Ordre...../F.S.S.A/UAMOB/2019

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE AKLI MOHAND OULHADJE-BOUIRA

Faculté des Sciences et des Sciences Appliquées
Département : **Génie Electrique**

Mémoire de fin d'étude

Présenté par :

REZOUG Sofiane
SAHNOUNE Hafsa

En vue de l'obtention du diplôme de **Master 02** en **:**

Filière : Electromécanique
Option : Electromécanique

Thème :

Pilotage d'une cellule robotisée simulée à l'aide d'un automate programmable

Devant le jury composé de :

S. Mouassa	MAA	UAMOB	Président
K. SAOUDI	MCA	UAMOB	Encadreur
Y. TOUAFEK	MAA	UAMOB	Examinateur
S. HAROUN	MAB	UAMOB	Examinateur
M. GAHAM	MAA	CDTA	Membre invité

Année Universitaire 2018/2019

Résumé :

Le pilotage d'une cellule robotisé simulée à l'aide d'un automate programmable industriel est l'objet du présent document. Ce travail consiste à réaliser une plateforme virtuelle pour commander un bras de robot de modèle KUKA à travers des automates programmables siemens ET 200SP décentralisés et les centralisent par un S7 1500 sous le logiciel TIA PORTAL avec le SIMATIC S7-PLCSIM Advanced qui permet de créer des contrôleurs virtuels. La solution proposée est basée sur la technique HIL « Hardware In the Loop » qui permet de tester un système physique virtuellement sans utiliser une installation réelle. Les mouvements du bras sont visualisés en trois dimensions sur le logiciel simulateur V-REP où les programmes fonctionnels de la plate-forme sont écrits.

Mots clés : Cellule robotisée, API ET200SP, API S7-1500, Commande, Plateforme virtuelle, le logiciel TIA PORTAL, le SIMATIC S7-PLCSIM Advanced, Technique HIL « Hardware In the Loop », Système physique, V-REP.

Abstract:

Piloting a robotic cell using an industrial programmable logic controller is the subject of this document. This work consists of creating a virtual platform to control a KUKA robot arm via decentralized ET200SP programmable controller and centralize them with an S7 1500 through the TIA PORTAL software with SIMATIC S7-PLCSIM advanced which allows to create virtual controllers. The proposed solution is based essentially on the HIL technique "hardware in The Loop", that allows you to test a physical system virtually without using a real installation. The arm movements will be visualized in three dimensions on the V-REP simulator software where the functional programs of the platform are written.

Keywords: Robotic cell, ET200SP PLC, S7-1500 PLC, Command, Virtual Platform, TIA PORTAL software, SIMATIC S7-PLCSIM Advanced, The HIL technique "hardware in The Loop", Physical System, V-REP.

ملخص:

يعتبرالتحكم في خلية روبوتية محاكاة باستخدام جهاز تحكم منطقي صناعي قابل للبرمجة موضوع هذا السند. يتكون هذا العمل من إنشاء منصة افتراضية للتحكم في ذراع الروبوت KUKA عن طريق وحدات التحكم المنطقية اللامركزية SIEMENS ET200SP وإضفاء الطابع المركزي عليها بواسطة S71500 تحت برنامج TIA PORTAL مع SIMATIC S7-PLCSIM Advanced والذي يسمح بإنشاء وحدات تحكم افتراضية. يعتمد الحل المقترح على تقنية الأجهزة في حلقة « HIL" Hardware In the Loop Solution" على تقنية ، التي تسمح اختبار النظام الحالي تقريبًا دون استخدام المعدات الحقيقية, يتم عرض حركات الذراع بثلاثة أبعاد على برنامج محاكاة V-REP اين تتم كتابة البرامج الوظيفية للنظام الأساسي.

الكلمات المفتاحية: خلية روبوتية ، وحدة التحكم ET200SP ، وحدة التحكم S7-1500 ، التحكم ، منصة افتراضية ، برنامج TIA PORTAL، SIMATIC S7-PLCSIM Advanced ، تقنية HIL "الأجهزة في الحلقة" ، النظام الفعلي ، V-REP.

Dédicace :

Nous avons l''honneur de dédier ce travail,
Aux êtres les plus cher de notre vie : père & mère qui nous ont encouragé, aidé du mieux qu'il leur est possible de faire, et qui avec patience ont attendu ce joyeux événement. Que Dieu leur prête une très longue vie de paix et de prospérité.

