise du langage de supervision et du langage dans lequel l'API est programmé pour communiquer et collecter des adresses des variables qui nous intéressent.

L'objectif de notre travail était l'étude et la simulation de la commande, après l'injection en temps réel de système automatisé contenant 03 modules (bande transporteuse, manipulateur, contrôle de pièce) dans un automate programmable Siemens S7-300 se trouvant au niveau de labo de recherche de l'université Abou Bekr Belkaid de Tlemcen.

Ce travail nous a permis d'approfondir nos connaissances théoriques et pratiques acquise durant notre cursus d'études universitaire, Surtout sur les deux logiciels phares utilisés dans le contrôle de processus dans le domaine de production industrielle.

En conclusion, notre étude se résume dans le développement de ces points :

- Ces travaux nous ont permis d'approfondir nos connaissances des systèmes automatisés et des automates programmables.
- Nous avons obtenu plusieurs notions sur le système de commande dans le système étudié. Avec l'aide uniquement du cahier de charge du système, nous avons pu programmer le fonctionnement et faire la supervision de système à partir des connaissances très vagues sur les logiciels utilisés.
- Nous avons augmenté nos connaissances en programmation des automates de la Fameuse marque SIEMENS « S7-300 » à l'aide du logiciel SIMATIC Manager, qui inclut le logiciel de simulation PLCSIM que nous avons utilisé pour simuler le fonctionnement de l'automate programmable industriel dans notre application. Nous trouvons que l'objectif principal de notre étude a été atteint.
- Le projet réel que nous venons de développer reproduit fidèlement le fonctionnement de système de commande, qui est l'élément essentiel utilisé pour le fonctionnement d'un tel système mais en grande échelle.

Le résultat de simulation et l'injection de programme sont satisfaisants. Nous avons voulu que le programme fonctionne et nous sommes arrivés car le système n'a pas été fonctionnel depuis l'expiration de la licence dédiée au logiciel Step7 MicroWin utilisé pour programmer les API S7-200, livré avec ce matériel.

La supervision à travers l'IHM développée va permettre une bonne compréhension du principe de visualisation de l'état de notre Procédé en temps réel ainsi que la commande à distance depuis le PC.

Références

- [1] M. F. H. MlleAREZKI Karima, Etude simulée de la commande de la bouée SPM par un, Université A. MIRA de Bejaia, 2015/2016.
- [2] M.LEMSYEH, Systèmes automatisés, Dossier référence - 1 -, Maroco: Lycée Technique ER-RAZI El Jadida.
- [3] Ikhlef boualem, Contribution à l'étude de supervision industrielle automatique dans un environnement SCADA mémoire de magistère, boumerdes: université Mohamed Bougara, 2009.
- [4] D. R. E. W. Gordon Clarke, Practical Modern SCADA Protocols, Newnes, 2004.
- [5] S. M. E. W. John Park, Practical Data Communications For instrumentation and control, Newnes , 2003.
- [6] w. Daneels, What is SCADA, international conference on accelerator and large experimental physics control systems, 199.
- [7] S. Auder, Automate programmable industriel, l'ELAN, liège, 1991.
- [8] Supervisory Control and Data Acquisition (SCADA) Systems,, Thecnical information bultetin, National Communion System, 04-1 octobre 2004.
- [9] Information et formation, automatisation et entrainements, Programmation niveau 1, Siemens AG, 2003.
- [10] G. C. Paul BRARD, Le GRAFCET, outil de description des automates sequentiel, paris: Technique de l'ingenieur, 2009.
- [11] H. Jean-Louis, Les Automates Programmables, BRUN, 2002.
- [12] SIEMENS, Simatic HMI WinCC flexible 2008 Runtime , Manuel d'utilisation, Siemens AG, 2007.
- [13] E. W. David Bailey, practical SCADA for industry ,Elsevier,pages 47-55, 2003.
- [14] R. L. Krutz, Securing SCADA Systems Edition, Wiley Publishing, 2006.

الملخص:

هذا العمل تم اجرائه في مختبر الأبحاث بجامعة أبو بكر بالقائد بتلمسان , من أجل وضع برنامج جديد للتحكم في نظام المدروس ، لهذا أجرينا دراسة محاكاة حول هذا الموضوع. لقد وجدنا حلاً لاستبدال اليد العاملة بالكامل بواسطة التحكم الآلي لقد قمنا بدمج تقنية التحكم التي هي الإشراف وتم تنفيذ بواسطة برنامجين وهما Simatic manager Step7 و WinCC Flexible 2008 . بعدها قمنا باتباع خطوات إنشاء المشروع. أخيراً ، قمنا بربط النظام المدروس مع التحكم الآلي.

Résumé

Ce travail a été réalisé au sein de labo de recherche de l'université ABOU BEKR BELKAID de Tlemcen. Afin de mettre en place un nouveau programme de contrôle de système étude, nous avons mené une étude sur ce sujet.

Nous avons trouvé une solution pour remplacer complètement l'opérateur humain par un automate programmable industriel, nous avons intégré la technologie de contrôle qui constitue la supervision et la programmation sous logiciel Simatic Manager Step7 et WinCC flexible 2008. Ensuite, on a créé le projet d'automatisation qu'on a injecté le système avec l'API et le suivre en temps réel et analyser son fonctionnement.

Abstact :

This work was carried out within the research laboratory of BOU BEKR BELKAID University of Tlemcen. In order to set up a new study system control program, we conducted a simulation study on this topic.

We have found a solution to completely replace the human operator by an industrial programmable logic controller: we have integrated the control technology that is the supervision and programming software Simatic Manager Step7 and WinCC flexible 2008. Then we follow the steps to create the automation project. Finally, we have injected the system with the API.