

République Algérienne Démocratique et Populaire  
Ministère de L'Enseignement Supérieur et de la  
Recherche Scientifique

Université Mouloud Mammeri De Tizi-Ouzou



Faculté De Génie Electrique Et D'informatique  
DEPARTEMENT D'AUTOMATIQUE

**Mémoire de Fin d'Etudes  
de MASTER PROFESSIONNEL**  
Spécialité : **Automatique Industrielle**

*Présenté par*  
**OURAD Houria**  
**SMAHI Nora**

Mémoire dirigé par Mr KARA Redouane et co-dirigé par Mr GAHAM Mehdi et Melle TEBANI Karima

Thème

**Pilotage d'une Plateforme d'Assemblage  
Robotisée Virtuelle**  
**Apport des techniques Hardware-in-The-  
Loop**

*Mémoire soutenu publiquement le 10 juillet 2018 devant le jury composé de :*

**M<sup>r</sup> MELAH Rabah**  
Pro, UMMTO, Président

**M<sup>me</sup> NAIT ABDESLAM Aldjia**  
MAA, UMMTO, Examinatrice

**M<sup>r</sup> MAIDI Ahmed**  
Pro, UMMTO, Examineur

**Mr KARA Redouane**  
Pro, UMMTO, Promoteur

**Melle TEBANI Karima**  
Grade, CDTA, Co-promotrice

# Remerciement

*Tout d'abord nous tenons à remercier Dieu le tout puissant pour la volonté courage qu'il nous a donné afin de pouvoir finir ce travail.*

*Nous exprimons notre sincère remerciement à toutes les personnes qui nous ont aidés, Monsieur GAHAM responsable du projet et notre encadreur au sein du centre de développement des technologies avancées (CDTA) Baba Hassen et Mademoiselle Tebani notre co-promotrice qui nous a aidée durant toutes cette période du travail. Nous les remercions pour leur orientation, la confiance et la patience qui ont été d'un apport considérable sans lequel ce travail n'aurait pas abouti.*

*Nous tenons également à remercier Monsieur Kara, notre promoteur qui nous a soutenu pour réussir ce travail, et pour la confiance qu'il nous a donnée ainsi que pour sa collaboration et ses précieux conseils.*

*Nous tenons aussi à remercier chaleureusement les membres de jury pour l'honneur qu'ils nous ont fait en acceptent d'évaluer notre projet.*

*Nous présentons enfin, notre profonde gratitude à notre famille et une pensée sincère à nos amis (es), ainsi qu'à tous ceux qui nous ont soutenues de loin ou de près pour réaliser ce travail.*



**Dédicaces**

*C'est avec une grande émotion,  
Je dédier ce modeste travail de fin d'étude  
aux êtres les plus chères :*

*Mon père et ma mère  
qui m'ont soutenu durant mon parcours d'étude,*

*A mes sœurs : Farida, Samia, Lynda ;*

*A mes frères : Mourad et Boussaad ;*

*Et à toute la famille.*

*Et je dédie aussi à tous mes amis et collègues*

*(Celia, Sonia, Lila....)*

**SMAHI Nora**



**Dédicaces**

*A la mémoire de mon cher et  
regretté père*

*A ma mère ;*

*A mon frère et à mes sœurs ;*

*A tous mes proches ;*

*A mes professeurs ;*

*A mes amis et à toutes les  
personnes qui me sont chères.*

***OURAD Houria***

## **Résumé**

La mise en place d'une plateforme d'assemblage robotisée nécessite des phases de validation en environnement émulé appelée « Hardware In the Loop », ça veut dire tester les éléments de commande (contrôleurs SIEMENS : S7-1500 et ET 200 SP) sur une plateforme virtuelle conçue sur un logiciel simulateur appelé Flexsim. Donc l'établissement d'une communication du type socket entre la partie virtuelle et physique en temps réel est nécessaire.

Le lancement du système et la surveillance d'état de connexion se fait par un écran IHM IFP 2200 .

### **Mots clefs:**

Hardware In the Loop, flexsim, socket.

# SOMMAIRE

## INTRODUCTION GENERALE

<b>CHAPITRE I : GÉNÉRALITÉS</b> .....	<b>3</b>
<b>I Introduction</b> : .....	<b>3</b>
<b>I.1 Présentation des systèmes automatisés</b> : .....	<b>3</b>
<b>I.1.1 Définition</b> : .....	<b>3</b>
<b>I.1.2 Objectif de l'automatisation des chaînes de production</b> : .....	<b>3</b>
<b>I.1.3 La structure d'un système automatisé</b> : .....	<b>4</b>
<b>a- La partie commande (PC)</b> : .....	<b>4</b>
<b>b- La partie dialogue (PD)</b> : .....	<b>4</b>
<b>c- La partie opérative (PO)</b> : .....	<b>4</b>
<b>I.2 Généralités sur les automates programmables</b> : .....	<b>6</b>
<b>I.2.1 historique des systèmes automatisés</b> : .....	<b>6</b>
<b>I.2.2 Définition d'un automate programmable industriel</b> : .....	<b>6</b>
<b>I.2.3 Structure d'un API</b> : .....	<b>6</b>
<b>I.2.4 Choix d'un automate programmable industriel</b> : .....	<b>7</b>
<b>I.2.5 Les différentes marques des automates programmables industriels</b> : .....	<b>7</b>
<b>I.3 Conduite et surveillance d'un système automatisé</b> : .....	<b>9</b>
<b>I.4 Architecture de pilotage des systèmes automatisés</b> : .....	<b>9</b>
<b>I.4.1 Les automatismes centralisés</b> : .....	<b>10</b>
<b>I.4.2 Les automatismes décentralisés</b> : .....	<b>10</b>
<b>I.4.3 La décentralisation des entrées/sorties et de la périphérie d'automatisme</b> : .....	<b>11</b>
<b>I.5 Communication</b> : .....	<b>12</b>
<b>I.5.1 Définition des réseaux de terrain industriels</b> : .....	<b>12</b>
<b>I.5.2 Avantages des réseaux de terrain</b> : .....	<b>12</b>
<b>I.6 Supervision des systèmes automatisés</b> : .....	<b>14</b>
<b>I.6.1 Définition de la supervision</b> : .....	<b>14</b>
<b>I.6.2 Fonctionnalités d'un système de Supervision</b> : .....	<b>14</b>
<b>a. la commande</b> : .....	<b>14</b>
<b>b. la surveillance</b> : .....	<b>15</b>
<b>I.7 Conclusion</b> : .....	<b>15</b>

## **CHAPITRE II : LA SIMULATION D'UNE PLATEFORME D'ASSEMBLAGE EN ENVIRONNEMENT ÉMULÉ**

<b>II Introduction :</b>	<b>17</b>
<b>II.1.1 Un convoyeur :</b>	<b>17</b>
<b>II.1.2 Des capteurs :</b>	<b>18</b>
<b>II.1.3 Des robots :</b>	<b>18</b>
<b>II.2 Fonctionnement de la cellule :</b>	<b>19</b>
<b>II.2.1 Présentation d'un modèle de la cellule et son fonctionnement :</b>	<b>19</b>
<b>II.3 Les techniques de validation en environnement émulé (Hardware-in-The Loop) :</b>	<b>20</b>
<b>II.3.1 Définition de Hardware-in-the-loop (HIL):</b>	<b>21</b>
<b>II.3.2 L'intérêt de la simulation HIL :</b>	<b>21</b>
<b>II.3.3 Spécification sur le rôle des techniques HIL dans notre projet :</b>	<b>21</b>
<b>II.4 Simulation :</b>	<b>22</b>
<b>II.4.1 Définition de la simulation :</b>	<b>22</b>
<b>II.4.2 Les avantages et les inconvénients de la simulation :</b>	<b>23</b>
a. <b>Avantages de la simulation :</b>	<b>23</b>
b. <b>Les inconvénients de la Simulation :</b>	<b>23</b>
<b>II.4.3 FlexSim</b>	<b>23</b>
a. <b>Définition du logiciel flexsim :</b>	<b>23</b>
b. <b>Historique :</b>	<b>24</b>
c. <b>Présentation de logiciel Flexsim :</b>	<b>25</b>
d. <b>Les avantages du flexsim:</b>	<b>26</b>
e. <b>La mise en place de la plateforme simulée :</b>	<b>26</b>
<b>II.5 La communication socket :</b>	<b>29</b>
<b>II.6 Conclusion :</b>	<b>30</b>

## **CHAPITRE III: IDENTIFICATION DES COMPOSANTS DE LA CELLULE DU CDTA (coté hardware) ..... 32**

<b>III INTRODUCTION :</b>	<b>33</b>
<b>III.1 Identification les éléments de la cellule existante dans le laboratoire :</b>	<b>33</b>
<b>III.1.1 Présentation l'automate SIMATIC S7_1500 :</b>	<b>33</b>
1) <b>Unité central CPU 1515-2PN (6ES7515-2AM01-0AB0):</b>	<b>34</b>
2) <b>Les caractéristiques de la CPU 1515-2PN :</b>	<b>35</b>

3) L'alimentation et les modules d'entree/ sortie de l'automates S7-1500 : .....	35
III.1.2 Présentation d'ET 200 SP : .....	36
1) Unité central CPU 1512-SP1PN (6ES7512-1DK01-0AB0) : .....	37
2) Les caractéristiques de la CPU 1512-SP1 PN : .....	37
3) Les modules d'entree/ sortie de l'automates ET 200 SP et son alimentation : ..	38
III.1.3 Une IHM SIEMENS IFP2200 Touche Ext écran "22" : .....	38
1) Caractéristiques du SIMATIC IFP2200 (Industrial Flat Panel) : .....	39
III.2 Logiciel Totally Integrated Automation Portal "TIA Portal V14 " : .....	39
III.2.1 Vue du portail et vue du projet : .....	40
III.2.1.1 Vue du portail : .....	40
III.2.1.2 Vue du projet : .....	41
III.2.2 Création d'un projet et configuration d'une station de travail .....	42
III.2.3 Configuration et paramétrage du matériel.....	42
III.2.4 Adressage des E/S : .....	43
III.2.5 WinCC sur TIA portal : .....	44
III.2.6 Les blocs de programme : .....	45
1) Les blocs d'organisation (OB) : .....	45
2) Les blocs de fonctions (FC) : .....	45
3) Les blocs fonctionnels (FB): .....	46
4) Blocs de données d'instance (DB): .....	46
III.2.7 Les langages de programmation : .....	46
a) CONT et LOG – Langages de programmation graphiques : .....	46
b) SCL (Structured Control Language) – Programmation d'algorithmes complexes : .....	46
c) LIST – Liste d'instructions : .....	47
d) GRAPH (SFC) - Programmation de chaînes séquentielles : .....	47
III.3 Conclusion : .....	48

## **CHAPITRE IV : PILOTAGE ET SUPERVISION DE LA CELLULE SIMULÉE**

<b>IV Introduction :</b> .....	<b>50</b>
<b>IV.1 Architecture de pilotage de la cellule et système de communication :</b> .....	<b>50</b>
<b>IV.1.1 Réseau PROFINET :</b> .....	<b>50</b>
<b>a. Définition de PROFINET :</b> .....	<b>50</b>
<b>b. Spécification sur le PROFINET IO :</b> .....	<b>51</b>
<b>IV.1.2 Communication SOCKET :</b> .....	<b>53</b>
<b>IV.1.2.1 La création d'une interface socket sur flexsim :</b> .....	<b>54</b>
<b>IV.1.2.2 La création d'une interface socket sur l'ET 200 SP:</b> .....	<b>55</b>
<b>IV.1.3 Communication S7 :</b> .....	<b>58</b>
<b>IV.2 Pilotage de la cellule :</b> .....	<b>61</b>
<b>IV.2.1 Grafcet de l'application :</b> .....	<b>61</b>
<b>IV.2.2 La structuration du programme :</b> .....	<b>62</b>
<b>IV.3 La supervision de la plateforme :</b> .....	<b>64</b>
<b>IV.3.1 Variable HMI :</b> .....	<b>64</b>
<b>IV.3.2 Table des variables HMI :</b> .....	<b>64</b>
<b>IV.3.3 les vues :</b> .....	<b>64</b>
<b>IV.3.3.1 La hiérarchie des vues :</b> .....	<b>65</b>
<b>IV.3.3.2 La vue initiale :</b> .....	<b>65</b>
<b>IV.3.3.3 La vue globale :</b> .....	<b>65</b>
<b>IV.3.3.4 La vue de réseau :</b> .....	<b>66</b>
<b>IV.4 Conclusion</b> .....	<b>67</b>
<b>Conclusion générale</b> .....	<b>68</b>
<b>BIBLIOGRAPHIE</b> .....	<b>69</b>

**ANNEXE 1: GRAFCET**

**ANNEXE 2 : FlexSim**

## LISTE DES ABREVIATIONS

Tia Portal V14 : Totally Integrated Automation Portal V14.

CDTA : Centre de Développement des Technologies Avancées

PC : Partie Commande.

PO : Partie Opérative.

SAP : Système Automatisé de Production

API : Automate Programmable Industriel.

PLC : Programmable Logic Controller.

CPU: Central Processing Unite.

TOR : Tout Ou Rien

E/S : Entrée/Sortie

RLI : Réseau Local Industriel

HIL : Hardwar-In-tn-Loop

ED : Entreprise Dynamics

TCP : Transmission Control Protocol

IP : Internet Protocol

UDP : User Datagram Protocol

IFP :Industrial Flat Panel

IO : Input Output

IRT : Isochronous Real Time

DVI-D : Digital Visuel Interface

PC : Personale Computer

LED : light-emitting diode

IHM : Interface Homme Machine

GRAFCET : GRAPhe Fonctionnel de Commande Etape-Transition

# Table des figures

Figure I.1 : Les parties d'un système automatisé .....	4
Figure I. 2 : structure d'un SAP.....	5
Figure I.3 Automate Programmable Industriel SIEMENS .....	7
Figure I.4:Les automatismes centralisés .....	10
Figure I.5 : Les automatismes décentralisés .....	11
Figure I.6 : La décentralisation des E/S et de la périphérie d'automatisme .....	11
Figure I.7: Système de contrôle direct .....	12
Figure I.8: Réseau de terrain .....	13
Figure II.1. Un convoyeur .....	18
Figure II.2. Un détecteur photoélectrique à diffusion .....	19
Figure II.3. Un robot .....	19
Figure II.4. Un model représentant de la cellule .....	20
Figure II.5 : figure montante la position du capteur sur convoyeur.....	21
Figure II.6. Le logiciel de simulation Flexsim.....	25
Figure II.7 : Ecran d'accueil de logiciel Flexsim.....	26
Figure II.8 : Bibliothèque de logiciel de FlexSim.....	26
Figure II.9 : Fenêtre de propriétés de la source .....	27
Figure II.10. Convoyeur en boucle formé de plusieurs convoyeurs simple et muni des « photo eyes » .....	28
Figure II.11. Source de palettes et ses propriétés.....	29
Figure II.12 : Combineur.....	29
Figure II.13 : La queue des pièces du type 1 et sa source .....	29
Figure II.14. Robot 1 .....	30
Figure II.15. Figure du poste 1 montrant les liaisons entre l'ensemble de ses éléments composants .....	30
Figure III.1: Automate SIMATIC S7-1500.....	34
Figure III.2 : Les composants de l'automate S7-1500 .....	34
Figure III.3 : CPU 1515-2PN .....	34
Figure III.4 : Constitution de l'API S7-1500 .....	36
Figure III.5 : L'automate ET 200 SP.....	36
Figure III.6: Vue de face de la CPU 1512SP-1 PN avec BusAdapter.....	37
Figure III.7 : L'alimentation PSU100S .....	38

Figure III.8 : Vue d'écran IFP2200 .....	39
Figure III.9 : Vue de portails .....	40
Figure III.10 : Vue du projet.....	41
Figure II.11 : Création d'un projet.....	42
Figure III.12 : Configuration et paramétrage du matériel.....	42
Figure III.13 : Configuration et paramétrage du matériel.....	13
Figure III.14 : Adressage des E/S.....	43
Figure III.15 : Vue SIMATIC HMI .....	44
Figure III.16 : Création d'un nouveau bloc .....	45
Figure III.17 : langage CONT et LOG.....	46
Figure III.18 : langage SCL.....	47
Figure III.29 : Langage LIST .....	47
Figure III.20:Langage SFC .....	47
Figure IV.1:Schéma représentant la connexion des différents éléments de la partie commande sous un réseau PROFINET .....	52
Figure IV.2: switch TP-LINK .....	52
Figure IV.3 : Schéma explicatif de déroulement de l'échange de données entre la partie commande et la plateforme virtuelle .....	54
Figure IV.4 : programmes faits sur flexsim .....	55
Figure IV.5 : La bibliothèque communication .....	55
Figure IV.6 : bloc d'instance de l'instruction TCON.....	56
Figure IV.7: Paramétrage de l'instruction de TCON .....	56
Figure IV.8:bloc d'instance de l'instruction TRCV .....	57
Figure IV.9:L'instruction TSEND qui permet l'envoi de données.....	57
Figure IV.10: bloc d'instance de l'instruction de déconnexion.....	58
Figure IV.11: L'instruction GET.....	58
Figure IV.12: L'instruction PUT .....	59
Figure IV.13: Schéma résumant les différents types de communication utilisés .....	60

**INTRODUCTION GENERALE :**

L'industriel d'aujourd'hui fait face à une multitude de nouvelles contraintes, dont le rythme accéléré de l'innovation, la complexification des processus et l'élévation des normes de satisfaction. Donc il est important pour lui de faire une étude approfondie de son système, d'étudier divers scénarios en prenant en considération toutes les données et informations, tous les aléas relatifs à la bonne marche de son processus avant de le mettre en œuvre.

Là, il fait face à deux autres principaux problèmes : le coût de cette étude, et le temps qu'elle prendra. Dans ce cadre, plusieurs techniques ont été développées pour répondre aux difficultés rencontrées.

Parmi les techniques qui permettent à l'industriel de mener des tests de validation pour un système qu'il envisage concevoir sans utiliser une installation réelle, on trouve les techniques Hardware In The Loop.

Notre projet de fin d'études consiste à exploiter ces techniques pour la validation d'une plateforme d'assemblage robotisée qui sera mise en place au CDTA où notre projet a eu lieu.

L'idée est de concevoir une plateforme virtuelle sur un logiciel simulateur(Flexsim) et la commander par un contrôleur (SIEMENS), ce qui veut dire qu'une communication en temps réel entre les deux est indispensable.

Ce mémoire décrit les étapes qu'on a suivi pour la réalisation de l'idée du projet, cela était fait en quatre chapitres :

Dans le premier chapitre, on a donné un aspect général des systèmes automatisés, ainsi que le rôle de la communication dans leur bonne marche, et défini la supervision et son importance dans les chaînes automatisées.

Dans le second chapitre, on a présenté la plateforme d'assemblage robotisée qu'on veut concevoir dans notre projet, et on a expliqué la nécessité d'utiliser les techniques Hardware-In-The-Loop et l'utilité du logiciel de simulation flexsim dans la réalisation la plateforme virtuelle.

Dans le troisième chapitre, nous avons identifié les composants de la partie contrôle de la cellule qui est constituée d'un automate SIMATIC S7-1500 et quatre entrées/sorties (E/S) décentralisée SIMATIC ET 200 SP ainsi qu'une IHM SIMATIC IFP2200.

Dans le quatrième chapitre nous avons présenté l'architecture de pilotage adoptée pour la commande de la plateforme virtuelle, tout en expliquant les méthodes exploitées pour l'établissement d'une communication entre la partie réelle et la partie virtuelle, on a aussi présenté le système de supervision de la cellule.

# CHAPITRE I : GÉNÉRALITÉS

**I Introduction :**

De nos jours, le développement des systèmes industriels de la production se localise sur l'évolution de la logique câblée à la logique programmée, précisément sur l'automatisation qui est l'une des technologies modernes la plus recherchée dans la rénovation des équipements industriels. Ceci dans le but de simplifier et d'améliorer les conditions de travail, éliminer les tâches répétitives, et d'assurer la sécurité et notamment d'augmenter la production.

Dans ce chapitre, nous allons voir la notion des systèmes automatisés, leurs structures et le rôle de la communication dans la bonne marche de ces systèmes, tout en traitant les avantages que l'automatisme apporte à l'industriel d'aujourd'hui.

**I.1 Présentation des systèmes automatisés :****I.1.1 Définition :**

Un système automatisé ou automatique est un système qui réalise des opérations successives et périodiques ou cycliques. L'homme n'intervient que dans la programmation du système et dans son réglage.

Les buts 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 [1].

**I.1.2 Objectif de l'automatisation des chaînes de production :**

Les objectifs d'une automatisation peuvent être assez variés. On peut citer quelques-uns :

- Accroître la productivité du système pour augmenter la quantité de produit.
- Améliorer la qualité du produit.
- Adaptation à des environnements hostiles pour l'homme.
- L'automatisation entraîne la maîtrise de la disponibilité des produits.

La compétitivité (concurrence) d'un produit final peut être définie comme sa capacité à être bien vendu. La compétitive résulte essentiellement des résultats obtenus sur les facteurs suivants : coût, qualité, innovation, disponibilité. L'automatisation des équipements de production peut améliorer les coûts, la qualité et même la disponibilité des produits. Il est cependant important de vérifier que le produit sur lequel s'applique cette automatisation soit optimisé et répond aux besoins de marché. L'expérience montre qu'une automatisation conduit souvent à remettre en cause le processus de fabrication et donc le produit [1].

### I.1.3 La structure d'un système automatisé :

Tout système automatisé peut être décomposé en trois grandes parties :

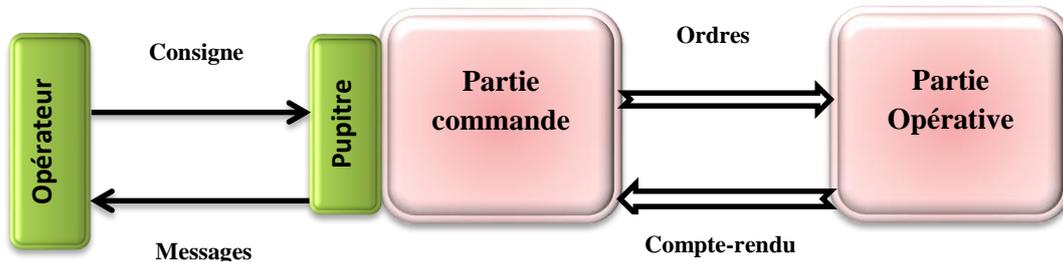


Figure I.1 : Les parties d'un système automatisé

#### a- La partie commande (PC) :

Elle permet à l'opérateur d'établir le programme qui commande la partie opérative. La partie commande reçoit les consignes d'un opérateur, et puis elle adresse des ordres à la partie opérative. Inversement le PC reçoit des comptes rendus de la partie opérative et envoie des signaux à l'opérateur. [1]

#### b- La partie dialogue (PD) :

Elle permet la communication entre l'homme et la machine.

#### c- La partie opérative (PO) :

La partie opérative d'un automatisme est le sous ensemble qui effectue les actions physiques (déplacement, émission de lumière...), mesure des grandeurs physiques (température, humidité, luminosité...) et compte à la partie commande. Elle est généralement composée d'actionneurs (vérins et moteur), des prés actionneurs (distributeurs et contacteurs) et des éléments de détection (capteur, détecteur). Plus simplement, la partie opérative reçoit les ordres de la partie commande et exécute [1].