A nos frère, à nos sœurs, à toute la famille REZOUG et SAHNOUNE ;

A tous nos camarades et amis d'ici et d'ailleurs ;

A tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste travail ;

Et à tous ceux qui sèment le bonheur dans notre chemin.

En ces quelques mots, on leurs exprime toute notre gratitude et nos sincères salutation.

Remerciement

En premier lieu, nous remercions le bon Dieu, le tout puissant, pour nous avoir donné, la patience, la volonté et la force nécessaires pour achever ce travail.

Nous tenons à exprimer toute nos reconnaissances à notre directeur de mémoire de l'université, Monsieur SAOUDI Kamel. Nous le remercions de nous avoir encadré, orienté, aidé et conseillé surtout.

Nous remercions en particulier monsieur GAHAM Mahdi, notre encadreur du stage pratique au niveau du CDTA, qui nous a donné l'occasion de réaliser ce travail et pour le temps qu'il a consacré à nous apporter les outils méthodologiques indispensables à la conduite de cette réalisation.

Un grand merci également à monsieur REZOUG Amar pour leurs encouragements et parce qu'il était toujours présent pour nos questions.

Nous adressons nos remerciements les plus respectueux aux membres du jury. Ils nous ont fait l'honneur d'accepter de rapporter et d'examiner ce mémoire de master et nous ont apporté, à travers leurs remarques et leurs questionnements.

On aimerait exprimer notre gratitude à tous qui nous a aidée à faire avancer notre projet.

Enfin, on tient à remercier les membres de nos familles frères et sœurs, exceptionnellement nos chers parents pour tout ce qu'ils ont fait pour nous. Ils ont joué un rôle peut être plus important que le nôtre dans ce mémoire. Leurs encouragements, leur motivation débordante et souvent contagieuse.

Introduction générale

Introduction générale : 1

Chapitre I : Généralités sur les systèmes automatisés

I.1 Introduction : **3**
I.2 Industrie 4.0 : 3
I.2.1 Définition : 3
I.2.2 Avantages de l'industrie 4.0 [3] **:** 4
I.3 Les systèmes automatisés : 4
I.3.1 Echange d'information : 5
I.3.2 Avantages et inconvénients de l'automatisation [4] **:** 5
I.3.3 Structure et organisation générale d'un système automatisé : 5
I.4 Automates Programmables Industrielles (API) : 8
I.4.1 Historique : 8
I.4.2 Définition : 9
I.4.3 Structure d'un automate programmable industriel : 9
I.4.4 Langages de programmation des API [11] : 13
I.4.5 Critères de choix d'un automate [12] **:** 14
I.4.6 Domaines d'utilisation des automates programmables industriels : 14
I.5 Les automatismes centralisés et décentralisés : 14
I.5.1 Les automatismes centralisés [14] : 14
I.5.2 Les automatismes décentralisés [14] : 15
I.6 Conclusion : 16

Chapitre II : Modélisation de la cellule en utilisant le logiciel V-REP

II.1 Introduction : 17
II.2 La Plateforme Virtuelle d'Expérimentation Robotique V-REP : 17
II.2.1 Description : 17
II.2.2 L'utilisation de logiciel V-REP : 19
II.2.3 Le langage de programmation de V-REP : 20
II.3 Fonctionnement de la cellule : 20
II.3.1 Présentation d'un modèle de la cellule et son fonctionnement : 20

II.3.2 Cahier de charge de l'application : 21
II.4 Les techniques de validation en environnement émulé (Hardware-In-The Loop) : 21
II.4.1 Définition de Hardware-In-The-Loop (HIL): 22
II.4.2 L'intérêt de la simulation HIL [16] : 22
II.4.3 Utilisation de HILS (Hardware In The Loop Simulation): 22
II.4.4 Spécification sur le rôle des techniques HIL dans notre projet : 23
II.5 La simulation : 24
II.5.1 Définition de la simulation : 24
II.5.2 Domaines d'application de la simulation : 24
II.5.3 Les avantages et les inconvénients de la simulation [18] : 24
II.6 La mise en place et la programmation de la plateforme simulée : 25
II.7 La communication socket [20] : 29
II.8 Conclusion : 30