Structure d'un système automatisé de production :

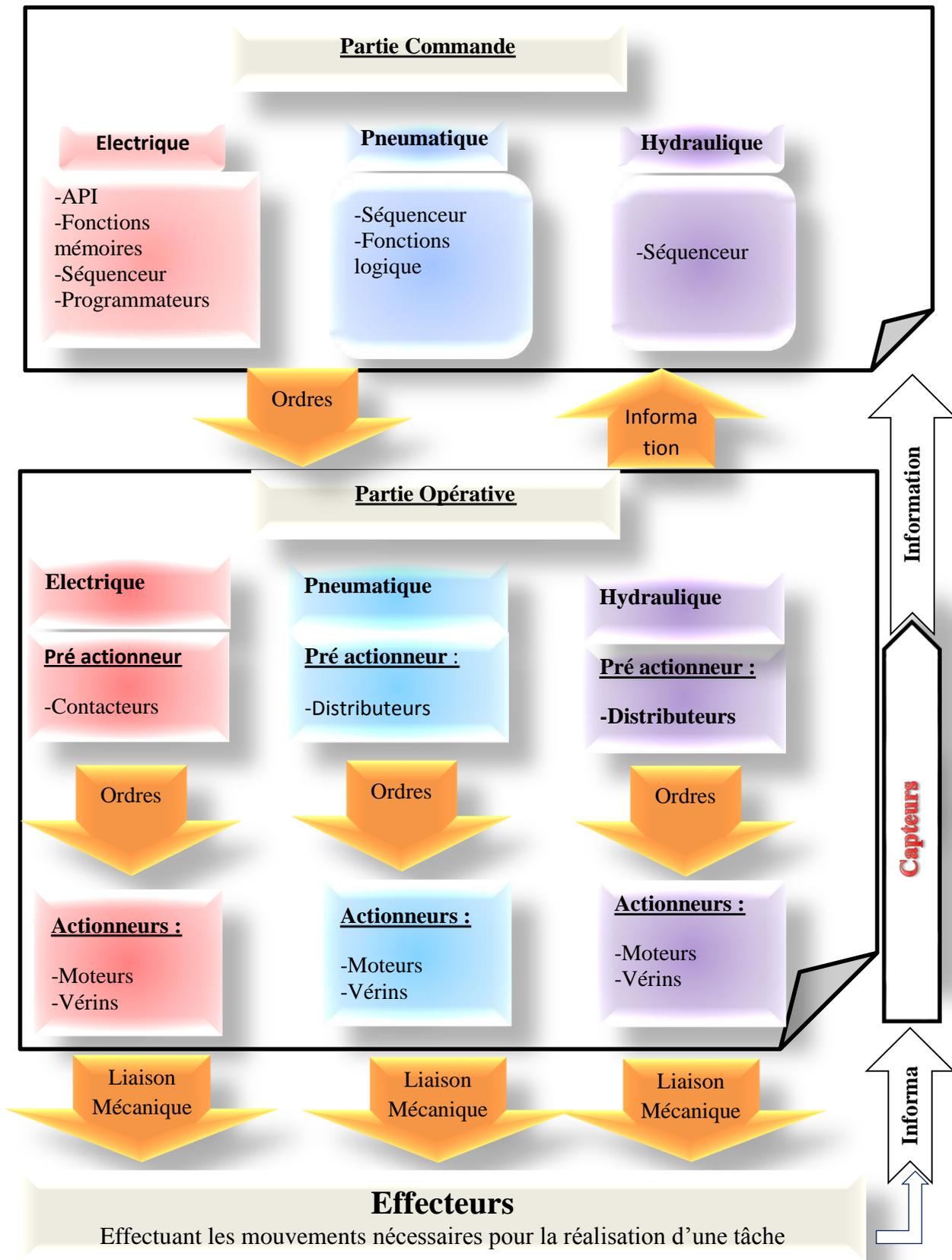


Figure I. 2 : structure d'un SAP

## **I.2 Généralités sur les automates programmables :**

### **I.2.1 historique des systèmes automatisés :**

Les Automates Programmables Industriels (API) sont apparus aux Etats-Unis vers la fin des années soixante, à la demande de l'industrie automobile américaine (General Motors) qui réclamait plus d'adaptabilité de leurs systèmes de commande.

Les ingénieurs américains ont résolu le problème en créant un nouveau type de produit nommé automates programmables. Ils n'étaient rentables que pour des installations d'une certaine complexité, mais la situation a très vite changé, ce qui a rendu les systèmes câblés obsolètes.

Des nombreux modèles d'automates sont aujourd'hui disponibles; depuis les nano automate bien adaptés aux machines et aux installations simples avec un petit nombre d'entrées/sorties, jusqu'aux automates multifonctions capables de gérer plusieurs milliers d'entrées/sorties et destinés au pilotage de processus complexes.

### **I.2.2 Définition d'un automate programmable industriel :**

Un automate programmable industriel est un dispositif électronique programmable destiné à la commande des processus industriels par un traitement séquentiel. Il envoie des ordres vers les pres actionneurs (partie opérative PO coté actionneur) à partir des données d'entrées (capteurs) (partie commande ou PC coté capteur), l'API est un dispositif similaire à un ordinateur utilisé pour automatiser des processus comme la commande des machines sur une chaîne de montage dans une usine, là où les systèmes automatisés plus anciens emploieraient des centaines ou des milliers de relais et de cames, un simple automate suffit.

L'API peut traiter aussi :

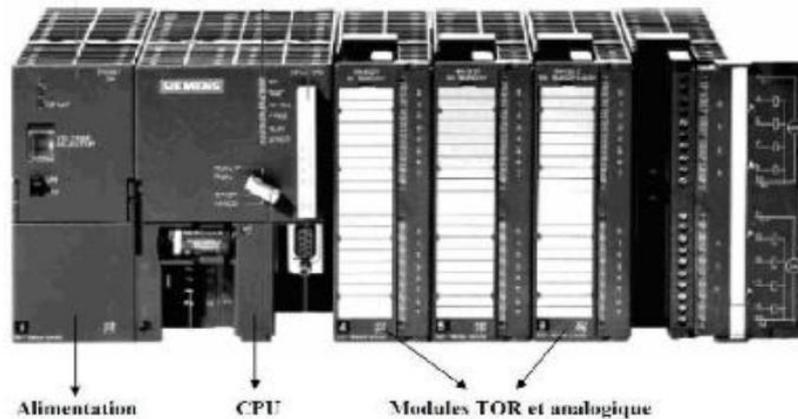
- Des fonctions de calcul arithmétique.
- Temporisation.
- Comptage.
- Des liaisons avec d'autres appareils.

### **I.2.3 Structure d'un API :**

Quel que soit le type de l'automate on peut dire qu'il est composé de 4 parties :

- **L'alimentation** : Elle doit fournir l'énergie nécessaire au fonctionnement correct de l'automate et à l'ensemble de ses cartes.
- **Le processeur** : C'est le cerveau de l'automate qui traite les données. Il contient en mémoire le programme et élabore les ordres de commande. Il sert également de plus en plus à la communication les réseaux Modbus, Ethernet, Canopen, etc.

- **Les modules d'entrées** : Ce sont des cartes spécialisées capables de recevoir en toute sécurité pour l'automate les signaux issus des capteurs, des boutons poussoirs, etc. Elles peuvent être en tout ou rien (TOR) ou analogiques
- **Les modules de sorties** : Ce sont des cartes spécialisés capables de commander en toute sécurité pour l'automate les circuits extérieurs .Elles peuvent être en tout ou rien (TOR), ou analogiques.



**Figure I.3 Automate Programmable Industriel SIEMENS**

#### **I.2.4 Choix d'un automate programmable industriel :**

Les critères de choix essentiels d'un automate programmable industriel sont :

- Les compétences/expériences de l'équipe d'automaticiens en mise en œuvre et en programmation de la gamme d'automate
- La qualité du service après-vente
- Les capacités de traitement du processeur (vitesse, données, opérations, temps réel...)
- Le type des entrées/sorties nécessaire
- Le nombre d'entrées/sorties nécessaire [2]

#### **I.2.5 Les différentes marques des automates programmables industriels :**

- Automate programmable SIEMENS

Siemens est un groupe international d'origine allemande spécialisé dans les hautes technologies et présent dans les secteurs de l'industrie, de l'énergie et de la santé. Siemens a développé une large palette des automates programmables industriels. Cette expérience a été capitalisée dans la conception de la famille S7.

- ❖ Le S7-200, Le S7-300, Le S7-400

- ❖ Le S7-1200 et S7-1500.
- ❖ ET200S, ET200SP, ET 200MP, ET200 M, ET200 iSP
- ❖ ET 200pro, ET200eco PN, ET 200eco

- **Automate programmable Schneider :**

**Schneider Electric**, est une entreprise française qui propose, sous les marques Telemecanique, Quantum et modicon ;

- ❖ Des automates programmables (par exemple gamme TSX Modicon Premium ou Quantum), d'architecture d'automatisme (entrées-sorties distribuées ) des solutions de sécurité, mais aussi sur des logiciels pour l'automatisation, permettant le paramétrage, la programmation ou d'aide à l'exploitation (PL7, Concept, ProWORKX 32, Unity Pro, etc.), des réseaux de communication et bus de terrain (ModBus, ModBus Plus, Profibus, InterBus, DeviceNet, CANopen, AS-i, LonWorks, FIPIO, FIPWAY, Uni-Telway, liaison séries RS232 et RS485...)
- ❖ De dialogue opérateur (produits d'IHM) tels que voyants, terminaux graphiques, PC industriels, pupitres de commandes, boîtes à boutons, les logiciels de supervision (Vijeo Citect, Monitor Pro).

- **Automate programmable ABB :**

Est une entreprise suisse, elle est un acteur majeur des technologies de l'énergie et de l'automation.

Les automates de la marque ABB sont programmés à l'aide de la suite logicielle Controlbuilder c'est un outil très puissant. Il supporte les langages normalisé comme le List, le ladder et le grafcet, il dispose d'un débogueur et d'un simulateur et peut être installé sur Win7 32 ou 64 bits

- **Automate programmable Allen Bradley :**

Les automates de la marque Allen Bradley sont programmés à l'aide de logicielle Rockwell Automation offrent un retour sur investissement plus rapide grâce à des outils haut de gamme et à un contenu accéléré.

Les automates de la marque Allen Bradley offre des architectures modulaires et un éventail d'options d'E/S et de réseau. Des solutions de commande puissantes offre des fonctionnalités haut de gamme, des procédés jusqu'au mouvement, en passant par la sécurité. Conçus pour les applications de commande distribuée ou de supervision, ces automates offrent des performances et une fiabilité exceptionnelles.

### **I.3 Conduite et surveillance d'un système automatisé :**

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

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

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

- Assurer le dialogue entre les intervenants et le système automatisé.
- Assurer la sécurité de ces intervenants dans l'exécution de leurs tâches manuelles.

En outre le modèle de fonctionnement de la Partie Commande, choisi par le concepteur du système, ne correspond qu'à un ensemble de situations prévues, c'est-à-dire retenues par le concepteur parmi un ensemble de situations possibles.

Or il est impératif de pouvoir faire face à des situations non-prévues (donc non retenues en général pour des raisons économiques compte tenu de leur faible probabilité), voire imprévisibles.

Seul un opérateur peut alors intervenir et prendre les décisions requises par cette situation : il assure une fonction de conduite et de surveillance du système automatisé. Cette fonction peut être plus ou moins assistée par un ensemble de moyens (pupitres, informatique...).

Le concepteur devra alors :

- Fournir à l'intervenant (ou lui permettre de prélever) toutes les informations significatives (ou indices) nécessaires à l'analyse de la situation,
- Lui permettre d'agir sur le système, soit directement (dépannage...), soit indirectement (consignes de sécurité, de marches et d'arrêts...) [2].

### **I.4 Architecture de pilotage des systèmes automatisés :**

Ces vingt dernières années, les architectures d'automatismes ont très fortement évolué. Ce phénomène s'amplifie avec l'arrivée des nouvelles technologies de l'information et de la communication.

Ces changements successifs sont dus, d'une part, à l'évolution des besoins des utilisateurs, et d'autre part au développement des technologies par utilisateurs.

### I.4.1 Les automatismes centralisés :

Dans les années 80, les automatismes s'appuyant sur des automates programmables industriels (API), traitaient essentiellement des fonctions séquentielles. En simplifiant, les API:

- Géraient des demandes d'exécution et d'état de l'automatisme (image des entrées).
- Elaboraient des demandes d'exécution d'actions (positionnement des sorties).

Par la suite, les API ont été amenés à gérer de nombreuses fonctions complémentaires comme des fonctions métier, des fonctions de diagnostic système et application, etc.

Les automatismes centralisés géraient tout un ensemble de fonctions qui n'avaient pas forcément d'interactions entre elles. Lorsqu'il y avait déjà un automate dans l'usine, les automaticiens qui devaient intégrer une fonction supplémentaire se posaient simplement la question : l'automate ou le système d'automatisme en place peut-il gérer les E/S supplémentaires et quelle est la capacité de mémoire disponible ?

Bien souvent, l'automatisation supplémentaire était réalisée avec cet automate existant, même si elle n'avait aucun rapport avec l'automatisme résident.

Ces automatismes centralisés amenaient des nombreuses contraintes :

- aucune autonomie des différents sous-ensembles.
- mise en service et maintenance lourdes et difficiles à effectuer du fait de la quantité d'E/S gérées.
- arrêt de l'ensemble des fonctions gérées par l'API en cas de défaut système de cet API ou d'arrêt pour la maintenance du moindre élément de l'outil de production [3].

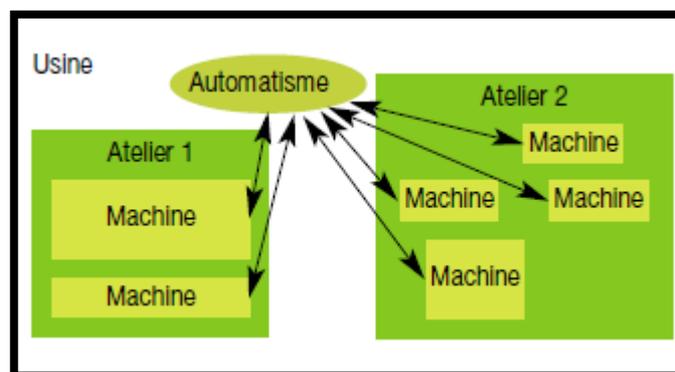


Figure I.4: Les automatismes centralisés

### I.4.2 Les automatismes décentralisés :

Du fait des contraintes imposées par les systèmes centralisés, les utilisateurs se sont orientés vers une segmentation de l'architecture. Celle-ci a été faite en découpant l'automatisme en entités fonctionnelles.

Elle permet de simplifier les automatismes en réduisant le nombre d'E/S gérées et présente donc l'avantage de faciliter la mise en service et la maintenance. Cette segmentation a généré le besoin de communication entre les entités fonctionnelles. La fonction de communication est devenue la clef de voûte de la conception des architectures d'automatismes.

Les constructeurs d'API ont donc créé des offres de réseaux locaux industriels (RLI) afin d'assurer une communication efficace entre les différents API [3].

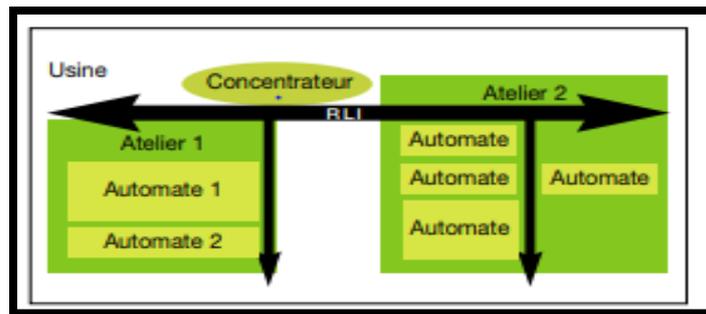


Figure I.5 : Les automatismes décentralisés

#### I.4.3 La décentralisation des entrées/sorties et de la périphérie d'automatisme :

La décentralisation des entrées/sorties et de la périphérie d'automatisme à la demande des utilisateurs finaux, notamment pour faire baisser les coûts de câblage, il a été nécessaire de prendre en compte la topologie des automatismes. Sur des sites plus étendus, il est souvent nécessaire de gérer un nombre de points diffus importants et de prendre en compte les fonctions métier réparties (variation de vitesse, dialogue homme/machine, pesage...). La réponse des constructeurs de produits d'automatismes est arrivée avec les réseaux et bus de terrain. Ceux-ci ont permis de gérer dans un premier temps des E/S décentralisées puis la périphérie d'automatisme. Ces réseaux de terrain contribuent à réaliser des gains de câblage importants, mais surtout ils permettent de rendre accessibles des services (diagnostic, programmation...) sur tout le site [3].

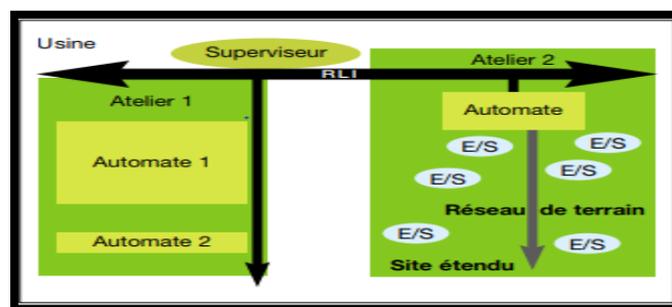


Figure I.6 : La décentralisation des E/S et de la périphérie d'automatisme

## I.5 Communication :

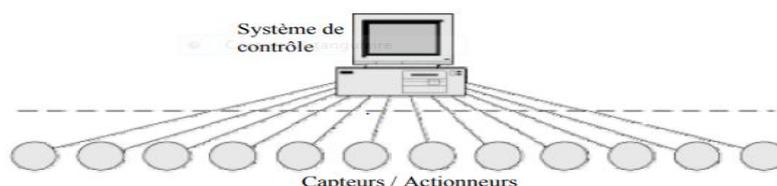
L'automatisation, puis l'informatisation des systèmes de production ont montré que pour l'amélioration de la productivité et de la diversité de la production, il ne suffisait pas de maîtriser la gestion de production, ou d'avoir une machine-outil à commande numérique sophistiquée, ou encore un système de contrôle-commande évolué et quelques capteurs et actionneurs autour. Mais encore il faut faire communiquer tous ces éléments constitutifs d'un système de production dans sa globalité. Faire communiquer c'est permettre d'échanger des informations de natures différentes et ce parfois nécessairement en temps réel. La mise en place d'un réseau de terrain est indispensable dans une chaîne de production.

### I.5.1 Définition des réseaux de terrain industriels:

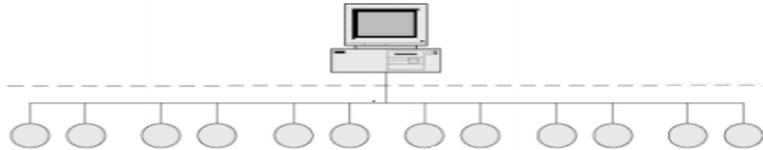
C'est un réseau de communication numérique reliant différents types d'équipements d'automatisme intelligents ou à intelligence limitée pour permettre leur coopération tel que : les capteurs, les actionneurs, les automates programmables, les machines à commande numérique, les robots, les calculateurs ...etc. Dans les réseaux de terrain, la taille des messages échangés est assez faible comparativement aux autres types de réseaux, locaux ou à grandes distances. Les flux d'information sont plutôt périodiques et l'aspect contraint de temps (temps réel) est prioritaire. Les réseaux de terrain fonctionnent au sein d'environnements perturbés du point de vue électromagnétique, le support utilisé est généralement un câble coaxial blindé ou une paire torsadée travaillant en mode différentiel ou encore une fibre optique pour une plus grande immunité au bruit. Les distances de communications sont assez faibles par rapport aux autres types de réseaux, souvent inférieures à quelques dizaines de mètres. Une topologie en bus est généralement adoptée pour sa facilité de mise en place [4].

### I.5.2 Avantages des réseaux de terrain :

Le but initial des bus de terrain était de remplacer les anciens systèmes centralisés en distribuant le contrôle, le traitement des alarmes, le diagnostic aux différents équipements qui sont devenus de plus en plus intelligents.



**Figure I.7: Système de contrôle direct**



**Figure I.8: Réseau de terrain**

Les anciens systèmes de communication industriels utilisaient la boucle de courant 4- 20 mA, qui est un moyen de transmission analogique permettant de transmettre un signal analogique sur une grande distance sans perte ou modification, pour relier les équipements aux machines de contrôle. Après l'apparition de la communication numérique, cette technique a été rapidement remplacée par les bus de terrain. Cela permet plusieurs avantages:

- Réduction des coûts initiaux :

- Réduction massive du câblage : un seul câble en général pour tous les équipements au lieu d'un par équipement

- Possibilité de réutiliser le câblage analogique existant dans certains cas.

- Réduction du temps d'installation

- Réduction du matériel nécessaire à l'installation

- Réduire le coût d'exploitation en :

- Augmentant les performances de l'automatisme

- Réduisant les coûts des extensions futures.

- Réduction du coût de maintenance :

- Complexité moindre donc moins de maintenance (fiabilité accrue)

- Maintenance plus aisée : temps de dépannage réduit, localisation des pannes possibles grâce à des diagnostics en ligne («on line») donc à distance

- Outils de test dédiés (programmes analyseurs...)

- Flexibilité pour l'extension du bus de terrain et pour les nouveaux raccordements.

La distribution du contrôle et sa numérisation permettent d'atteindre des performances intéressantes dans les réseaux de terrain :

- Précision : communications numériques : la donnée numérique transférée est sans erreur de distorsion, de réflexion... contrairement à un signal analogique

- Les données et mesures sont généralement disponibles à tous les équipements de terrain

- Communications possibles entre deux équipements sans passer par le système de supervision

- La structure distribuée permet de faire résider des algorithmes de contrôle au niveau de chaque équipement de terrain (chaque nœud).
- Accès à des variables multiples pour un nœud [4].

### **I.6 Supervision des systèmes automatisés :**

La complexité des systèmes dans lesquels l'homme est impliqué aujourd'hui conduit à l'émergence de systèmes de traitement de l'information de plus en plus sophistiqués et incontournables où la prise de décision est de plus en plus difficile. La supervision homme-machine des systèmes de production s'inscrit typiquement dans cette problématique.

#### **I.6.1 Définition de la supervision :**

Les définitions de la supervision, fournies par les dictionnaires sont les suivantes : « surveiller et contrôler l'exécution d'une opération ou la réalisation d'un travail accompli par d'autres » (Larousse) ; « contrôler sans entrer dans les détails » (Robert). Les deux notions de « surveiller et contrôler l'exécution d'une opération » et « sans entrer dans les détails ». En terme technique, un système de supervision est un système qui permet à l'opérateur l'assistance et la visualisation de l'état et de l'évolution d'une installation automatisée de contrôle de processus, avec une mise en évidence des anomalies (alarmes).

#### **I.6.2 Fonctionnalités d'un système de Supervision :**

Les deux majeures fonctionnalités d'un système de supervision sont la commande et la surveillance d'une installation automatisée.

##### **a. la commande :**

Le rôle de la commande est de faire exécuter un ensemble d'opérations (élémentaires ou non suivant le niveau d'abstraction auquel on se place) au procédé en fixant des consignes de fonctionnement en réponse à des ordres d'exécution. Il s'agit de réaliser généralement une séquence d'opérations constituant une gamme de fabrication dans le but de fabriquer un produit en réponse à une demande d'un client. La commande regroupe toutes les fonctions qui agissent directement sur les actionneurs du procédé qui permettent d'assurer :

- le fonctionnement en l'absence de défaillance.
- la reprise ou gestion des modes.
- les traitements d'urgence.
- une partie de la maintenance corrective [5].

Les fonctions de commande en marche normale sont:

- L'envoi de consignes vers le procédé dans le but de provoquer son évolution.
- L'acquisition de mesures ou de compte-rendu permettant de vérifier que les consignes envoyées vers le procédé produisent exactement les effets escomptés.
- L'acquisition de mesures ou d'informations permettant de reconstituer l'état réel du procédé et/ou du produit.

**b. la surveillance :**

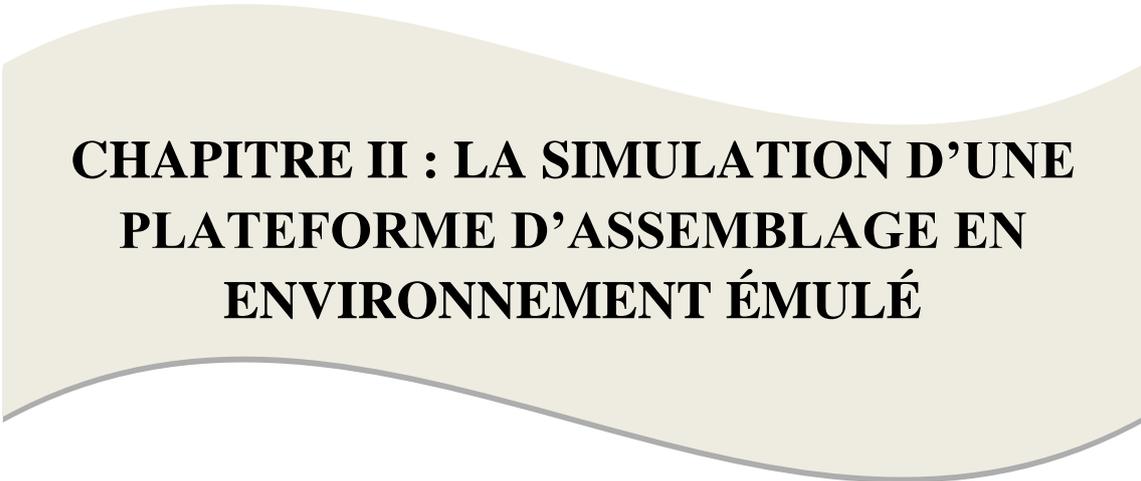
La partie surveillance:

- Recueille en permanence tous les signaux en provenance du procédé et de la commande.
- Reconstitue l'état réel du système commandé.
- Fait toutes les inférences nécessaires pour produire les données utilisées pour dresser des historiques de fonctionnement.
- Met en œuvre un processus de traitement de défaillance le cas échéant.

Dans cette définition, la surveillance est limitée aux fonctions qui collectent des informations, les archivent, font des inférences, etc. sans agir réellement ni sur le procédé ni sur la commande. La surveillance a donc un rôle passif vis-à-vis du système de commande et du procédé [5].

**I.7 Conclusion :**

Dans ce chapitre, on a donné un aspect générale sur les systèmes automatisées et quelques notions de base sur les quelles notre projet sera fondé. Dans le chapitre qui suit-on présentera notre logiciel de simulation Flexsim et l'architecture de notre plateforme virtuelle sur laquelle on va travailler.



**CHAPITRE II : LA SIMULATION D'UNE  
PLATEFORME D'ASSEMBLAGE EN  
ENVIRONNEMENT ÉMULÉ**

## **II Introduction :**

Dans ce chapitre, on présentera la plateforme d'assemblage robotisée qu'on veut concevoir dans notre projet, et on expliquera la complexité de sa réalisation, et la nécessité d'utiliser les techniques Hardware-In-The-Loop pour sa validation avant de passer à l'implémentation réelle.

On expliquera l'utilité du logiciel de simulation flexsim dans la réalisation d'une plateforme virtuelle semblable à la plateforme réelle, les avantages qu'il nous offre, notamment sa capacité de communication qui nous sera indispensable pour atteindre le principal but de ce projet.

### **II.1 Description de la cellule flexible :**

En industrie, on appelle une plateforme d'assemblage l'ensemble d'éléments (machine, capteurs, actionneurs...) interconnectés dans le but d'assembler plusieurs pièces, et donner à la fin un produit prêt à la commercialisation. On dit qu'elle est robotisée lorsqu'elle contient des robots ou des bras robotisés dans sa constitution. Dans notre projet, le système sur lequel on va travailler est une cellule de ce genre.

Pour décrire cette cellule, on commence par présenter ses composants principaux, qui sont les suivants :

#### **II.1.1 Un convoyeur :**

C'est tout d'abord un outil qui sert à véhiculer des pièces différentes entre plusieurs postes de fabrication ou de montage, dans notre système, il sera basé sur une structure en boucle sur lequel se déplacent des palettes. Son sens de rotation est unidirectionnel. Il sera entraîné par un moteur asynchrone avec un variateur de vitesse qui régulera sa vitesse.

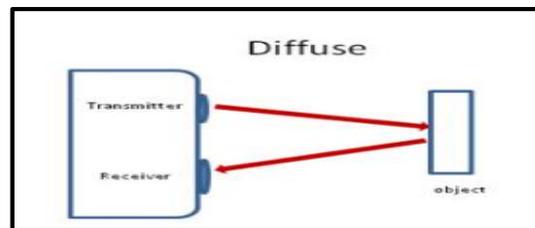


**Figure II.1. Un convoyeur**

**II.1.2 Des capteurs :**

Ce sont les éléments permettant d'obtenir des informations précises sur l'évolution de système.

Dans notre cas le type de capteurs le plus convenable ce sont des détecteurs photoélectriques qui permettent la détection d'objets de toutes natures, ils réalisent la détection d'une cible (objet ou personne) au moyen d'un faisceau lumineux. Ses deux constituants de base sont un émetteur et un récepteur de lumière, La détection est effective quand la cible pénètre dans le faisceau lumineux et modifie suffisamment la quantité de lumière reçue par le récepteur pour provoquer un changement d'Etat de la sortie [6].



**Figure II.2. Un détecteur photoélectrique à diffusion**

**II.1.3 Des robots :**

Ce sont des machines chargées d'effectuer une ou plusieurs tâches de manière autonome sur une chaîne de montage, ou de soudage...etc [7].

Dans notre cellule, les robots utilisés seront des robots fixes qui vont accomplir la tâche d'assemblage de pièces sans se déplacer de leurs places.



**Figure II.3. Un robot**

## II.2 Fonctionnement de la cellule :

### II.2.1 Présentation d'un modèle de la cellule et son fonctionnement :

La cellule flexible automatisée robotisée va contenir quatre postes de travail. Dans chaque poste, il se trouve un stock de pièces à assembler avec la palette, et un robot qui placera la pièce sur la palette dans un petit espace du travail (qu'on appelle combineur) après l'avoir prise du convoyeur.

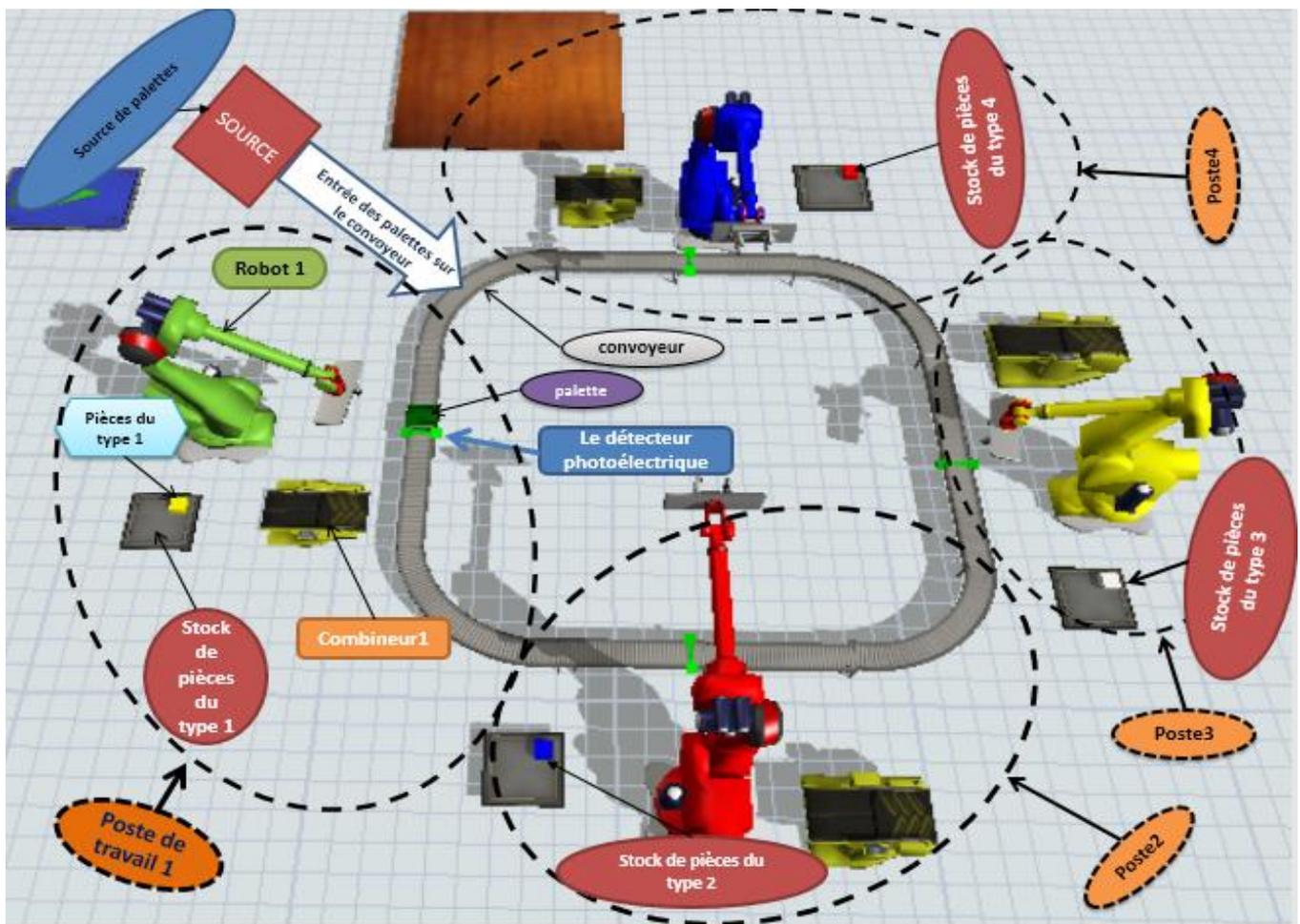


Figure II.4. Un model représentant de la cellule

Dans cette application, on veut réaliser un fonctionnement automatisé de cette plateforme d'assemblage, dont on ne va intervenir qu'une seule fois, c'est à la mise en marche du système au tout début de lancement, cela en appuyant sur un bouton poussoir qui mettra le convoyeur en marche bien après avoir initialisé les quatre robots.

Après le lancement du système, le convoyeur s'arrêtera automatiquement devant chaque poste du travail à la détection de la palette par le capteur photoélectrique pour permettre au robot de

la prendre et la mettre sur son espace du travail où il fera sa tâche d'assemblage. Le convoyeur se remet en marche après 3 secondes de son arrêt, cela pour éviter l'arrêt de tout le système durant le temps où un robot fait sa tâche, et permettre à tous les robots de travailler en parallèle, et gagner en termes de temps.

La remise de la palette sur le convoyeur après l'assemblage se fera sans faire arrêter le convoyeur.

### II.3.2. Cahier de charge de l'application :

- Mise en marche du système en appuyant sur un bouton poussoir.
- Arrêt du convoyeur à la présence d'une palette devant le poste.
- Prise de la palette du convoyeur par le robot et accomplissement de son assemblage avec la pièce du type contenu dans le poste.
- Redémarrage du convoyeur.
- Remise de la palette sur le convoyeur pour passer aux autres postes.

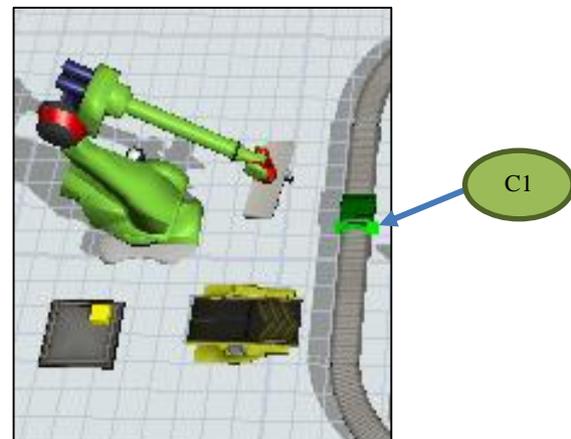


Figure II.5 : figure montante la position du capteur sur le convoyeur

### Remarque :

Dans le dernier poste, la palette ne sera pas remise sur le convoyeur après l'assemblage, elle sera mise par le robot sur un stock destiné aux produits finaux.

### II.3.3 Les techniques de validation en environnement émulé (Hardware-in-The Loop) :

Sous un certain point de vue, de nombreux systèmes automatisés de production sont classés comme des systèmes à événements discrets, ces derniers sont définis comme des systèmes dont l'état change seulement à certains instants lors de l'occurrence d'événements particuliers (état stable entre ces instants) [8].

En général, la conception de ces systèmes nécessite d'abord une analyse puis des tests de validation pour passer par la suite à l'implémentation réelle. Dans ce cadre, les techniques de validation en environnement émulé sont très répandues.

La réalisation de la cellule dont on a parlé dans les éléments précédents nous incite à utiliser ces techniques, mais commençons d'abord par expliquer comment et pourquoi les utiliser ?

### **II.3.1 Définition de Hardware-in-the-loop (HIL):**

La simulation Hardware-in-the-loop (HIL) est un type de simulation en temps réel. On utilise la simulation HIL pour tester les caractéristiques des contrôleurs (vitesse du traitement de processeur, fonction de communication...etc.). La simulation HIL montre comment le contrôleur répond, en temps réel, à des stimuli virtuels réalistes. On peut également utiliser HIL pour déterminer si un modèle de système physique est valide.

Dans la simulation HIL, on utilise un ordinateur en temps réel en tant que représentation virtuelle de modèle d'installation et d'une version réelle de contrôleur qu'on désire tester [9].

### **II.3.2 L'intérêt de la simulation HIL :**

Aujourd'hui, les techniques de validation en environnement émulé sont indispensables pour la vérification et la validation des systèmes dans diverses disciplines, telles que l'aéronautique, l'automobile, et l'environnement industriel en général. Dans le contexte de l'automobile par exemple, les HIL permet aux développeurs de valider de nouveaux matériels et logiciels pour trouver des solutions dans le but d'améliorer les produits, en respectant les exigences de qualité du marché et dans un temps restrictif.

Dans l'industrie, l'approche HIL a de nombreux avantages car la simulation est effectuée avec le vrai programme fonctionnant dans le vrai contrôleur. Ce peut être très important, surtout quand les différentes échelles de temps sont considérées d'un point de vue réel. Dans ce contexte, le modèle de système de fabrication fonctionne dans le PC et contrôlé avec un PLC réel [10].

### **II.3.3 Spécification sur le rôle des techniques HIL dans notre projet :**

La réalisation de la plateforme robotisée qu'on a décrite auparavant est réellement très complexe, Parce que c'est nécessaire d'étudier chacun de ses éléments à part, le tester avant de l'inclure dans l'ensemble, puis analyser l'interaction mutuelle des composants pour établir un automatisme à la fois efficace, rapide et sécurisé. D'ailleurs ce projet est divisé en plusieurs parties.

La partie sur laquelle on se focalise est la partie des contrôleurs (les PLC), on veut tester les contrôleurs dans des situations réelles, et cela sans utiliser du matériel réel ou du système physique (installation) dont on ne dispose pas de la majorité des éléments constituant à l'heure actuelle. Mais cela est toujours possible grâce aux techniques de validation en environnement émulé, il nous suffit de disposer de ces contrôleurs et d'un logiciel de simulation sur lequel on peut concevoir un modèle représentant de la cellule réelle.

En l'occurrence le choix d'un logiciel de simulation bien adapté à la réalisation réelle est très important, d'abord il faut choisir un qui est dédié aux systèmes de production et d'assemblage, qui dispose des éléments constituant de la cellule dans sa bibliothèque, et bien sûr qui a la capacité de communiquer en temps réel avec d'autres logiciels et matériels.

Nous, en tant que stagiaires au Centre de Développement des Technologies Avancées(CDTA), où cette réalisation aura lieu, on avait le droit d'avoir une licence d'un logiciel qui répond aux exigences qu'on a précisées auparavant, il s'agit d'un logiciel de simulation 3D nommé FlexSim, il sera donc le logiciel simulateur dont on aura besoin par la suite.

## **II.4 Simulation :**

Avant d'entamer la présentation du flexSim, son historique et son rôle dans notre travail, on explique d'abord c'est quoi une simulation, et à quoi elle sert ?

### **II.5.1 Définition de la simulation :**

La simulation est l'un des outils d'aide à la décision les plus efficaces à la disposition des concepteurs et des gestionnaires des systèmes complexes. Elle consiste à construire un modèle d'un système réel et à conduire des expériences sur ce modèle afin de comprendre le comportement de ce système et d'en améliorer les performances.

Les applications de la simulation sont innombrables. Parmi les domaines dans lesquels elle est plus utilisée, on peut citer :

- L'informatique : recherche de configurations, réseaux, architecture de bases de données, ...
- La production : gestion des ressources de fabrication, machines, stocks, moyens de manutention, ...
- La gestion : marketing, tarification, prévisions, gestion du personnel, l'administration : gestion du trafic, du système hospitalier, de la démographie, ...
- L'environnement : pollution et assainissement, météorologie, catastrophes naturelles.

#### **II.4.2 Les avantages et les inconvénients de la simulation :**

##### **a. Avantages de la simulation :**

La simulation a plusieurs avantages. On cite les plus importants :

- ✓ Elle permet de concevoir de nouveaux systèmes, de tester de nouvelles procédures opératoires ou organisationnelles dessus, des règles de décision, des flux d'information, de détecter des goulots d'étranglement, etc. sans interrompre le fonctionnement du système réel, ni investir dans de nouvelles ressources.
- ✓ Elle permet aussi d'étudier certains phénomènes en accélérant ou ralentissant le temps au cours de l'étude, chose infaisable si l'expérimentation est réalisée avec un système réel.
- ✓ La simulation peut aussi apporter une meilleure compréhension de l'interaction entre les variables d'un système et leur influence sur ses performances [2].

Par ailleurs, certaines études sont impossibles à mener sans l'utilisation d'un modèle en raison de leur coût éventuel, de leur dangerosité ou de leur délai de mise en place [2].

##### **b. Les inconvénients de la Simulation :**

En parallèle avec ses avantages, la simulation a également quelques inconvénients. On cite quatre :

- ✓ L'établissement d'un modèle requiert un entraînement spécifique et s'apprend avec une expérience effective. Dans ce contexte, on affirme qu'en négligeant d'adopter une vraie méthodologie de simulation, on trouve des conclusions erronées.
- ✓ Les résultats peuvent être difficiles à interpréter, en raison du caractère stochastique de la majorité des modèles. Ainsi, l'analyse des résultats demande certaines connaissances en statistiques.
- ✓ La simulation peut être consommatrice de temps et de ressources ; la sous-estimation de la charge de travail peut aboutir à des objectifs non atteints.
- ✓ La simulation est parfois utilisée à tort dans des cas où une solution analytique est possible et même préférable. Cela est particulièrement vrai dans certains systèmes de files d'attente ou des modèles analytiques existants déjà.

Par ailleurs, on souligne d'autres points négatifs qui tendent à ralentir l'expansion de la simulation [11].

#### **II.4.3 FlexSim**

##### **a. Présentation du logiciel flexsim :**

FlexSim est un logiciel de simulation 3D conçu pour modéliser des systèmes logistiques et de production dans tout type d'industrie : manufactures, emballage, entrepôts, manutention...etc. FlexSim est doté d'une gamme complète de puissants outils, comme la visualisation 3D à l'échelle, ou encore les tableaux de bord pour centraliser toutes les informations statistiques qui apportent de l'éclaircissement sur le comportement et les performances de votre système. FlexSim facilite la prise de décision sans risques grâce à la comparaison des résultats sur les scénarios simulés pour optimiser le flux des produits, la gestion des ressources humaines et matérielles, le plan d'implantation au sol, et tout autre aspect du système. FlexSim est en quelque sorte votre "boule de cristal" pour optimiser votre système avant d'implémenter les changements sur votre processus réel [12].



**Figure II.6. Le logiciel de simulation Flexsim**

### **b. Historique :**

FlexSim Software Products a été fondé par Bill Nordgren en 1993, Roger Hullinger, et Cliff King, à l'origine sous le nom de F&H Simulations. F&H Simulations fournissait des services de vente, support et formation sur le logiciel de simulation Taylor II - propriété de la société hollandaise F&H Simulation B.V (F&H Holland).

En 1998, F&H Holland a développé la première génération de moteur de simulation 3D orienté objet: Taylor ED (Enterprise Dynamics). F&H Simulations a contribué au développement d'objets robustes intégrés dans Taylor ED. En complément, F&H Simulations a contribué à vendre, conseiller, et former sur le nouveau logiciel.

En 2000, F&H Holland a été racheté. F&H Simulations a saisi cette opportunité pour devenir indépendant. Dr. Eamonn Lavery et Anthony Johnson ont rejoint la société pour superviser le développement d'un logiciel de simulation 3D orienté objet de nouvelle génération : FlexSim. F&H Simulations a changé son nom en FlexSim Software Product.

La version 1.0 de FlexSim a été délivrée en février 2003, dotée d'un tout nouveau moteur de simulation dernier cri, un environnement de modélisation en 3D, et une intégration en C. Depuis sa première version, FlexSim est devenu un standard de référence dans le domaine de la simulation par événements discrets [12].

### c. Présentation de logiciel Flexsim :

#### ➤ Ecran d'accueil :

L'écran d'accueil offre la possibilité de créer un nouveau modèle soit par le menu **File/New Model** soit en utilisant directement **New Modèle** à gauche de la page.

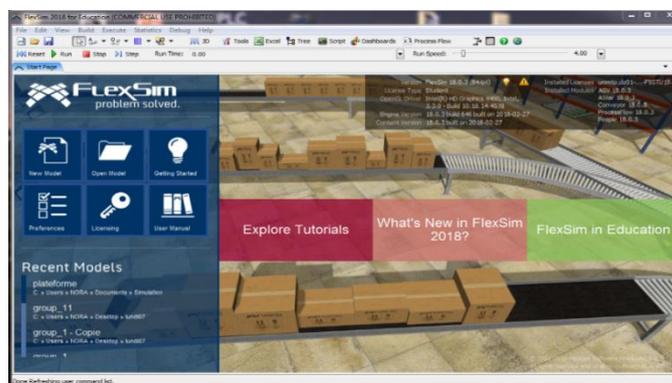


Figure II.7 : Ecran d'accueil de logiciel Flexsim.