Chapitre III : Implémentation de la commande en utilisant l'API

III.1 Introduction : 31
III.2 Objectifs du projet : 31
III.3 Automate Siemens S7-1500 : 31
III.3.1 Les différents composants de notre automate S7-1500 [21] : 32
III.4 Automate ET-200SP [22] : 33
III.4.1 Les différents composants de notre automate ET-200SP [22] : 34
III.5 Interface Homme-Machine « IHM » SIMATIC IFP2200 [20] : 35
III.6 TIA Portal (Totally Integrated Automation): 36
III.6.1 Description du logiciel TIA Portal : 36
III.6.2 Les avantage du logiciel TIA portal [6] : 36
III.6.3 Vue du portail et vue du projet : 37
III.6.4 Création d'un projet et configuration d'une station de travail : 39
III.6.5 Configuration matériel (hardware) [23] : 40
III.6.6 Adresse Ethernet de la CPU : 41
III.6.7 Compilation et chargement de la configuration matérielle : 42
III.6.8 PLCSIM Advanced V2.0 : 46
III.7 Compatibilité avec TIA Portal et les versions du microprogramme de la CPU : 46
III.8 Réseau PROFINET [26] **:** 47
III.8.1 Spécification sur le PROFINET IO [26] : 47
III.9 Conclusion : 48

Chapitre II : Modélisation de la cellule en utilisant le logiciel V-REP

IV.1 Introduction : 49
IV.2 Architecture de pilotage de la cellule et système de communication : 49
IV.3 Communication SOCKET : 51
IV.4 La création d'une communication socket sur V-REP : 51
IV.5 Création du tableau des variables API : 52
IV.6 Ajout des blocs d'organisation (OB) : 52
IV.7 La création d'une interface socket sur l'ET 200 SP : 53
IV.8 La Création d'une communication socket entre Les quatre ET200 SP : 58
IV.9 Communication S7 [28] : 59
IV.10 Pilotage de la cellule : 61
IV.10.1 GRAFCET du système complet : 61
IV.11 Conclusion : 63

Conclusion générale

Conclusion générale……64

Liste des figures

Figure I. 1 Les quatre révolutions industrielles 4
Figure I. 2 Structure d'un système automatisé 6
Figure I. 3 Les prés-actionneurs 7
Figure I. 4 Les actionneurs 7
Figure I. 5 Les Capteurs 7
Figure I. 6 Structure détaillée d'un API 9
Figure I. 7 Structure interne d'un API 10
Figure I. 8 Les interfaces d'entrées/sorties 12
Figure I. 9 Exemple d'une carte d'entrée typique d'un API 12
Figure I. 10 Exemple d'une carte de sortie typique d'un API 13
Figure I. 11 Les automatismes centralisés. 15
Figure I. 12 Les automatismes décentralisés 16

Figure II. 1 L'application V-REP 18
Figure II. 2 Un model représentant de la cellule 21
Figure II. 3 Le programme du simple convoyeur. 26
Figure II. 4 Programme de simple convoyeur circulaire. 26
Figure II. 5 Programme de la pince de robot sur V-REP. 28
Figure II. 6 Programme de position de robot. 28
Figure II. 7 Le programme de la tâche du robot. 29

Figure III. 1 Automate Siemens S7-1500 31
Figure III. 2 CPU 1515-2PN 32
Figure III. 3 Constitution de l'Automate S7-1500 33
Figure III. 4 L'automate ET-200SP 34
Figure III. 5 L'alimentation SITOP PSU100S 34
Figure III. 6 Constitution d'un automate ET-200SP. 35
Figure III. 7 L'interface homme-machine IHM SIMATIC IFP2200. 36
Figure III. 8 Vue du portail 38
Figure III. 9 Vue du projet 38
Figure III. 10 Création d'un projet 39
Figure III. 11 Première methode deconfiguration et paramétrage du matériel 40
Figure III. 12 Deuxième méthode de configuration et paramétrage du matériel 41
Figure III. 13 Adresse Ethernet de la CPU 42
Figure III. 14 Compilation de la configuration matérielle 43
Figure III. 15 Compilation et chargement en mode de connexion PN/IE. 44
Figure III. 16 Compilation et chargement en mode de connexion MPI 45
Figure III. 17 Confirmation de chargement de la configuration matérielle 46

Figure IV. 1 Schéma représentant la connexion des différents éléments 50
Figure IV. 2 Switch TP-LINK 50
Figure IV. 3Communication socket direct entre VREP et API 51
Figure IV. 4 Communication socket sur V-REP (connexion avec l'automate ET 200SP) 51
Figure IV. 5 Communication socket sur V-REP (réception des données) 52