#### ➤ Librairie de modélisation :

- ✓ Ressources fixes (Fixed Resources)
- ✓ Ressources mobiles (Task Executors)
- ✓ Réseaux de déplacement (Travel Networks)
- ✓ Convoyeurs
- ✓ Visuel
- ✓ Astar
- ✓ AGV (Automated Guided Vehicle)
- ✓ Fluide [13].



Figure II.8 : Bibliothèque de logiciels de FlexSim

### Remarque :

Tous les objets contenus dans la bibliothèque possèdent des paramètres pouvant être modifiés. On peut accéder au menu d'édition en cliquant deux fois sur l'objet ou bien en cliquant grâce au bouton droit de la souris sur l'objet et on sélectionne « Propriétés ».

Chaque classe d'objet possède des caractéristiques qui lui sont propres. La fenêtre de propriétés possède plusieurs onglets reliés à différentes fonctions. Voici par exemple la fenêtre de paramétrage de la source, avec :

**L'onglet Source :** on peut régler le mode d'arrivée

« Arrival Style » dont il existe trois modes d'arrivées pour les sources : un temps d'inter arrivé « interarrival time », une planification « arrival schedule » et enfin une séquence « arrival sequence ».

- **L'onglet Flow :** dans cet onglet il s'agit du paramétrage du flux des produits et du choix de leur destination

« Send To Port » et on peut aussi utiliser un transporteur tel un robot par exemple, en cochant « Use Transport » et en indiquant le transporteur évidemment.

- **Triggers :** on peut définir, dans cet onglet, le comportement de l'Objet en fonctions d'évènements par exemple : lors de la réception d'un message, de la création d'un produit ou bien lors de la sortie du produit, etc.

- **Labels :** dans l'onglet suivant onglet, on peut étiqueter nos produits lors de leur passage par tel ou tel Objet.

#### d. Les avantages du flexsim:

Flexsim est un logiciel orienté à la simulation des systèmes à événements discrets, son environnement est utilisé pour développer, modéliser, simuler, visualiser, et surveiller le flux dynamique des activités du processus et du système.

FlexSim est une suite complète d'outils conçus pour développer et compiler des applications de simulation.

Aujourd'hui, FlexSim est utilisé pour la modélisation dans la production, fabrication, stockage et livraison, systèmes de transport et autre champs.

Dans FlexSim, il est possible d'importer des données de différents processus collectés du monde réel et les transformer en formules mathématiques qui conduisent la simulation.

FlexSim prend en charge HIL mais on doit créer une logique pour lui-même afin de permettre la communication requise. Cela peut être fait via TCP/IP, UDP, ou en convertissant le modèle FlexSim en C++ et l'écriture la logique directement [14].

#### e. La mise en place de la plateforme simulée :

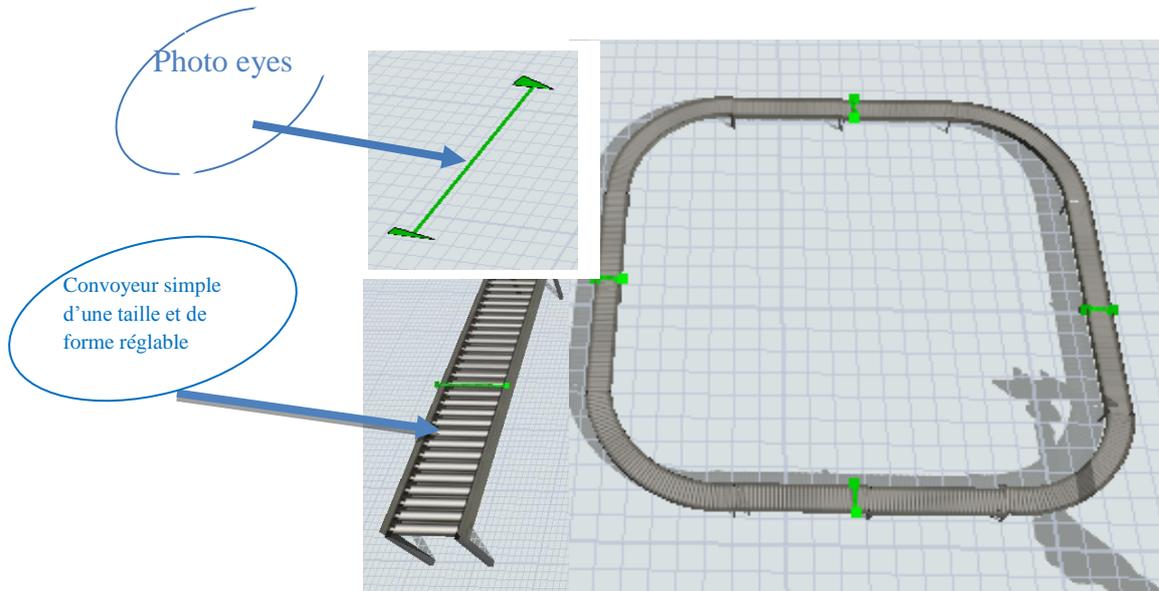
Après avoir choisi flexsim comme logiciel de simulation pour la conception de la plateforme virtuelle, on passera par la suite à sa mise en place.



**Figure II.9 : Fenêtre de propriétés de la source**

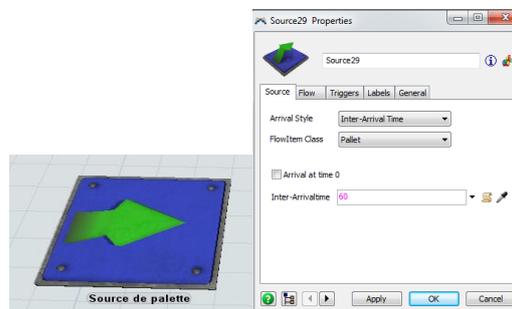
Voici les étapes suivies pour concevoir la plateforme :

- ✓ Importer les composants de la cellule de la bibliothèque du flexsim
  - Convoyeurs :  
On a formé un convoyeur en boucle en liant plusieurs convoyeurs simples, et  
On a inséré des « photo eyes » qui jouent le rôle des détecteurs photoélectriques.



**Figure II.10. Convoyeur en boucle formé de plusieurs convoyeurs simple et muni des « photo eyes »**

- Source de palettes :  
C'est Cette source qui nous donne des palettes, et elles seront directement mises sur le convoyeur, on l'a configurée de telle sorte qu'elle nous donne une palette toutes les 60 secondes.



**Figure II.11. Source de palettes et ses propriétés**

- Combineur :  
Ce composant est chargé de faire l'assemblage des pièces sur la palette, après l'avoir déposé dessus. Il est paramétré pour assembler une seule pièce sur chaque palette.

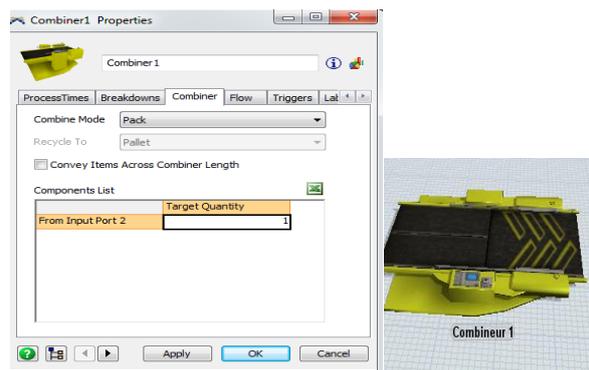


Figure II.12 :Combineur

- La queue et sa source:

Les pièces que le combineur va assembler avec la palette se pose sur cette queue après que cette source les a fournies.

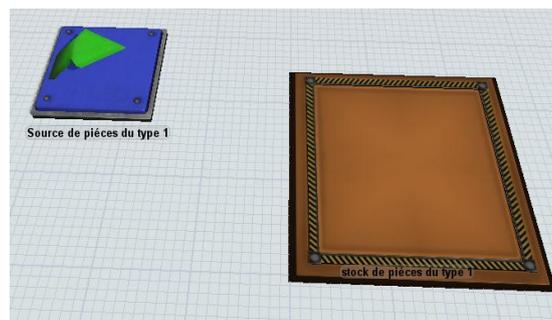


Figure II.13 : La queue des pièces du type 1 et sa source

- Robot:

Le robot qu'on va utilisé fera la tache du transport, c'est-à-dire il prendra la palette du convoyeur, puis la pièce à assembler de la queue et les mettre sur le combineur, une fois que l'assemblage est fai, il remettra la palette sur le convoyeur.

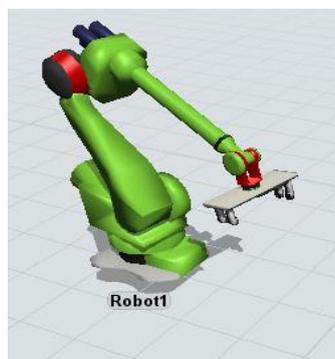
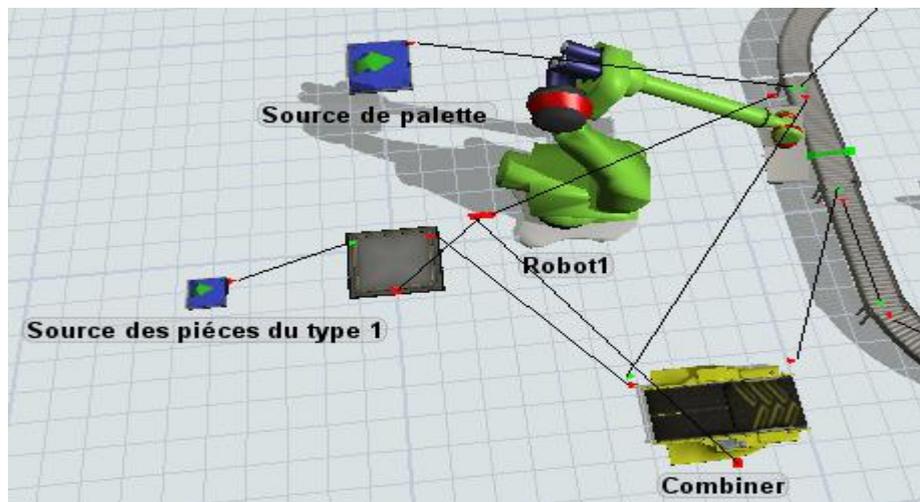


Figure II.14. Robot 1

- ✓ Créer les liaisons nécessaires entre les éléments :

C'est grâce à ces liaisons qu'on a pu former un système dont les constituants sont en interaction.



**Figure II.15. Figure du poste 1 montrant les liaisons entre l'ensemble de ses éléments composants**

La mise en place de la plateforme ne crée aucun problème vu que flexsim contient tous les constituants principaux nécessaires pour ça. Le problème auquel on fait face par la suite est la commande de cette plateforme, comment la commander, ou plutôt comment faire passer les commandes du PLC à la plateforme virtuelle et en contrepartie comment suivre l'évolution du système.

### **II.5 La communication socket :**

Comme on l'a précisé dans l'élément précédent, flexsim possède la capacité de communiquer avec l'environnement extérieur (d'autres matériels et logiciels), cela se fait grâce à son système d'interfaçage. Il a une interface appelée interface socket sur laquelle on peut créer une logique pour lui-même afin de permettre la communication requise. Mais c'est quoi un socket ?

#### **Définition :**

Un socket est un élément logiciel qui est aujourd'hui répandu dans la plupart des systèmes d'exploitation. Il s'agit d'une interface logicielle avec les services du système d'exploitation, grâce à laquelle un développeur exploitera facilement et de manière uniforme les services d'un protocole réseau.

Il lui sera ainsi par exemple aisé d'établir une session TCP, puis de recevoir et d'expédier des données grâce à elle. Cela simplifie sa tâche car cette couche logicielle, de laquelle il requiert des services en appelant des fonctions, masque le travail nécessaire de gestion du réseau, pris en charge par le système.

La communication par socket est souvent comparée aux communications humaines. On distingue ainsi deux modes de communication :

- Le mode connecté (comparable à une communication téléphonique), utilisant le protocole TCP. Dans ce mode de communication, une connexion durable est établie entre les deux processus, de telle façon que l'adresse de destination n'est pas nécessaire à chaque envoi de données ;
- Le mode non connecté (analogue à une communication par courrier), utilisant le protocole UDP. Ce mode nécessite l'adresse de destination à chaque envoi, et aucun accusé de réception n'est donné.

#### **II.6 Conclusion :**

Ce deuxième chapitre éclaircit mieux l'idée de notre projet qui consiste à créer une plateforme virtuelle sur flexsim et la commander par un PLC, et explique le fonctionnement qu'on veut établir. Dans le prochain chapitre, on s'intéressera à la partie matérielle (physique) de notre système, ça sera les PLC utilisés.

**CHAPITRE III : IDENTIFICATION DES  
COMPOSANTS DE LA CELLULE DU CDTA (coté  
hardware)**

### **III INTRODUCTION :**

Aujourd'hui, l'automatisation n'est pas une commodité, c'est un réel besoin pour les industriels. Une usine doit autant que faire se peut accroître de manière continue sa productivité. L'utilisation d'un automate conduit à une très grande rapidité, une meilleure régularité des résultats et évite à l'homme des tâches pénibles et répétitives.

Après avoir présenté globalement notre plateforme, on passe dans ce troisième chapitre à la présentation des contrôleurs qui seront utilisés pour le pilotage de cette cellule, leurs caractéristiques, et bien sûr le logiciel qui permet leur programmation.

#### **III.1 Identification les éléments de la cellule existante dans le laboratoire :**

Le laboratoire de CDTA où le projet de la réalisation de la plateforme aura lieu est doté d'un automate de la dernière gamme de SIEMENS qui est le SIMATIC S7-1500 muni de la CPU 1515-2PN, et de quatre entrées sorties (E/S) décentralisée de la famille SIMATIC ET 200 SP qui ont une fonctionnalité de CPU, ainsi qu'une IHM SIMATIC IFP2200 Touche Ext écran '22'.

Pour une bonne exploitation du matériel, on commence par sa présentation, et la détermination de ses caractéristiques.

##### **III.1.1 Présentation l'automate SIMATIC S7\_1500 :**

L'automate SIMATIC S7-1500 est un système de commande modulaire utilisé pour les moyennes et grandes performances. Il existe un éventail complet de module pour une adaptation optimisée à la tâche d'automatisation.

SIMATIC S7-1500 est un perfectionnement des systèmes d'automatisation SIMATIC S7-300 et S7-400 avec les nouvelles performances suivantes :

- Performance système accrue.
- Fonctionnalité Motion Control intégrée.
- PROFINET IO.
- Ecran intégré pour commande et diagnostic près de la machine [13] .

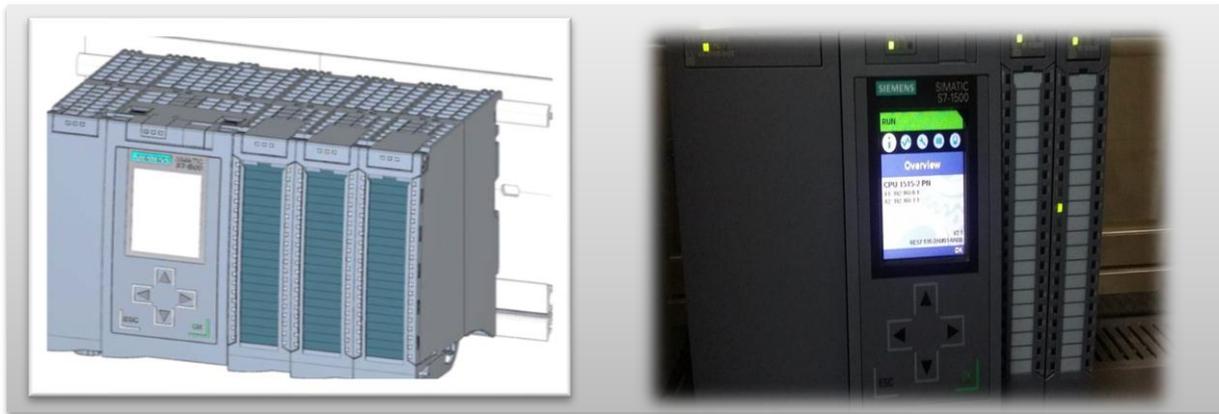


Figure III.1: Automate SIMATIC S7-1500.

L'automate S7-1500 est composé d'une alimentation électrique①, d'une CPU avec écran intégré ② et de modules d'entrées/sorties pour les signaux numériques et analogiques③. Les modules sont montés sur un profilé-support avec un rail DIN symétrique intégré④ [13].

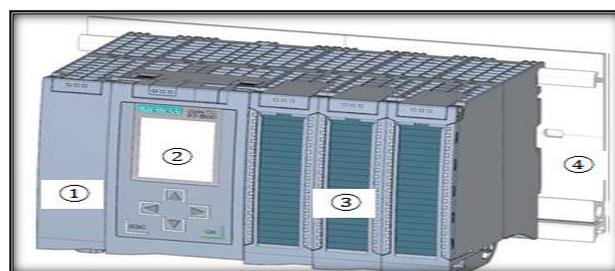


Figure III.2 : Les composants de l'automate S7-1500

**1) Unité central CPU 1515-2PN (6ES7515-2AM01-0AB0):**

La CPU 1515-2 PN est dotée d'une mémoire de programme et de données moyenne qui convient pour des applications contenant non seulement une périphérie centralisée, mais aussi des structures d'automatisation décentralisées. Elle contient deux interfaces.



Figure III.3 : CPU 1515-2PN

- L'interface PROFINET IO IRT intégrée se présente sous forme d'un commutateur à 2 ports.
- L'interface PROFINET intégrée supplémentaire avec adresse IP séparée peut être utilisée par exemple pour la séparation de réseaux, pour le raccordement d'autres périphériques PROFINET IO RT ou pour une communication rapide en tant que périphérique I. La CPU offre en outre de nombreuses fonctionnalités de régulation ainsi que la possibilité d'intégrer des entraînements via des blocs PLC-open standardisés [14].

**2) Les caractéristiques de la CPU 1515-2PN :**

La CPU 1515-2 PN dispose les caractéristiques suivantes :

- Un processeur performant : la CPU atteint des temps d'exécution de 30 ns par instruction sur bit.
- Une mémoire de travail importante : 500 ko pour le programme, 3 Mo pour les données.
- Cartes mémoire SIMATIC en tant que mémoire de chargement ; elles permettent des fonctions supplémentaires.
- Souplesse d'extension : configuration à une rangée avec max. 32 modules (CPU + 31 modules).
- Interface PROFINET IO IRT pour la connexion de la périphérie décentralisée via PROFINET.
- Interface PROFINET pour la séparation des réseaux [14].

<b>Caractéristique</b>	<b>CPU 1515-2 PN</b>
Tension d'alimentation	24V
Plage admissible ,limite inferieure	19,2 V
Plage admissible ,limite superieure	28,8 V
Consommation de courant d'entrée « valeur nominale »	2,4 A
Consommation de courant d'entrée	0,8 A
Nombre d'interfaces PROFINET	2
Nombre d'interfaces PROFIBUS	0
Puissance dissipée	6,3 W

**Tableaux III.1 : Les caractéristiques techniques de CPU 1515-2PN**

**3) L'alimentation et les modules d'entree/ sortie de l'automates S7-1500 :**

**❖ Le bloc d'alimentation :**

Le module d'alimentation assure la distribution d'énergie électrique aux différents modules. Il délivre à partir du 220 V alternatif, des sources de tension nécessaires à l'automate tels que : +5V,12Vet 24V en continu. Le bloc qu'on a utilisé est **SIPLUS S7-1500 PS 60W 24/48/60V DC**.

## ❖ Les modules d'entrées sorties :

Les modules d'entrées /sorties sont des interfaces de communication entre l'unité centrale et les différents capteurs et actionneurs, ils assurent le filtrage et l'adaptation des signaux électriques.

Pour l'instant, il n'y a que deux modules installés, ils sont les suivants :

### 1) Module d'entrées TOR : DI 32x24VDC BA

Il permet à l'automate de recevoir des informations prévenantes de la part des capteurs TOR (tout ou rien)

### 2) Module de sortie numérique : DQ 32x24VDC/0.5A ST

Les modules de sorties TOR assure le raccordement de l'automate aux différents actionneurs et pré-actionneurs tels que (moteurs, relais).



Figure III.4 : Constitution de l'API S7-1500

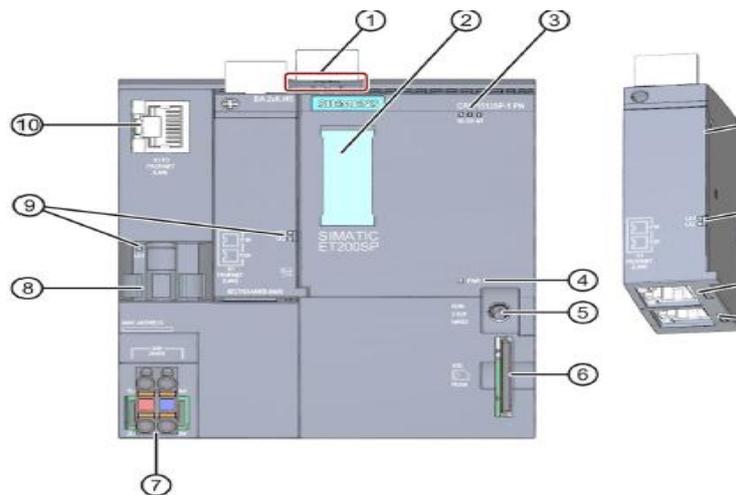
## III.1.2 Présentation d'ET 200 SP :

C'est un système d'E/S décentralisé qui permet un raccordement facile à un automate central via la liaison PROFINET.

Le ET 200 SP installé au laboratoire est composé des éléments suivants :

### 1) Unité central CPU 1512-SP1PN (6ES7512-1DK01-0AB0) :

La CPU 1512SP-1 PN est doté d'une mémoire de programme et de données possède une interface PRFINET (X1) cette interface a trois ports. Le port 3 se trouve sur la CPU. Le port 1 et le port 2 se trouvent sur le BusAdapter (Adaptateur de bus), on peut aussi raccorder ce système d'E/S à une PG / un PC ou un appareil HMI via le port 3 [15].



**Figure III.6 : Vue de face de la CPU 1512SP-1 PN avec BusAdapter**

- ① Déverrouillage du profilé support
- ② Bandes de repérage
- ③ LED d'affichage d'état et de défauts
- ④ LED d'affichage de la tension d'alimentation
- ⑤ Sélecteur de mode de fonctionnement
- ⑥ Logement pour la carte mémoire SIMATIC
- ⑦ Raccordement pour tension d'alimentation
- ⑧ Support de câble et fixation pour le port P3 de l'interface PROFINET
- ⑨ LED pour affichages d'état de l'interface PROFINET : LK1 et LK2 sur BusAdapter, LK3 sur CPU
- ⑩ Port P3 de l'interface PROFINET.

## **2) Les caractéristiques de la CPU 1512-SP1 PN :**

Le SIMATIC ET 200SP comprend généralement :

- un module d'interface qui communique avec tous les automates conformes au standard PROFINET.
- jusqu'à 64 modules d'E/S, enfichés dans des unités de base passives dans n'importe quelle combinaison
- Un module serveur qui complète la structure du SIMATIC ET 200SP [16].

Caractéristique	CPU 1512-SP1PN
Tension d'alimentation	24 VCC
Plage admissible ,limite inferieure	19.2 V
Plage admissible ,limite superieure	28.8 V
Consommation de courant d'entrée	0.6 A
Consommation de courant d'entrée « valeur nominale »	4.7 A
Nombre d'interfaces PROFINET	1
Puissance dissipée	5.6 W

**Tableau III.2: les caractéristiques de la CPU**

**3) Les modules d'entree/ sortie de l'automates ET 200 SP et son alimentation :**

- ❖ **Le module d'entree numerique : DI 16x24VDC ST**
- ❖ **Le module de sortie numerique : DQ 16x24VDC/0.5A ST**
- ❖ **Module d'alimentation : BA 2XRJ45**

**Remarque :**

Pour l'alimentation performante pour l'installation, une entrée d'alimentation stabilisée est ajoutée à l'armoire, il s'agit de SITOP PSU100S (DC 24V/10A)

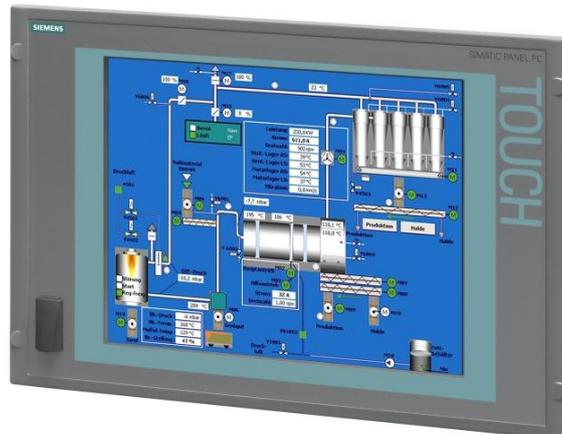


**Figure III.7 : L'alimentation PSU100S**

**III.1.3 Une IHM SIEMENS IFP2200 Touche Ext écran "22" :**

L'écran plat industriel SIMATIC IFP2200 est parfaitement adapté en tant que moniteur industriel avec un temps de réponse rapide pour la mise à jour d'image en temps réel, par exemple pour le mode travail ou l'affichage des tendances.

Les SIMATIC IFP sont connectés via la nouvelle interface DisplayPort ou DVI-D au PC industriel, qui peut être éloigné de 30 mètres. [17]



**Figure III.8 : Vue d'écran IFP2200**

### 1) Caractéristiques du SIMATIC IFP2200 (Industrial Flat Panel) :

- Robuste façade en aluminium
- Ecran TFT lumineux avec grand angle de vision ; Ecran TFT "22"
- Peut être livré comme moniteur ou pupitre tactile
- Installation déportée possible jusqu'à 5 m de l'IPC
- Interface DVI-D et interface DisplayPort V1.1
- Prise en charge du multi-moniteur
- Rétroéclairage réglable par logiciel
- Alimentation 24 V CC
- Degré de protection IP65/en face avant uniquement et type 4X/type 12 (utilisation exclusivement en intérieur) à l'état monté
- Jusqu'à 16 millions de couleurs. [17]

### III.2 Logiciel Totally Integrated Automation Portal "TIA Portal V14" :

La plateforme Totally Integrated Automation Portal est le nouvel environnement de travail Siemens qui permet de mettre en œuvre des solutions d'automatisation avec un système d'ingénierie intégré comprenant les logiciels SIMATIC STEP 7 et SIMATIC WinCC (dans la version du programme disponible) [6].

Le logiciel STEP 7 Professional (TIA Portal) est l'outil de programmation des automates -SIMATIC S7-1500, S7-1200, S7-300, S7-400

Avec STEP 7 Professional, les fonctions suivantes peuvent être utilisées pour automatiser une installation :

- Configuration et paramétrage du matériel
- Paramétrage de la communication
- Programmation
- Test, mise en service et dépannage avec les fonctions d'exploitation et de diagnostic
- Documentation
- Génération d'écrans de visualisation pour les Basic Panels SIMATIC avec WinCC Basic intégré.
- Il est également possible de générer des écrans de visualisation pour les PC et autres Panels à l'aide d'autres progiciels WinCC [16].

### III.2.1 Vue du portail et vue du projet :

Lorsque l'on lance TIA Portal, l'environnement de travail se décompose en deux types de vue :

- **La vue du portail :** elle est axée sur les tâches à exécuter et sa prise en main est très rapide.
- **La vue du projet :** elle comporte une arborescence avec les différents éléments du projet. Les éditeurs requis s'ouvrent en fonction des tâches à réaliser. Données, paramètres et éditeurs peuvent être visualisés dans une seule et même vue [18].

#### III.2.1.1 Vue du portail :

Chaque portail permet de traiter une catégorie de tâche (actions), la fenêtre affiche la liste des actions pouvant être réalisées pour la tâche sélectionnée, la figure ci-dessous représente une vue du portail [18].

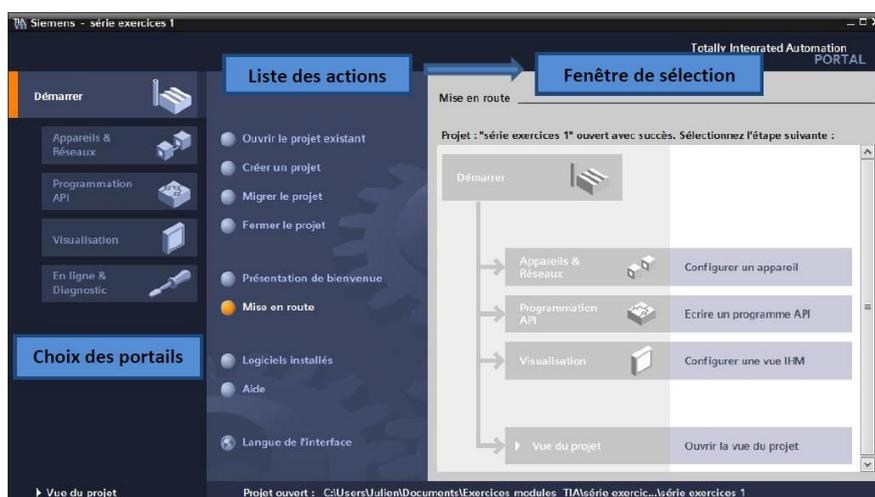


Figure III.9 : Vue de portails

## III.2.1.2 Vue du projet :

L'élément « Projet » contient l'ensemble des éléments et des données nécessaires pour mettre en œuvre la solution d'automatisation souhaitée, la figure ci-dessous représente la vue du projet [18].

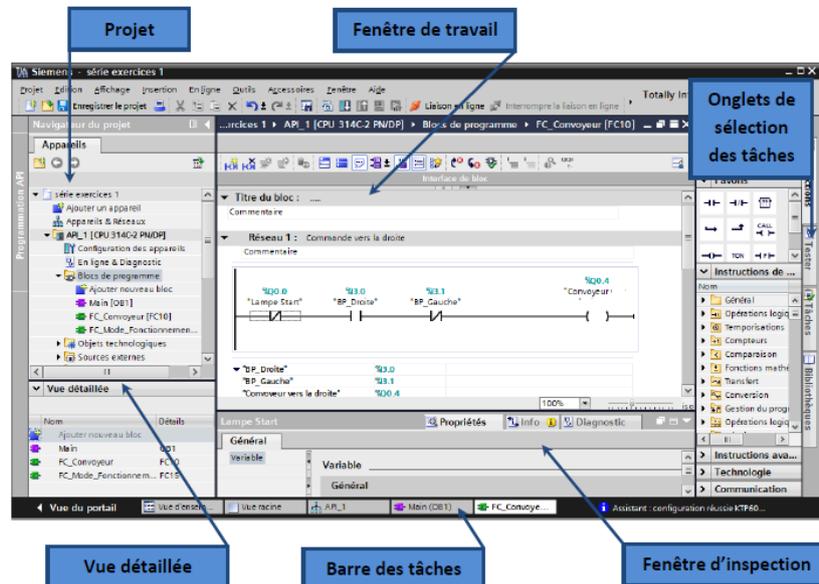


Figure III.10 : Vue du projet

- ❖ **La fenêtre de travail** permet de visualiser les objets sélectionnés dans le projet pour être traités. Il peut s'agir des composants matériels, des blocs de programme, des tables des variables, des interfaces homme machine (IHM).
- ❖ **La fenêtre d'inspection** permet de visualiser des informations complémentaires sur un objet sélectionné ou sur les actions en cours d'exécution (propriété du matériel sélectionné, message d'erreur lors de la compilation des blocs de programme,...).
- ❖ **Les onglets de sélection de tâches** ont un contenu qui varie en fonction de l'objet sélectionné (configuration matérielle → bibliothèques des composants, bloc de programme → instructions de programmation).

Cet environnement de travail contient énormément de données. Il est possible de masquer ou réduire certaines de ces fenêtres lorsque l'on ne les utilise pas.

Il est également possible de redimensionner, réorganiser, désancrer les différentes fenêtres [18].

## III.2.2 Création d'un projet et configuration d'une station de travail

Pour créer un projet dans la vue du portail, il faut sélectionner l'action «Créer un projet ».

On peut donner un nom au projet, choisir un chemin où il sera enregistré, indiquer un commentaire ou encore définir l'auteur du projet. Une fois que ces informations sont entrées, il suffit de cliquer sur le bouton « créer », la figure ci-dessous représente la création d'un projet [18].

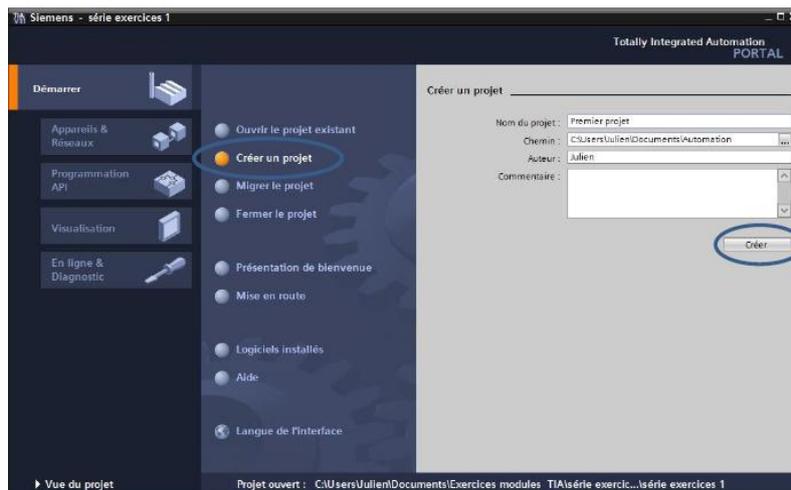


Figure II.11 : Création d'un projet

### III.2.3 Configuration et paramétrage du matériel

Une fois votre projet créé, on peut configurer la station de travail. La première étape consiste à définir le matériel existant. Pour cela, on peut passer par la « **vue du projet** » et cliquer sur « **ajouter un appareil** » dans le navigateur du projet.

La liste des éléments que l'on peut ajouter apparaît (API, IHM, système PC). On commencera par faire le choix de notre CPU pour ensuite venir ajouter les modules complémentaires (alimentation, E/S TOR ou analogiques, module de communication.....Etc.), La figure ci-dessous représente la configuration et le paramétrage du matériel [18].

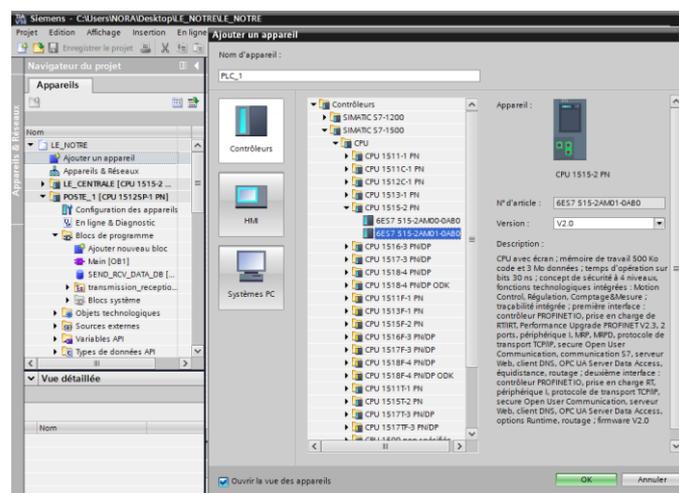


Figure III.12 : Configuration et paramétrage du matériel

Les modules complémentaires de l'API peuvent être ajoutés en utilisant le catalogue. Si l'on veut ajouter un écran ou un autre API, il faut repasser par la commande « ajouter un appareil » dans le navigateur du projet. Lorsque l'on sélectionne un élément à insérer dans le projet, une description

est proposée dans l'onglet information. La figure ci-dessous est une deuxième représentation de la configuration et du paramétrage du matériel [18].

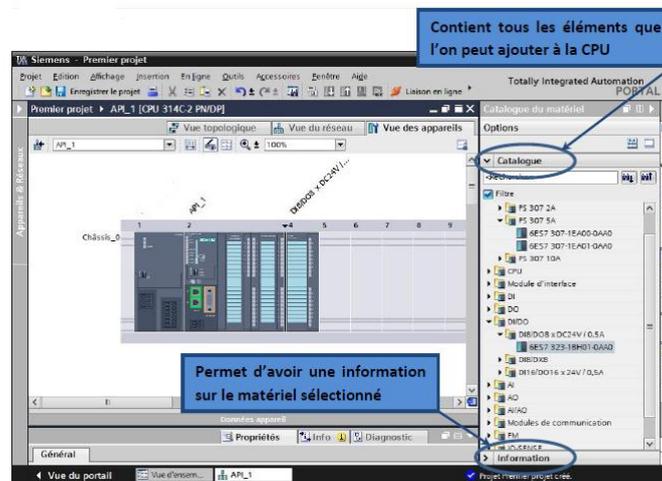


Figure III.13 : Configuration et paramétrage du matériel

### III.2.4 Adressage des E/S :

Pour connaître l'adressage des entrées et sorties présentes dans la configuration matériel, il suffit d'aller dans « **appareil et réseau** » dans le navigateur du projet, dans la fenêtre de travail, on doit s'assurer d'être dans l'onglet « **Vue des appareils** », de sélectionner l'appareil voulu, la figure ci-dessous est une représentation des adressages des Entrée / Sortie [18].

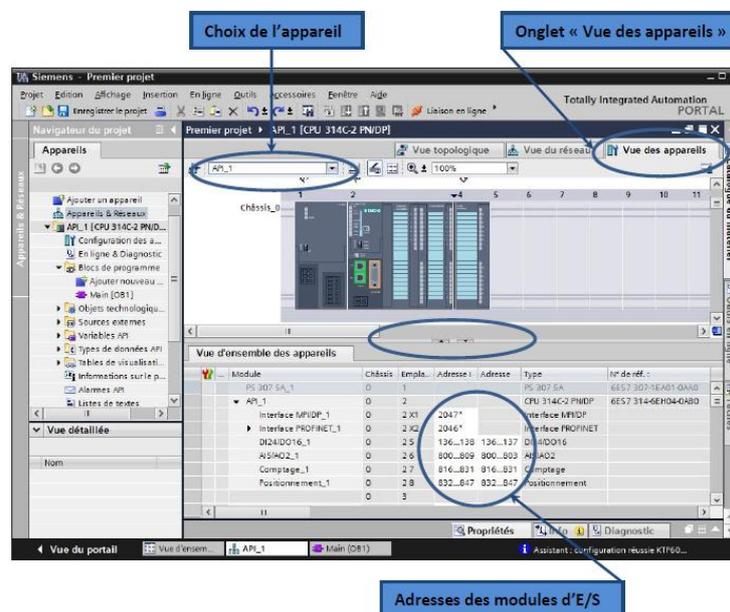


Figure III.14 : Adressage des E/S.

On sélectionne la CPU puis à l'aide des deux petites flèches, on fait apparaître l'onglet « **Vue d'ensemble des appareils** ». Les adresses des entrées et sorties apparaissent.

Vous pouvez les modifier en entrant une nouvelle valeur dans la case correspondante.

### III.2.5 WinCC sur TIA portal :

WinCC (TIA portal) est un logiciel d'ingénierie pour la configuration de pupitres SIMATIC, de PC industriel SIMATIC et de PC standard par le logiciel de visualisation. Le SIMATIC WinCC dans le TIA portal fait partie d'un nouveau concept d'ingénierie intégré qui offre un environnement d'ingénierie homogène pour la programmation et la configuration de solution de commande, de visualisation d'entraînement, c'est le logiciel pour toutes les applications IHM allant de solution de commande simple avec basic panel aux applications SCADA pour système multipostes basé sur PC [18].

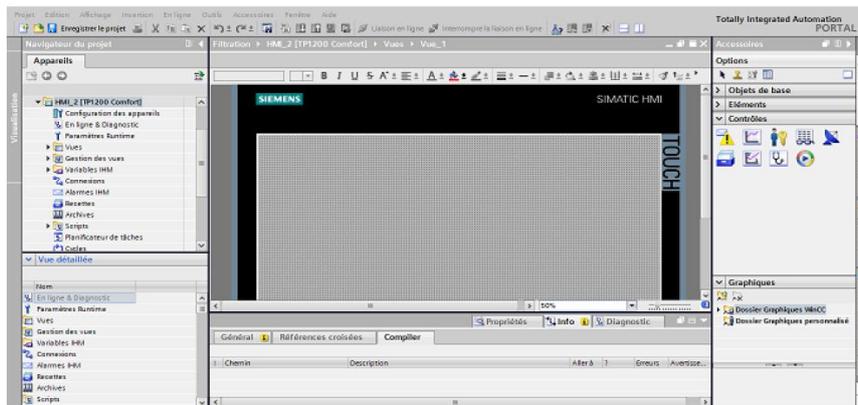


Figure III.15 : Vue SIMATIC HMI

### III.2.6 Les blocs de programme :

L'automate met à disposition différents types de blocs qui contiennent le programme et les données correspondantes. Selon les exigences et la complexité du processus, il est possible de structurer le programme en différents blocs : OB, FB et FC [19].

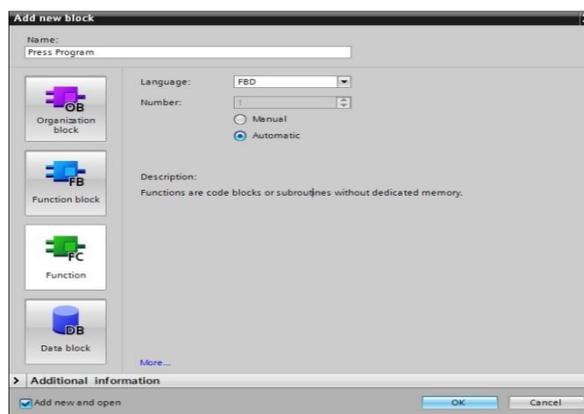


Figure III.16 : Création d'un nouveau bloc

Il existe différents types de blocs pour exécuter les tâches dans un système d'automatisation.

#### 1) Les blocs d'organisation (OB) :



Ils constituent l'interface entre le système d'exploitation et le programme utilisateur. Les OB sont programmables par l'utilisateur, ce qui permet de déterminer le comportement de la CPU.

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

- Comportement au démarrage
- Exécution cyclique du programme
- Exécution du programme déclenchée par des alarmes (cyclique, processus, diagnostic,...)
- Traitement des erreurs.

Pour que le traitement du programme démarre, le projet doit posséder au moins un OB cyclique (par exemple l'OB 1) [19].

### 2) Les blocs de fonctions (FC) :



Ce sont des blocs de code sans mémoire. Elles n'ont pas de mémoire de données dans laquelle il est possible d'enregistrer les valeurs de paramètres de bloc.

Les données des variables temporaires sont perdues après l'exécution de la fonction.

Si on veut mémoriser ces données, il faut utiliser des opérands globaux [19].

### 3) Les blocs fonctionnels (FB):



Ce sont des blocs de code qui mémorisent durablement leurs paramètres d'entrée, de sortie et d'entrée/sortie dans des blocs de données d'instance afin qu'il soit possible d'y accéder même après le traitement de blocs [19].

### 4) Blocs de données d'instance (DB):

L'appel d'un bloc fonctionnel est une instance. Les données avec lesquelles opère l'instance sont mémorisées dans un bloc de données d'instance.

La taille maximale des blocs de données d'instance varie selon la CPU. Les variables déclarées dans le bloc fonctionnel déterminent la structure du bloc de données d'instance [19].

## III.2.7 Les langages de programmation :

Un ensemble homogène d'instructions dans tous les langages de programmation (CONT, LOG, LIST, SCL et GRAPH).

SIMATIC STEP 7 V14 met à disposition de puissants éditeurs pour la programmation des automates SIMATIC S7.

### a) CONT et LOG – Langages de programmation graphiques :

Les langages de programmation graphique sont supportés par de nouveaux compilateurs à hautes performances. Les puissants outils et fonctionnalités intégrées, comme, par exemple, la programmation indirecte, accroissent l’efficacité du développement des programmes.

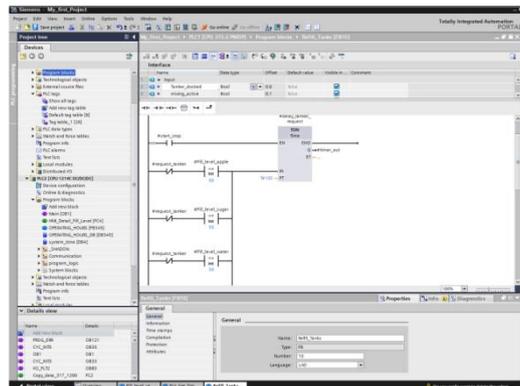


Figure III.17 : langage CONT et LOG

### b) SCL (Structured Control Language) – Programmation d’algorithmes complexes :

Le Structured Control Language (SCL) correspond au langage textuel de haut niveau ST (Structured Text), il convient notamment à la programmation rapide d’algorithmes complexes et de fonctions mathématiques. Le code SCL est plus simple, plus court et plus clair, ce qui en facilite et en accélère l’écriture et la manipulation. Dans la TIA V14, un nouveau compilateur SCL à hautes performances accroît l’efficacité de la programmation.

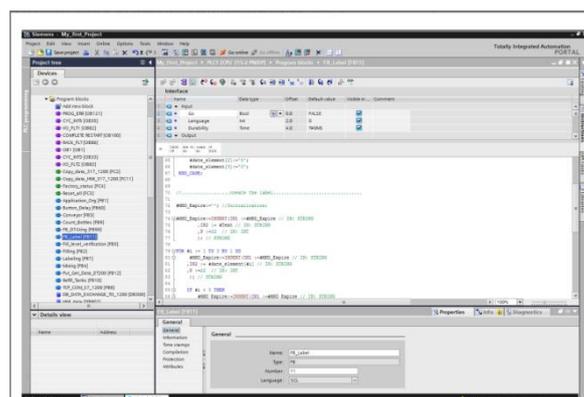
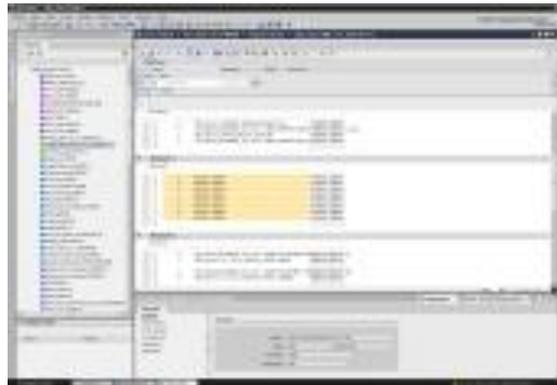


Figure III.18 : langage SCL

### c) LIST – Liste d’instructions :

Le langage de programmation textuelle LIST permet de créer des programmes d’application à un niveau proche du matériel et en optimisant le temps d’exécution et la place en mémoire.