Figure IV. 6 Tableau de variable 52
Figure IV. 7 Réseaux de blocs OB réalisé. 53
Figure IV. 8 La bibliothèque communication. 54
Figure IV. 9 Bloc d'instance de l'instruction TCON. 55
Figure IV. 10 Paramétrage de l'instruction TCON. 55
Figure IV. 11 Bloc d'instance de l'instruction TSEND. 56
Figure IV. 12 Bloc d'instance de l'instruction TRCV. 57
Figure IV. 13 Bloc d'instance de l'instruction de déconnexion. 57
Figure IV. 14 Les blocs d'instance de TCON de quatre ET 200 SP. 58
Figure IV. 15 Paramétrage de TCON. 58
Figure IV. 16 L'instruction GET 59
Figure IV. 17 L'instruction PUT 60
Figure IV. 18 Schéma résumant les différents types de communication utilisés. 60
Figure IV. 19 GRAFCET du système. 62

Liste des Tableaux

Tableau III. 1 Compatibilité avec TIA Portalet les versions 47

Tableau IV. 1 Les entrées sorties du GRAFCET. 61

Liste des Tableaux

Tableau III. 1 Compatibilité avec TIA Portalet les versions **47**

Tableau IV. 1 Les entrées sorties du GRAFCET. **61**

LISTE DES ABREVIATIONS

CDTA : Centre de Développement des Technologies Avancées

V-REP: Virtual robot experimentation platform

(E/S): Entrée/Sortie

TIA PORTAL: Totally Integrated Automation Portal

IHM: Interface Humane Machine

API: Automate Programmable Industriel

M2M: Machine to Machine

PID : Régulateur Proportionnel Intégrable Dérivable

PO : Partie Opérative

PC : Partie Commande

PR : Partie Pupitre

SCC : Système de Contrôle / Commande

CPU : Unité Centrale de Traitement

TOR : Tout Ou Rien

Grafcet : GRAphe Fonctionnel de Commande Etapes Transitions.

FBD : Function Bloc Diagram

RLI : réseaux locaux industriels

HIL : Hardware-in-the-Loop

ECU : l'unité de contrôle électronique

PLC : Programmable Logic Controller

LBR IIWA : Leicht bau roboter intelligent industrial work assistant

TCP : Transmission Control Protocol

UDP : Protocole de datagramme utilisateur

IO: Input output.

IP: Internet Protocol

PN/IE : Profinet/ Ethernet Industriel

MPI : Interface MultiPoint

DEL : Diode Electro Luminescente

SCL : Langage de Control Structuré

CONT : schéma à contact

LOG : Logigramme

LIST : Liste d’instruction

PROFINET: Process Field Ethernet

PROFIBUS: Process Field Bus

IRT: Temps Réel Isochrone

S7: communication Siemens

Introduction générale

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 moderne 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.

L'ensemble du secteur industriel est entré dans une phase de profonde mutation qui voit les technologies numériques s'intégrer au cœur des processus industriels. C'est l'industrie 4.0 « quatrième révolution industrielle » qui apparut après « la production automatisée avec les automates et les robots ». Cette intégration industrielle offre un extraordinaire champ d'innovation, de progrès et de croissance. Elle est caractérisée par la fusion du monde virtuel et le monde réel des installations industrielles. L'industrie 4.0 devient la référence incontournable pour la production industrielle.

Parmi les techniques qui permettent le développement et les tests de contrôle de systèmes complexes utilisés pour le fonctionnement des machines, on trouve les techniques HIL « Hardware In The Loop ».

La réalisation de ce projet au sein du CDTA « Centre de Développement des Technologies Avancées », il consiste à piloter une cellule robotisée sur le logiciel simulateur V-REP et de la commander à l'aide d'un automate programmable industriel (SIEMENS).

A cet effet, le présent mémoire est réparti en quatre chapitres décrivant les volets principaux :

Le premier chapitre présente le concept de l'industrie 4.0, suivie d'une description des systèmes automatisés et les différentes structures de base et aspects des automates programmables industriels (APIs).

Dans le deuxième chapitre, le logiciel simulateur V-REP sera présenté en expliquant son utilité et principe de fonctionnement qui sera semblable à la plateforme réelle. Et nous montrons la nécessité d'utiliser les techniques Hardware-In-The-Loop (HIL) pour sa validation avant de passer à l'implémentation réelle est démontrée.

Ensuite, dans le troisième chapitre sera présenté les composants de la partie contrôle (physique) de la cellule qui est constituée d'un automate central SIMATIC S7-1500 et quatre

entrées/sorties (E /S) décentralisée SIMATIC ET 200 SP. Nous présenterons aussi le logiciel TIA PORTAL qui permet la programmation pour atteindre le principal objectif de ce projet, ainsi qu'une IHM SIMATIC IFP2200.