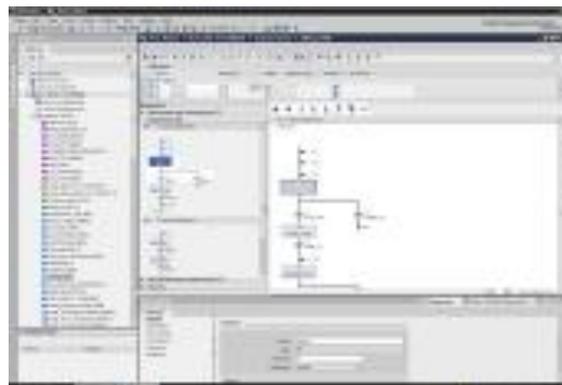


**Figure III.19 : Langage LIST**

**d) GRAPH (SFC) - Programmation de chaînes séquentielles :**

Le langage Sequential Function Chart (SFC) s'utilise pour la description de procédures séquentielles à séquences alternatives ou parallèles.

Les procédures se configurent et se programment clairement et rapidement dans un mode de visualisation standardisé.

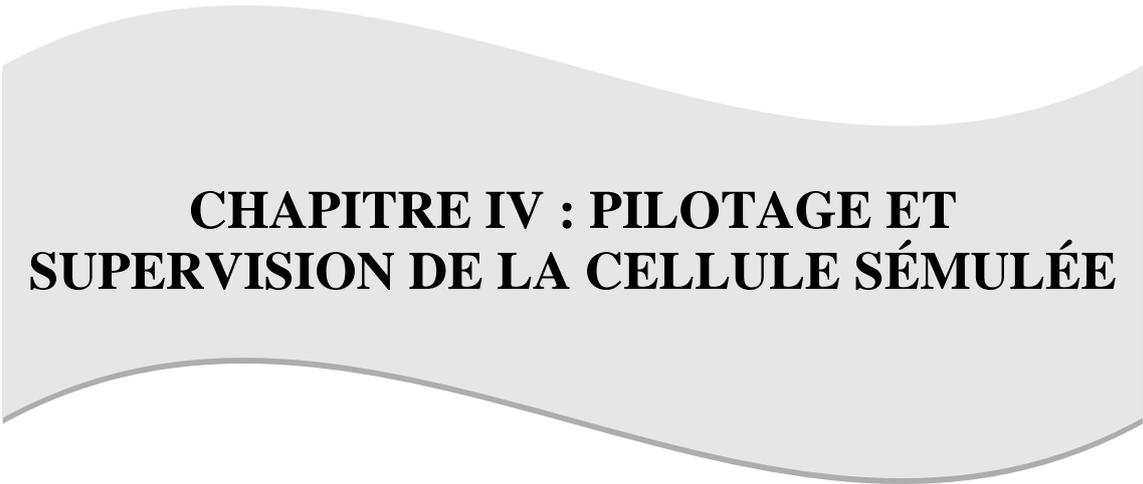


**Figure III.20: Langage SFC**

**Remarque :** Dans notre projet on a travaillé avec deux langages de programmation LADDER et le SCL pour certains blocs.

**III.3 Conclusion :**

Ce chapitre on a présenté les éléments de notre cellule et un bref exposé sur le logiciel Totally Integrated Automation Portal (TIA Portal V14). Dans le prochain chapitre, on donnera l'architecture de pilotage de ces éléments et on va bien éclaircir leur fonctionnement.



**CHAPITRE IV : PILOTAGE ET  
SUPERVISION DE LA CELLULE SÉMULÉE**

**IV Introduction :**

Dans ce qui précède, on a parlé de la plateforme virtuelle, de son fonctionnement, et des contrôleurs qui feront sa commande, mais sans préciser la manière de lier tous ces éléments pour avoir un système automatisé.

Dans ce chapitre, on commencera par la présentation de l'architecture du pilotage, puis définir les différents types de communication exploitées, pour passer par la suite à la programmation et le pilotage de la cellule. Et on terminera par la présentation du système superviseur de la plateforme.

**IV.1 Architecture de pilotage de la cellule et système de communication :**

Avant d'entamer la programmation de la cellule, on décrit d'abord la partie commande de ce système, elle contiendra un automate S7\_1500 et quatre ET 200 SP comme on l'a précisé dans le chapitre précédent.

On va adopter une architecture de pilotage centralisée, tel que l'automate S7\_1500 sera l'automate central sur lequel on établira le programme complet qui commandera la cellule, mais on va utiliser les systèmes d'E/S (ET 200 SP) décentralisées avec leurs fonctionnalités de CPU pour des raisons qu'on expliquera dans ce qui suit.

Pour connecter les quatre ET 200 SP à l'automate central, on va les mettre sous un réseau PROFINET.

**IV.1.1 Réseau PROFINET :**

SIEMENS adopte PROFINET comme une technologie de communication pour connecter les différents appareils d'un réseau.

**a. Définition de PROFINET :**

PROFINET est un réseau de haut niveau utilisé pour les applications d'automatisation industrielle. Il est basé sur des technologies Ethernet standard. Il utilise un matériel et des logiciels Ethernet traditionnels pour définir un réseau qui structure la tâche d'échanger des données des alarmes et des diagnostics avec des contrôleurs programmables et d'autres contrôleurs d'automatisation [20].

**Ethernet :**

Ethernet est la technologie de réseau local la plus largement installée. C'est un protocole de couche liaison dans la pile TCP / IP , décrivant comment les périphériques en réseau peuvent formater les données pour la transmission à d'autres périphériques réseau sur le même segment

réseau et comment mettre ces données sur la connexion réseau d'erreurs pour détecter les problèmes de transmission [21].

**TCP / IP :**

TCP/IP est un sigle qui recouvre deux protocoles utilisés par de nombreuses sociétés commercialisant des équipements de réseau. Ces deux protocoles sont IP (Internet Protocol) et TCP (Transmission Control Protocol) forment respectivement la couche réseau et la couche transport qui ont été développées pour les besoins d'interconnexion des divers réseaux hétérogènes. L'idée de base est de rendre ces réseaux homogènes en leur imposant un protocole commun qui est le TCP. ET pour passer d'un sous-réseau à un autre, il faut passer par le protocole IP qui gère le routage [22].

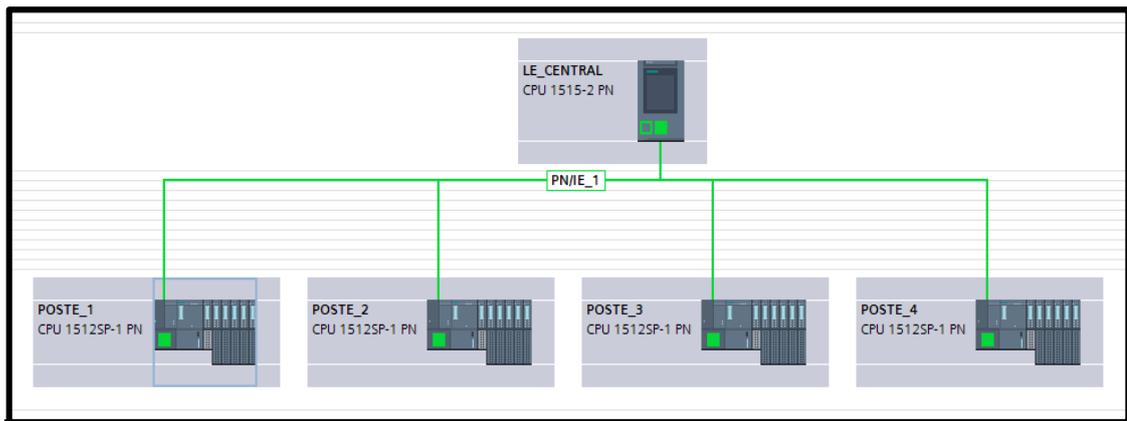
**b. Spécification sur le PROFINET IO :**

Les dernières gammes des PLC SIEMENS possèdent des interfaces PROFINET IO qui représente une couche d'application Ethernet industrielle unique. Il offre de nombreux avantages par rapport à PROFIBUS qui est le réseau utilisé par les gammes S7\_300 et S7-400, on cite certains de ces avantages :

- Fonctionnement à grande vitesse : le canal de communication en temps réel fournit un échange des messages à haute vitesse en évitant le temps requis pour traiter la pile TCP / IP.
- Prise en charge des applications de contrôle de mouvement à temps critique.
- Démarrage court.
- Facilité d'installation.

PROFINET IO utilise trois canaux de communication différents pour échanger des données avec des automates programmables et d'autres appareils :

- Le canal TCP / IP standard est utilisé pour le paramétrage, la configuration et les opérations de lecture / écriture acycliques.
- Le canal RT ou Real Time est utilisé pour le transfert de données cyclique standard et les alarmes, cette communication contourne l'interface TCP / IP standard pour accélérer l'échange de données avec les automates programmables.
- Le canal, Isochronous Real Time (IRT), ou temps réel isochrone, c'est un canal à très haut débit utilisé pour les applications Motion Control [23].



**Figure IV.1 : Schéma représentant la connexion des différents éléments de la partie commande sous un réseau PROFINET**

**Remarque :**

Pour connecter ces appareils sous le même réseau, il suffit de configurer ce réseau dans la vue de réseau de logiciel TIA PORTAL, et puis lier chacun d’eux avec un câble Ethernet à un switch qui est un matériel d’interconnexion de type concentrateur, on dispose d’un au laboratoire, c’est un switch TP-LINK à huit ports.



**Figure IV.2 : switch TP-LINK**

Maintenant qu’on a expliqué la mise en réseau des éléments physiques de la partie commande, on doit expliquer comment la lier à la plateforme virtuelle, donc on revient sur la communication socket qu’on a défini dans le chapitre 2. On donnera les détails de la création des interfaces socket tout en expliquant comment que l’échange de données se déroulera entre la partie physique (contrôleurs) et la partie cyber (la plateforme virtuelle).

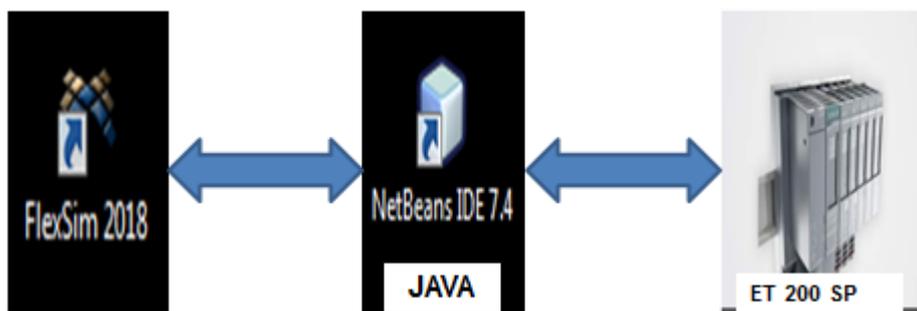
## IV.1.2 Communication SOCKET :

Notre but c'est de faire communiquer la partie commande avec la plateforme virtuelle conçu sur Flexsim via une communication socket du type client/serveur.

Donc deux programmes sont nécessaires, un sur Flexsim l'autre sur le PLC (ET 200SP), du telle sorte qu'un sera le serveur et l'autre client, c'est le client qui demandera la connexion en premier, et puis par la suite l'échange de données sera bidirectionnel.

On rappelle qu'on a quatre postes de travail sur la plateforme virtuelle qui sont semblables (ils effectuent la même tâche), donc on pourra programmer un seul poste, et le copier sur les trois autres. Mais pour différencier entre la provenance et la destination des données, on doit créer quatre interfaces sur Flexsim, une interface pour chaque poste, et quatre interfaces sur la partie commande. C'est là qu'on va utiliser la fonctionnalité CPU des quatre ET 200 pour créer dessus des interfaces socket, chacun des ET 200 SP est spécifié pour un seul poste.

Mais à cause des difficultés rencontrées dans l'établissement d'une communication socket directe entre les interfaces de Flexsim et celles des ET 200 SP, on était obligé d'utiliser un programme JAVA comme une passerelle entre eux.



Sur JAVA, on crée des serveurs, quatre se connectent avec les ET 200 SP, quatre se connectent avec les interfaces de flexsim.

Pour éclaircir le déroulement de l'échange de données entre flexsim et la partie commande, on explique l'échange de données entre une interface de flexsim (qui correspond à un seul poste de travail) avec un ET 200SP.

On appelle serveur 1 le serveur qui se connecte avec le premier ET 200 SP, et serveur 2 le serveur qui connecte avec l'interface flexsim créée pour le premier poste. Les données (les entrées du système) venantes de premier poste de flexsim sont reçues par le serveur 2 et puis transférées vers le serveur 1 pour les transmettre à l'ET 200. Et celles qui viennent de l'ET 200 (ce sont des commandes), seront reçues par le serveur 1 et transférées vers le serveur 2 pour qu'il les transmise à flexsim.

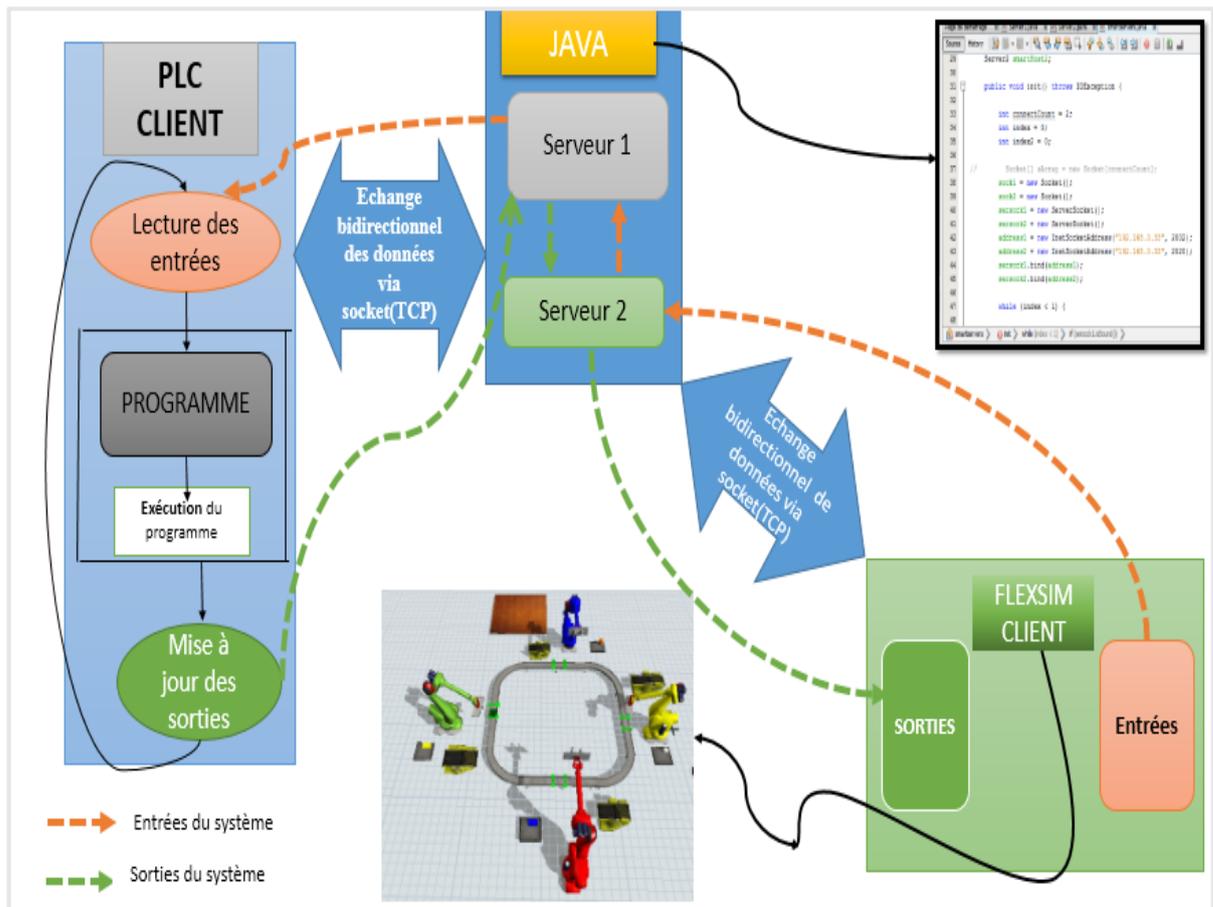


Figure IV.3 : Schéma explicatif de déroulement de l'échange de données entre la partie commande et la plateforme virtuelle

#### IV.1.2.1 La création d'une interface socket sur flexsim :

La bibliothèque de commande de flexsim possède les instructions qu'il faut pour la création d'un socket. Il suffit de créer un script sur lequel on le programme.

Les instructions qu'on a utilisées pour l'établissement de cette communication socket sont les suivantes :

**Socketinit** : Cette commande initialise les processus d'arrière-plan qui permettent l'utilisation de Windows Sockets. Elle doit être appelée avant que les appels aux commandes socket fonctionnent.

**Clientcreate** : Cette commande crée un nouveau socket client configuré pour communiquer en utilisant le protocole TCP / IP. FlexSim peut créer jusqu'à cinquante sockets client en même temps. Chaque client peut être connecté à un serveur différent.

**Clientconnecte** : Dans les paramètres de cette instruction on doit déterminer le port et l'adresse IP du serveur auquel il est sensé se connecté, quand le programme s'exécute, cette commande tente

de connecter par socket à un serveur s'exécutant sur la machine qui a l'adresse IP défini dans ses paramètres, qui écoute sur le port spécifié.

**Clientsend** : Cette commande envoie un message sur le socket spécifié.

**Clientreceive** : Cette commande reçoit les données qui ont été envoyées au socket spécifié.

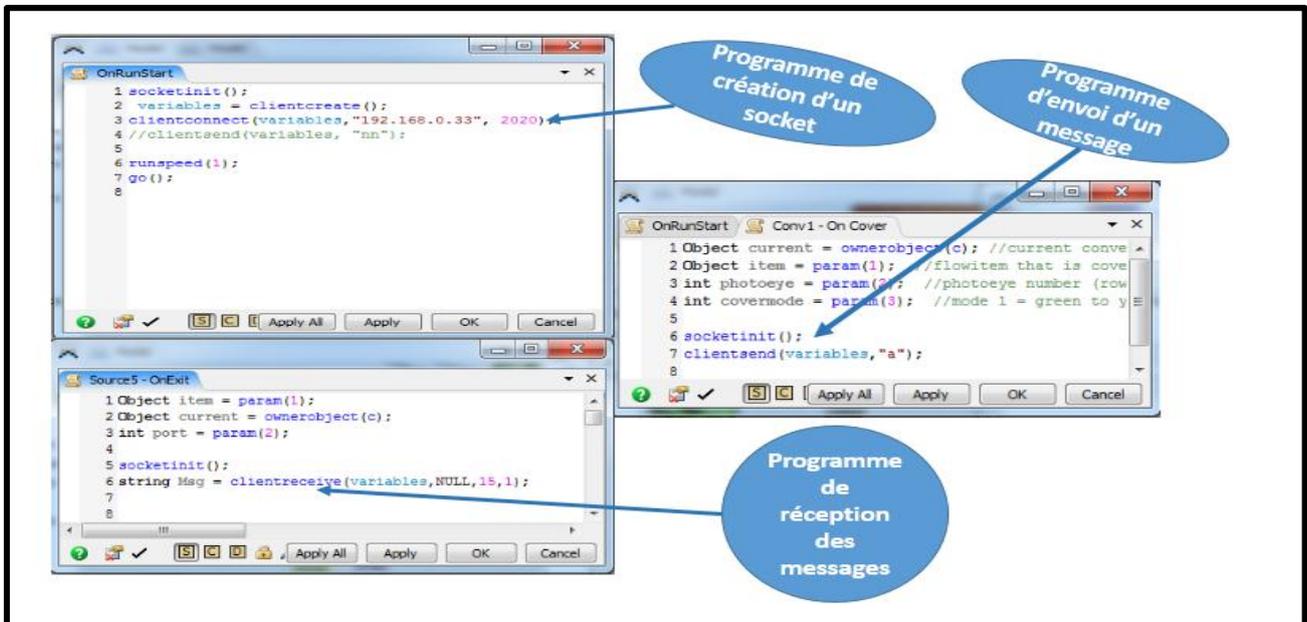


Figure IV.4 : programmes faits sur flexsim

#### IV.1.2.2 La création d'une interface socket sur l'ET 200 SP:

L'ET 200 SP possède la capacité d'établir une communication socket et échanger des données avec d'autres appareils en utilisant des instructions existantes sur la bibliothèque de communication du logiciel TIA portale.

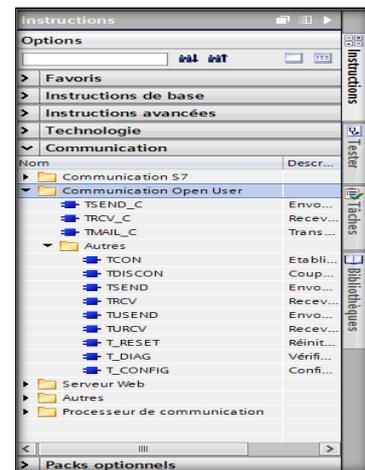


Figure IV.5 : La bibliothèque communication

Les instructions qu'on a utilisées pour l'établissement de cette communication sont les suivantes :

- **TCON** : pour l'établissement de la liaison

L'instruction "TCON" permet de définir et d'établir une liaison de communication. Une fois définie et établie, la liaison est maintenue et surveillée automatiquement par la CPU. L'instruction "TCON" est exécutée de manière asynchrone [19].

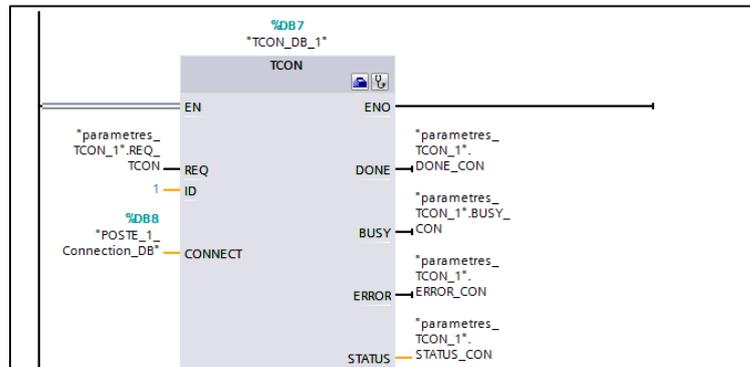


Figure IV.6 : bloc d'instance de l'instruction TCON

La figure ci-dessous représente paramètre de liaison de TCON, là où on précise l'adresse IP et le port de l'appareil partenaire :

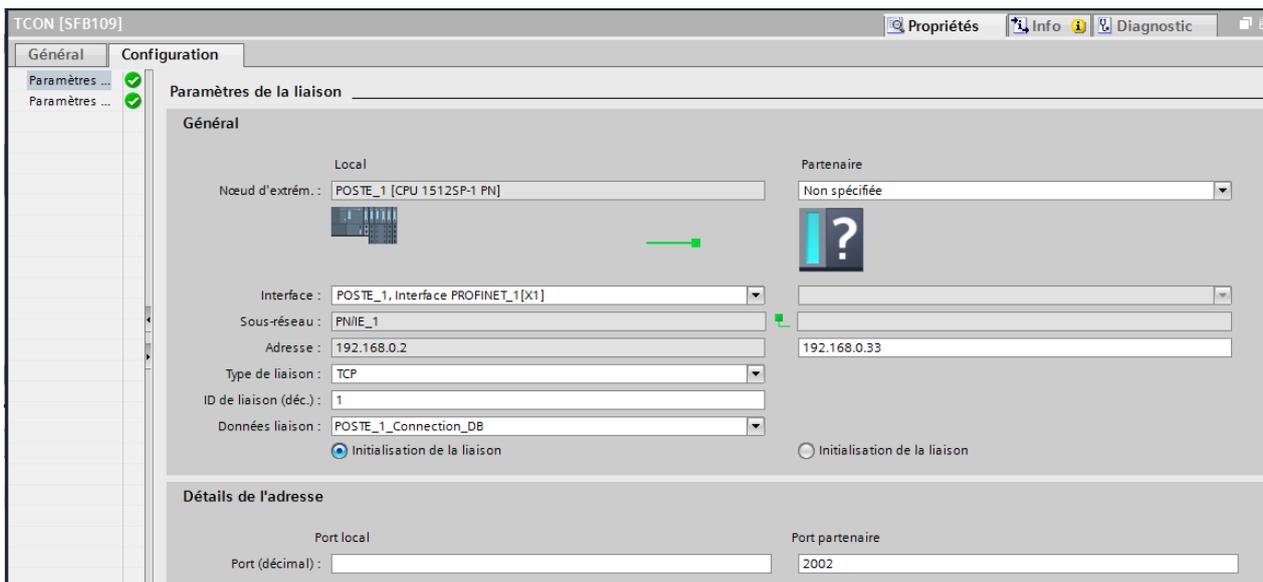


Figure IV.7: Paramétrage de l'instruction de TCON.

Dans notre cas, l'appareil partenaire est le PC qui contient le programme JAVA, donc on lui donne son adresse IP, et le port du serveur avec qui il est sensé se connecter.

- **TRCV pour la réception des données.**

L'instruction "TRCV" permet la réception de données via une liaison de communication existante. Et elle s'exécute de manière asynchrone [19].

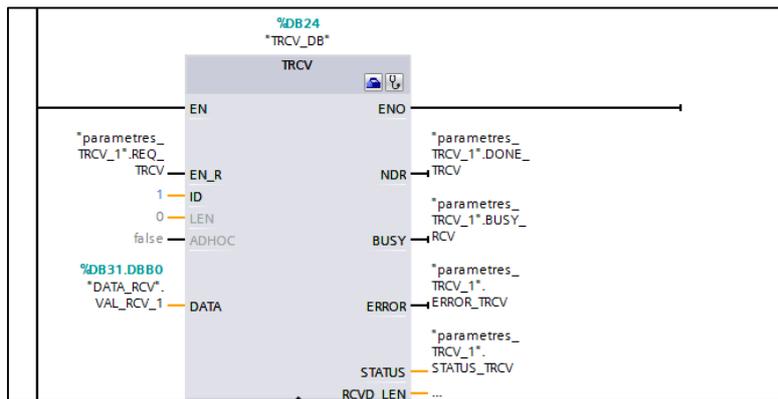


Figure IV.8: bloc d'instance de l'instruction TRCV.

Les paramètres les plus importants à configurer pour cette instruction sont :

**ID** : Référence à la liaison établie avec "TCON".

**DATA** : Pointeur sur la zone de réception, les structures doivent être identiques du côté émission et du côté réception. Dans notre cas, on va Recevoir des données du type char

- **TSEND** : pour l'émission des données

L'instruction "TSEND" permet l'envoi de données via une liaison de communication existante. Et elle s'exécute de manière asynchrone [19].

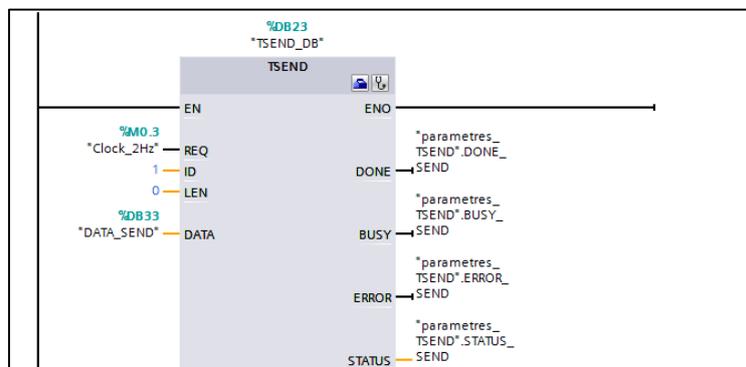


Figure IV.9: L'instruction TSEND qui permet l'envoi de données

Les paramètres les plus importants à configurer pour cette instruction sont :

**ID** : Référence à la liaison établie avec "TCON".

**DATA** : Pointeur sur la zone d'émission qui contient l'adresse et la longueur des données à envoyer.

- **TDICON** : pour la fermeture de la liaison

L'instruction "TDISCON" interrompt une liaison de communication de la CPU à un partenaire de liaison [19].

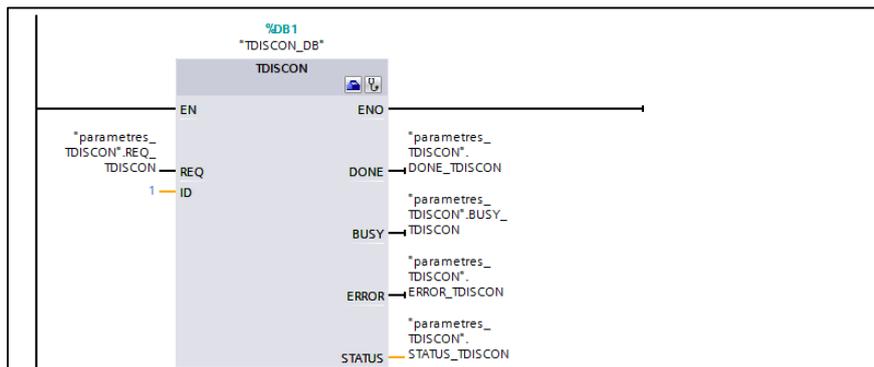


Figure IV.10 : bloc d'instance de l'instruction de déconnexion

Au tout début de cet élément, on a précisé que le programme de commande est fait sur l'automate central (S7\_1500), donc il doit avoir accès aux données que les ET 200 SP reçoivent de la part de flaxsim, tout comme il doit leurs transférer les ordres qu'ils doivent transmettre à la plateforme virtuelle. Pour cela, on utilise une communication appelée la communication S7.

#### IV.1.3 Communication S7 :

S7comm (S7 Communication) est un mode de communication propriétaire de Siemens. Il permet de faire communiquer des automates Siemens de la famille S7-300 / 400/1200/1500, et les ET 200 qui possèdent la fonctionnalité CPU.

Les deux instructions qui nous ont servi sont les suivantes :

- **GET** : est une instruction qui permet de lire des données dans une CPU distant. ses paramètres principaux :

**ADDR\_1** : c'est pointeur désignant les zones à lire dans la CPU partenaire.

**RD\_1** : c'est le pointeur désignant les zones de la propre CPU où stocker les données lues.

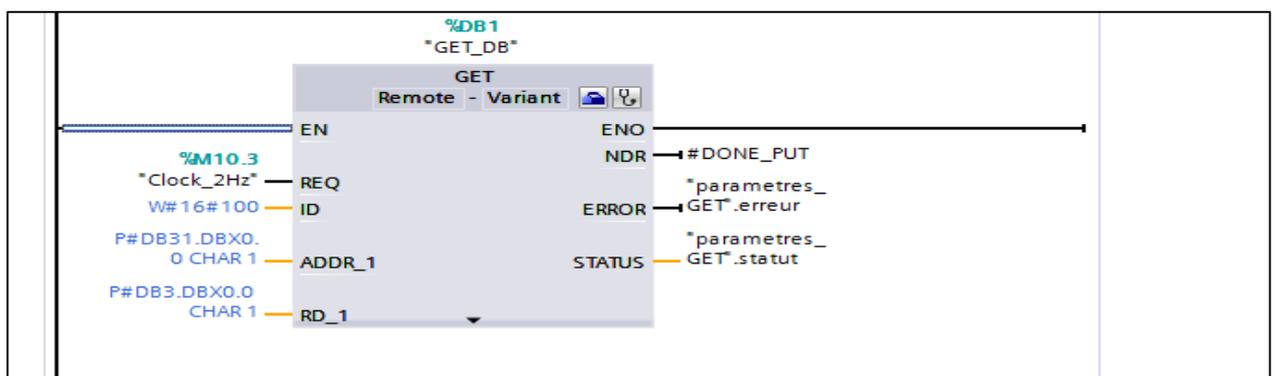


Figure IV.11:L'instruction GET

- **PUT**: est une instruction qui permet d'écrire des données dans une CPU distante. Ses paramètres principaux :

**ADDR\_1** :c'est un pointeur désignant les zones où écrire dans la CPU partenaire (ET 200SP).

**SD\_1**:c'est un pointeur désignant les zones de la propre CPU (S7-1500) qui contiennent les données à envoyer.

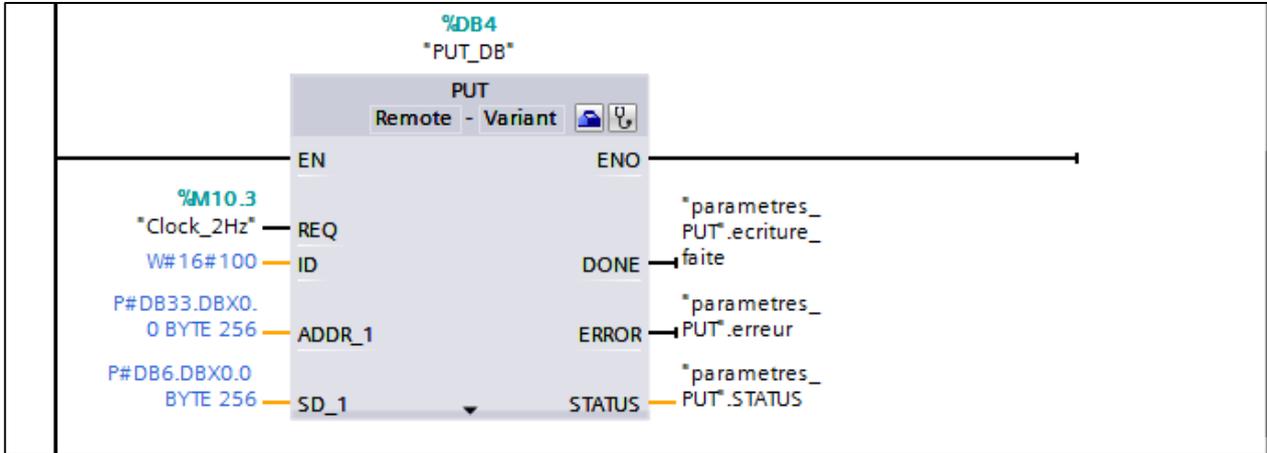


Figure IV.12: L'instruction PUT

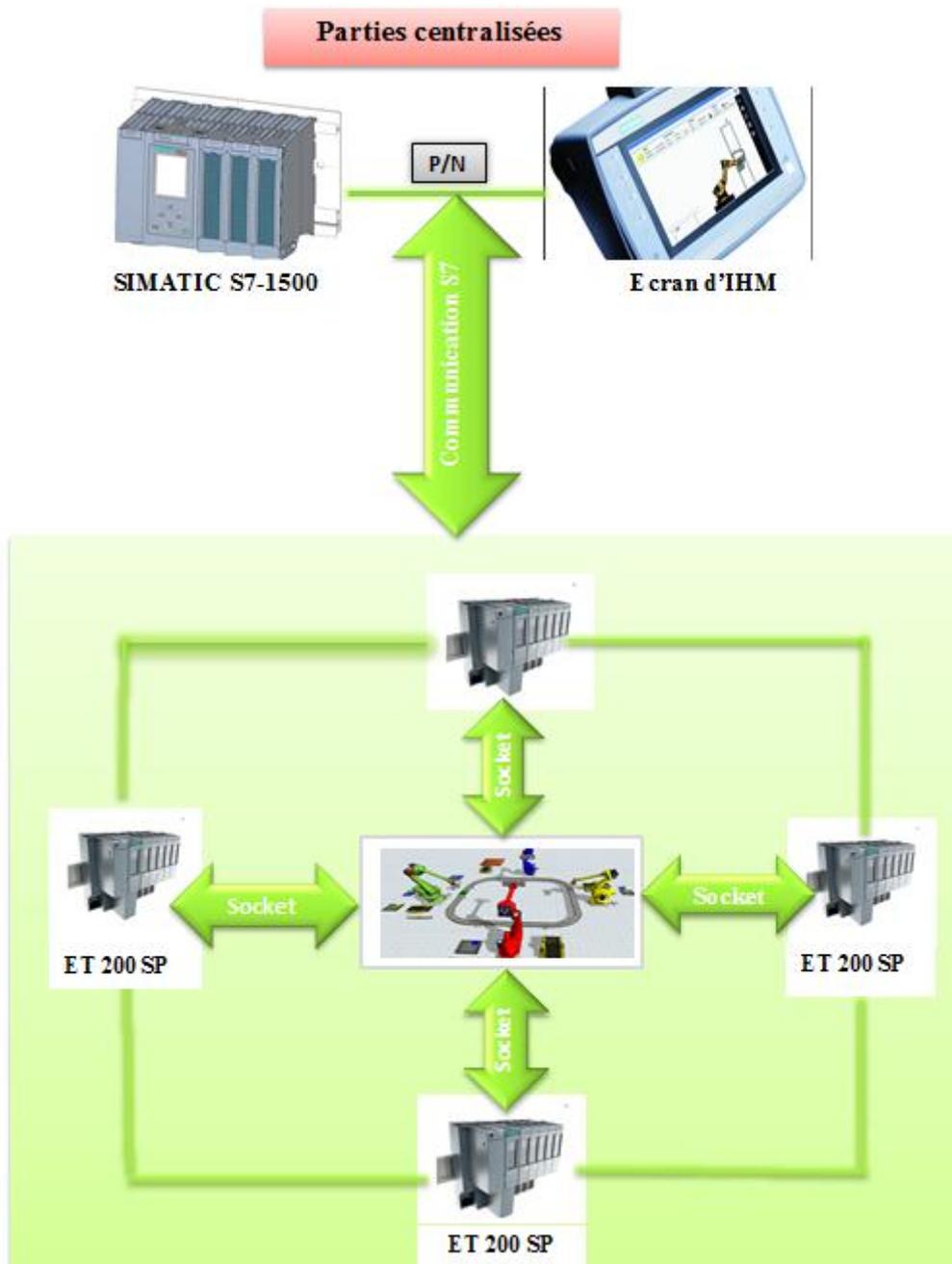
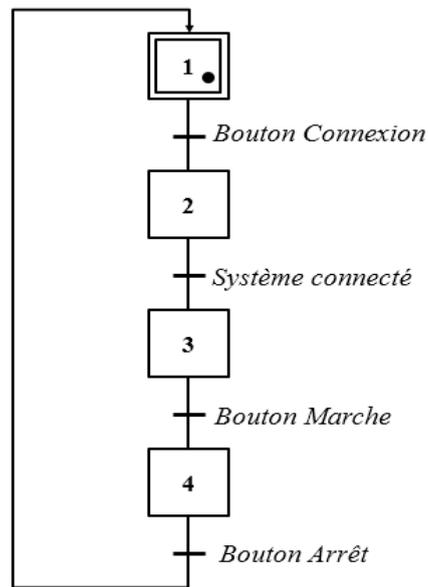


Figure IV.13: Schéma résumant les différents types de communication utilisés

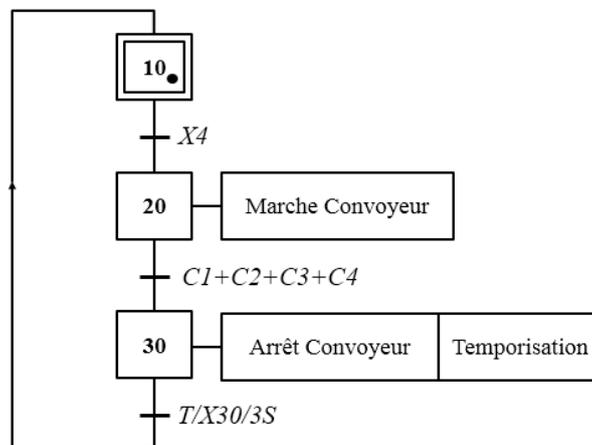
IV.2 Pilotage de la cellule :

IV.2.1 Grafcet de l'application :

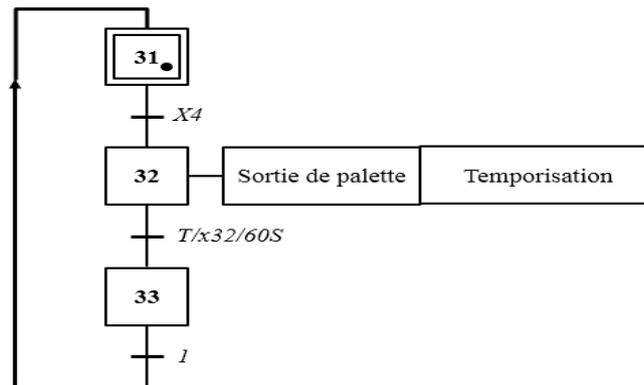
a. Grafcet de lancement du système :



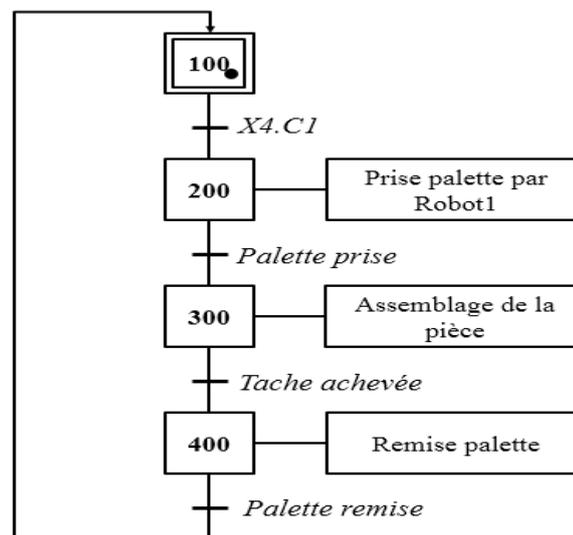
b. Grafctde système de convoyage :



c. Grafcet de la source des palettes :



d. Grafcet de système d'assemblage :

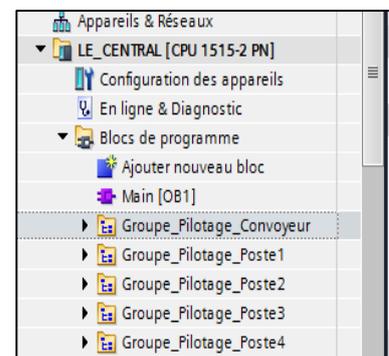


Ce graphe représente le fonctionnement du premier poste, ceux des autres postes seront identiques à celui-là en changeant le capteur C1 et le Robot1 par le capteur et le robot correspondants au poste.

IV.2.2 La structuration du programme :

Pour organiser notre programme de pilotage sur l'automate central, on a dû le séparer en cinq groupes :

- Groupe de pilotage de poste 1.
- Groupe de pilotage de poste 2.
- Groupe de pilotage de poste 3.
- Groupe de pilotage de poste 4.
- Groupe de pilotage de convoyeur.

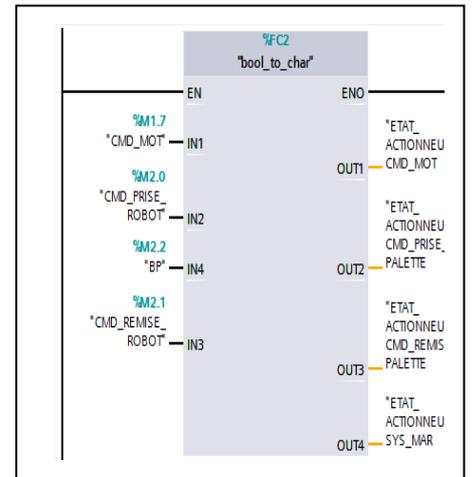




✓ Groupe d’encodage de données :

Une fois que les sorties du systèmes sont mises à jour selon l’état des entrées, elles seront encodées de telle sorte que flexsim pourra les interpréter ; on encode l’état de chaque sortie qui est du type booléen en une chaine de caractère en utilisant une fonction qu’on a créé et qu’on a programmé en SCL nomée « bool to char », et puis on concatine toutes les chaines correspondantes à toutes les sorties dans une seule chaine en utilisant l’instruction

« CONCAT », et on la transfère vers le bloc de l’écriture pour la transmettre à flexsim, cela nous permet d’assurer une commande complète du système et en temps réel.



IV.3 La supervision de la plateforme :

IV.3.1 Variable HMI :

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 un pupitre opérateur et un automate.
- Les variables internes ne possèdent aucun lien avec l’automate, elles sont enregistrées dans la mémoire du pupitre.

IV.3.2 Table des variables HMI :

La figure suivante représente une partie de la table des variables HMI :

Table de variables standard				
Nom	Type de données	Connexion	Nom API	Variable API
BP_Arret	Bool	HMI_Liaison...	LE_CENTRAL	BP_Arret
BP_Connex	Bool	HMI_Liaison_1	LE_CENTRAL	BP_Connex
BP_Marche	Bool	HMI_Liaison_1	LE_CENTRAL	BP_MARCHE
CMD_Convoyeur	Bool	HMI_Liaison_1	LE_CENTRAL	CMD_Convoyeur
CMD_Robot1_Prise	Bool	HMI_Liaison_1	LE_CENTRAL	CMD_Robot1_Prise
ET1_Connecté	Bool	HMI_Liaison_1	LE_CENTRAL	ET1_Connecté

Figure IV.14 : Table des variables HMI.

IV.3.3 les vues :

Pour le contrôle et la commande du système on a configuré 3 vues.

IV.3.3.1 La hiérarchie des vues :

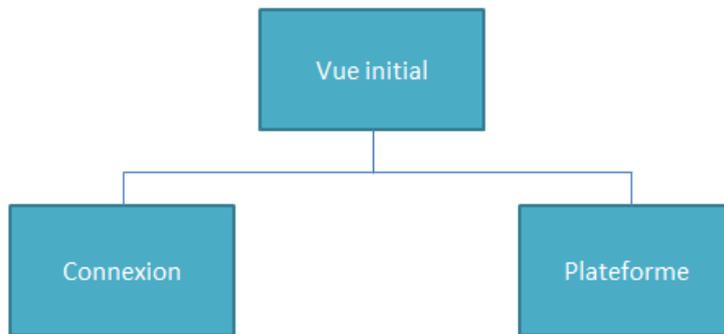


Figure IV.15: La hiérarchie des vues

IV.3.3.2 La vue initiale :

C'est la vue qui s'affiche au lancement elle est générale, Depuis cette vue l'opérateur peut accéder à la vue global (plateforme) mes encore à la vue de réseau.



Figure IV.16 : La vue initial

IV.3.3.3 La vue globale :

C'est une vues détaillé elle représente les quatre postes de travail de la plateforme, robot, convoyeur.

Depuis cette vue on peut revenir à la vue initial ou bien aller vers la vue des réseaux

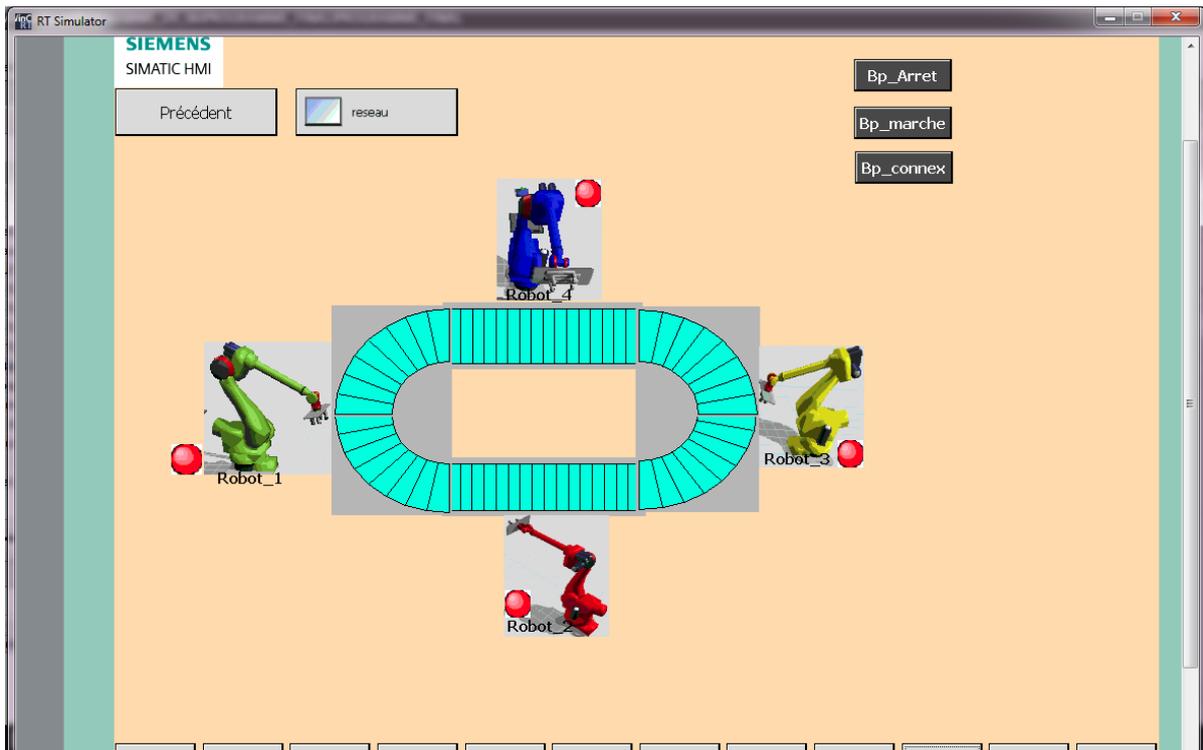


Figure IV.17 : La vue globale

**IV.3.3.4 La vue de réseau :**

C'est une vue qui permet de vérifier que la connexion entre chaque ET 200SP avec le centrale S7-1500 est établie, et aussi de pouvoir commandé la plateforme( marche/arret) et d'établir la connexion .

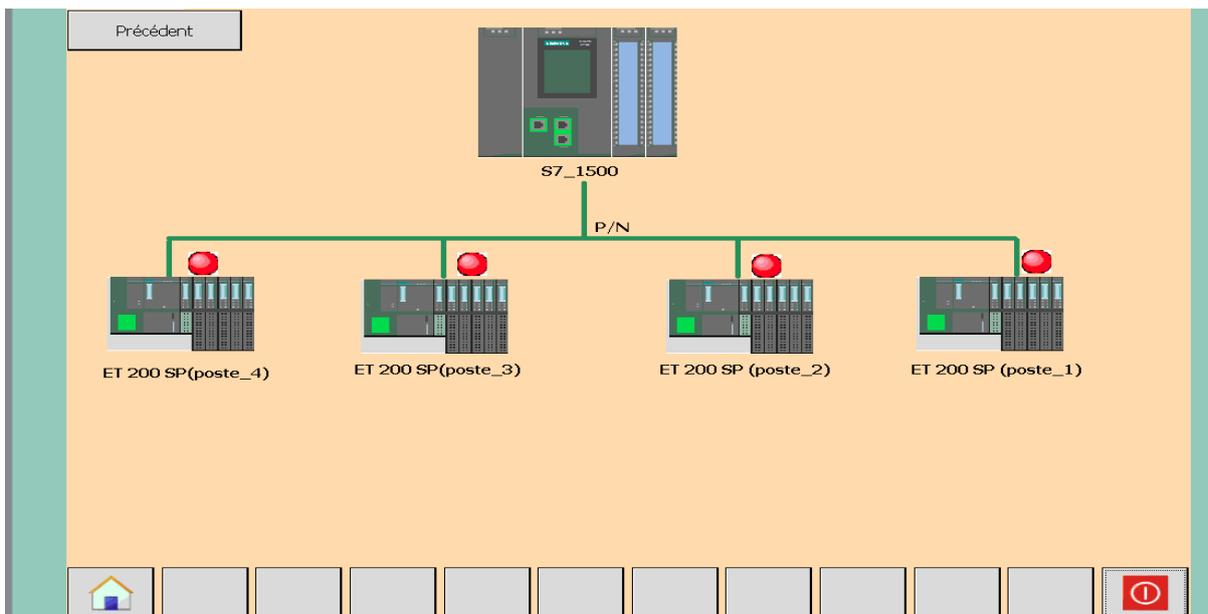


Figure IV.18 : Vue de réseau

#### **IV.4 Conclusion**

Dans ce chapitre on a présenté le pilotage de notre cellule et les différents types de communication qu'on a utilisé pour établir un bon échange de données entre la partie commande et la partie opérative , et la partie IHM qui nous permettra de contrôler et de commander notre plateforme.

**Conclusion générale :**

La commande de la plateforme d'assemblage robotisée virtuelle conçue sur le logiciel simulateur flexsim par un PLC (S7-1500) en se basant sur les techniques Hardware In The Loop a nécessitait d'une part, la création de plusieurs interfaces socket pour la réception et la transmission de données, d'autre part la mise en place d'un réseau local industriel reliant les quatre entrées/sorties décentralisées (ET 200 SP) avec le PLC central (S7\_1500) pour assurer un bon déroulement d'échanges de données et alléger les taches de l'automate central.

Ce projet nous a permet de voir les caractéristiques du matériel utilisé, de découvrir comment que nos contrôleurs répondent à des stimuli virtuels réalistes, et de tester leurs flexibilités et leurs vitesses de traitement.

Le projet qu'on a réalisé peut évoluer vers différents horizons. Il peut être envisageable de changer le fonctionnement de telle sorte que la plateforme produise plusieurs gammes opératoires en adoptants des techniques d'ordonnancement plus complexe. Il sera aussi intéressant de faire l'implémentation réelle de la plateforme actuelle tout en gardant notre modèle virtuel en tant que système observateur.

## Bibliographie

- [1] « Système bouclé » cours, école Renardfollereau, 2009.
- [2] Philippe LE BRUN « Automate programmable industriels » cours, Strasbourg, 1999.
- [3] W.B « Les bus et les réseaux de terrain en automatisme industriel » école Jacquard,Paris, 2002.
- [4] D.A « Cours Réseaux Locaux Industriels » mémoire d'ingénieur, Université Mohamed Khider-Biskra, 2011.
- [5]P.B, « Introduction à la supervision »,cours, Université Lile 1, 2010.
- [6] « Détecteurs photoélectriques » Guide Technique,2003.
- [7] « [futura-sciences.com/tech/](http://futura-sciences.com/tech/) ».
- [8] K.G« Automatismes et Systèmes à évènements discrets » cours, Polytech' Montpellier.
- [9] « what is Hardware In The loop ?», documentation Mathworks.
- [10] « Hardware-In The loop simulation for validating PLC Programs »,CDTA ,Alger .
- [11] «[https ://ww.emse.fr](https://ww.emse.fr) »,introduction à la programmation et à la modélisation.
- [12] haouari-lobna « MODELISATION ET SIMULATION DE L'INTRODUCTION DE TECHNOLOGIES RFID DANS DES SYSTEMES DE CONFIGURATION A LA DEMANDE » Ecole Nationale Supérieure des Mines de Saint-Etienne, France, 2012.
- [13] Site Flexsim.
- [14] Manuel Flexsim.
- [13] Siemens AG 2014, « Module 020-011 TIA Portal Introduction à la programmation avec SIMATIC S7-1500 »,Documentation,2014.
- [14] site SIEMENS.
- [15] SIMATIC, « ET 200 SP CPU 1512-1PN,Manuel,2016.

[16] SIEMENS, <https://w5.siemens.com/belux/web/fr/industrie/industrie/automatisation/et200sp/et200sp/pages/default.aspx>.

[17] SEIMENS, « Moniteurs industriels Industrial Flat Panel IFP1500, IFP1900, IFP2200 », 2012.

[18] Siemens AG, « SIMATIC STEP 7 dans le portail Totally Integrated Automation Portal », 2013.

[19] TIA portale V14, Help.

[20] « <https://w3.siemens.com> ».

[21] « <https://searchnetworking.techtarget.com> ».

[22] « <https://www.techniques-ingenieur.fr> ».

[23] « <https://www.PROFINET IO/PROFINET IO Protocol Overview.html> ».

## ANNXE 1

### GRAFCET

Le GRAFCET (Graphe Fonctionnel de Commande Etape-Transition) ou SFC (sequential functional chart) est un diagramme de description du comportement déterministe de la partie commande d'un système automatisé.

Le GRAFCET permet de construire des modèles ayant une structure graphique (représentation statique) à laquelle on associe une interprétation (elle correspond à l'aspect fonctionnel du grafcet). De plus, ce modèle possède un comportement dicté par des règles d'évolution (représentation dynamique), complétées pour l'implémentation par des algorithmes d'application de ces règles.

#### 1. Structure graphique du Grafcet

Le Grafcet est un graphe cyclique composé alternativement de transition et d'étape, reliée entre elles par des liaisons orientées. Des actions peuvent être associées aux différentes étapes, la figure ci-dessous représente un Grafcet.

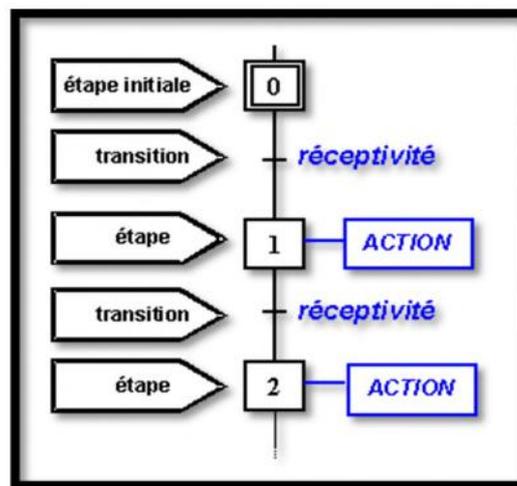


Figure A.1 : Représentation d'un Grafcet

#### L'étape

L'étape symbolise un état ou une partie de l'état du système, elle caractérise un comportement invariant du système considéré.

#### La transition

La transition permet de décrire l'évolution possible de l'état actif d'une étape à une autre, c'est elle qui va permettre, lors de son franchissement, l'évolution du système, elle représente une possibilité de changement d'état du système.

#### Les réceptivités

Une réceptivité est associée à chaque transition (l'absence de réceptivité est en fait la réceptivité toujours vraie), c'est une condition qui détermine la possibilité ou non d'évolution du système par cette transition.

## Les actions associées aux étapes

Les actions servent à émettre des ordres vers la partie opérative. Une action est une sortie logique envoyée au système. Ces actions peuvent être :

- Des actions continues
- Des actions conditionnelles
- Des actions mémorisées
- Des actions temporisées

## 2. Bases du langage

- Etape, action
- Caractérise le comportement de la partie commande.
- Une étape est soit active soit inactive.
- Action associée (facultatif).

La figure ci-dessous est une représentation des différents types d'étapes :

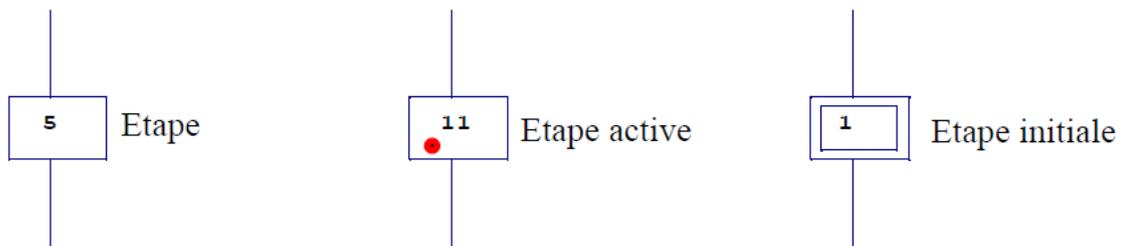
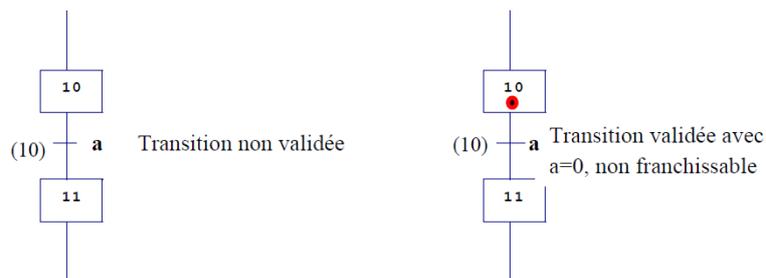


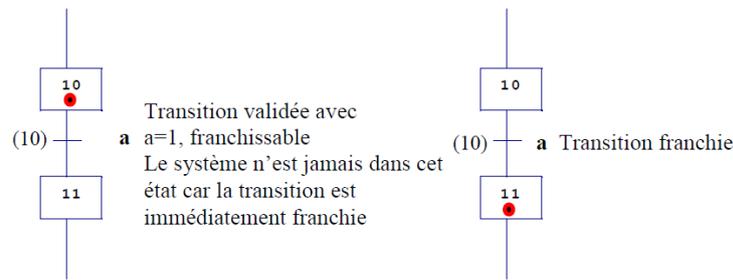
Figure A.2 : Différents types d'étapes

Transition, réceptivité

- Possibilité d'évolution entre les étapes.
- Validée lorsque les étapes précédentes sont actives.
- Réceptivité associée (condition logique).

La figure ci-dessous est une représentation des différents types de transitions :





**Figure A.3 Différents types de transitions**

- Des liaisons orientées (arcs).
- Liaison orientée (étape transition étape).

### 3. Règles d'évolution d'un grafcet

Un grafcet possède un comportement dynamique dirigé par cinq règles, elles précisent les causes et les effets du franchissement des transitions.

#### - **Règle1 : situation initiale**

La situation initiale d'un grafcet caractérise le comportement initial de la partie commande vis-à-vis de la partie opérative, de l'opérateur et/ou des éléments extérieurs.

Elle correspond aux étapes actives au début du fonctionnement : ces étapes sont les étapes initiales (doublement du symbole d'étapes).

#### - **Règles2 : franchissement d'une transition**

Une transition est dite validée lorsque toutes les étapes en amont (immédiatement précédents reliées à cette transition) sont actives. Le franchissement d'une transition se produit :

- Lorsque la transition est validée
- Que la réceptivité associée à cette transition est vraie

#### - **Règles3 : évolution des étapes actives**

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

#### - **Règles4 : activation et désactivation simultanée d'une étape**

Si au cours du fonctionnement la même étape est simultanément activée et désactivée elle reste active. On évite ainsi des commandes transitoires (néfastes au procédé) non désirées.

### 4 Le diagramme à relais ou schéma à contacts

Les diagrammes à relais (LADDER) permettant de représenter des conditions logiques de façon similaire aux armoires de commande à relais utilisées avant l'arrivée des automates programmables industriels. Les entrées sont des relais (en série et/ou en parallèle) qui sont reliés à une bobine (la sortie), ces éléments sont placés entre deux lignes d'alimentation, la bobine d'un

premier réseau peut être utilisée comme entrée d'un autre réseau, d'autres éléments peuvent entrer dans un réseau : compteurs, temporisateurs.

## 5. Les différents Grafcet

Il y a deux types de représentation :

- **La représentation fonctionnelle ou de niveau 1** donne une interprétation de la solution retenue pour un problème posé, en précisant la coordination des tâches opératives. Elle permet une compréhension globale du système.
- **La représentation technologique ou de niveau 2** donne une interprétation en tenant compte des choix technologiques relatifs à la partie de commande de l'automatisme ; le type et la désignation des appareillages.

Dans les cas plus généraux et plus complexes, le grafcet maître (superviseur) s'occupe du démarrage, de la séquence, de la synchronisation et de l'arrêt de différentes tâches. Chaque tâche est décrite par un grafcet commandé par le superviseur ou grafcet de gestion des tâches on parle ici de Notion de structures hiérarchisées

## 6. Structure hiérarchisées d'un grafcet :

### 6.1. Structure d'un Sous-grafcet :

Dans les automatismes séquentiels, il est fréquent de rencontrer des séquences répétitives dans le même cycle. Une séquence répétitive peut être représentée par un sous-grafcet ou grafcet sous-programme. Cette notion est empruntée au langage informatique.

Un grafcet sous-programme est écrit sous la forme d'un grafcet indépendant, connecté au grafcet principal.

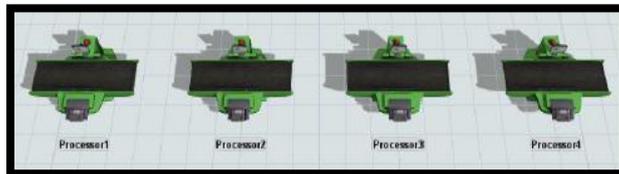
**6.2. Structure d'un grafcet de tâche :** Les tâches ont pour but de simplifier et de faciliter la description de systèmes complexes en allégeant le graphisme d'un grafcet et en détaillant séparément certaines parties. (à rapprocher des procédures en programmation Informatique)

## ANNEXE 2

## FlexSim :

## 1. Insertion d'objets dans le modèle :

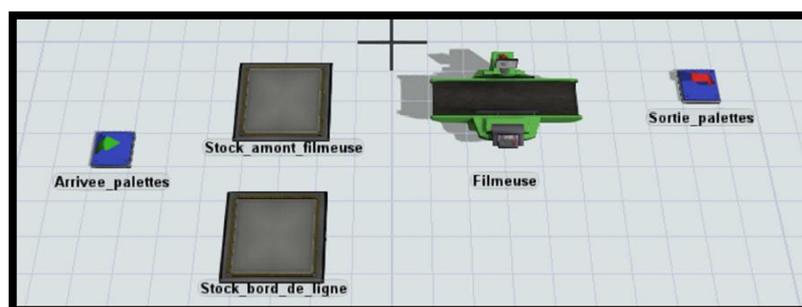
- 1. Par **glisser-déposer** dans la vue 3D
- 2. Par **double clic** dans la vue 3D
  - puis sélectionner l'objet choisi
- 3. Mode **rapide** de création
  - 1 clic gauche de la souris sur un objet de la librairie
  - puis à chaque clic dans la vue 3D une copie est créée
  - très pratique si vous avez **plusieurs objets identiques à insérer**



- pour **quitter** ce mode, 1 **clic droit de la souris** dans la vue ou sur la touche échappe.

## 2. Disposition du modèle :

**Glisser-déposer** les objets de la librairie vers la vue 3D

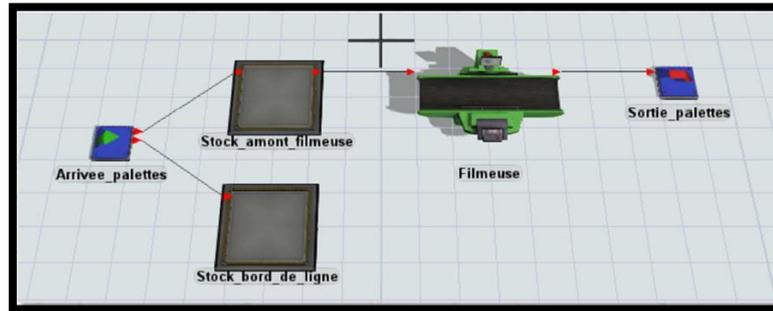


- Une source.
- Deux Queue.
- Un Processeur.
- Un puits.

Toujours un modèle démarre par une source et se termine par un puits.

## 3. Créer les connexions de flux :

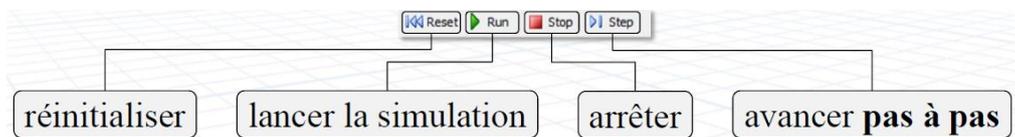
En maintenant la touche **Q** enfoncée, on relie les éléments entre eux. Connecter les objets pour définir les différents **chemins** que peuvent emprunter les palettes.



Les liaisons entre les éléments

#### 4. Exécution du modèle :

- Boutons de contrôle



#### 5. Configuration de la machine pour visualiser les pièces en mouvement :

Il faut faire un double clic sur le processeur pour obtenir l'ensemble des propriétés. L'action à ajouter concerne l'événement **OnEntry** et utiliser (+) pour sélectionner dans le menu contextuel le menu **Data** et ensuite **Set item Type and Color**.

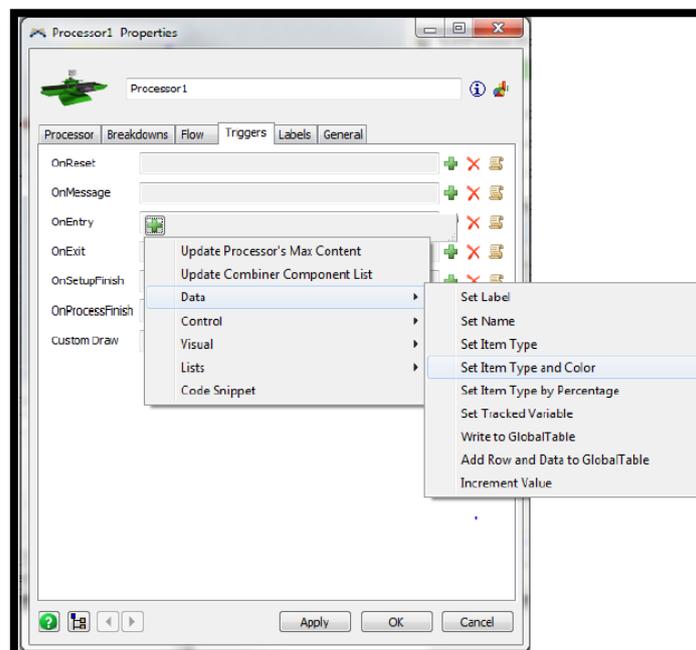


Figure II.3 : Configuration de la machine

## 8. Translation d'un objet