A la fin, Le dernier chapitre présentera les étapes suivies pour l'architecture de pilotage adoptée pour la commande de la plateforme virtuelle et traitera la partie programmation de ce projet en explication détaillée.

En fin, ce mémoire se termine par une conclusion générale.

I.1 Introduction :

Les technologies numériques permettent au monde entier d'accélérer le développement socio-économique, de rapprocher leur habitat des services et des opportunités d'emploi, et de bâtir un avenir meilleur. Cette transformation appelée « Industrie 4.0 » ou « Quatrième Révolution Industrielles ».

L'intégration de l'informatique dans tous les domaines et plus particulièrement dans l'industrie a considérablement accéléré le développement de l'automatisation. Cette dernière est assurée généralement par des Automates Programmables Industriels (API).

L'automatisation industrielle a connu, au cours de ces dernières décennies, une évolution importante consécutive a croissement des exigences de qualité, de flexibilité et de disponibilité dans les procédés industriels. Dans l'esprit de l'époque, toutes les taches du processus de production étaient automatisables et devaient être automatisées entrainant l'évolution forte des architectures d'automatisme [1].

Ce chapitre est dédié à la présentation de l'industrie 4.0 suivie d'une description des systèmes automatisés et des automates programmables industriels (API) en générale.

I.2 Industrie 4.0 :

Aujourd'hui, il n'est plus question qu'un moyen de production produise à la chaîne (ou plutôt reproduise) un produit des milliers de fois. Nous sommes entrés dans l'ère de la personnalisation des produits. Le consommateur veut un produit complètement personnalisé, qui ne ressemble pas à celui de son voisin. L'industrie 4.0 s'engage à répondre à cette exigence de produits uniques et personnalisés tout en conservant des coûts équivalents, et cela malgré les faibles volumes de production engendrés. C'est pourquoi l'un des défis de cette 4ème révolution industrielle est de réussir à connecter le besoin du client à l'organe de production. Cette connexion ne peut se faire sans l'apport des nouvelles technologies, qui devront être exploitées dans cette "nouvelle usine"… [2]

I.2.1 Définition :

Le terme **''Industrie 4.0''** ou **''Industrie Futur''** fait référence à une 4ème révolution industrielle, après la mécanisation, la production de masse au 19ème siècle et l'automatisation de la production au 20ème siècle, elle se caractérise par l'intégration des technologies numériques dans les processus de fabrication. Cette nouvelle industrie s'affirme comme la convergence du monde virtuel, de la conception numérique, de la gestion (opérations, finance et marketing)

avec les produits et objets du monde réel. L'industrie 4.0 vise à réaliser de nouveaux gains de compétitivité et à optimiser des consommations par l'efficacité énergétique.

La figure suivante représente les quatre révolutions industrielles.

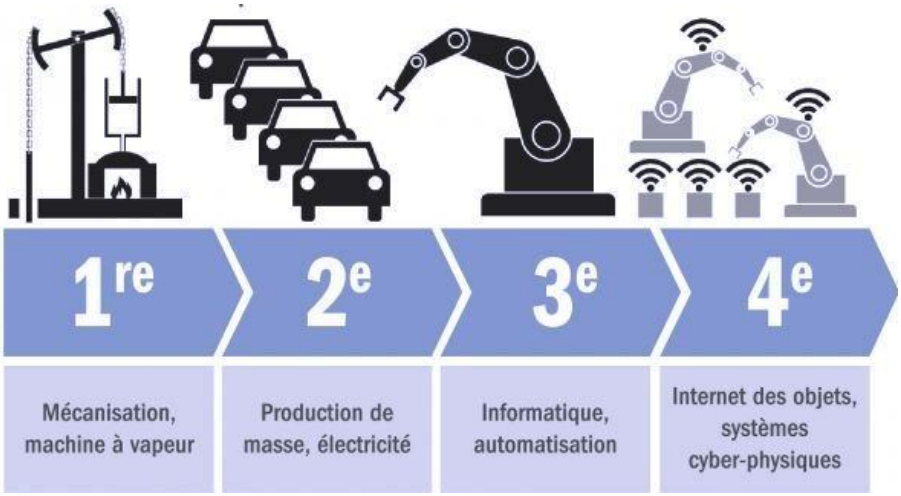

Figure I.1 Les quatre révolutions industrielles [2]

I.2.2 Avantages de l'industrie 4.0 [3] :

Les principaux points forts de cette 4ème révolution industrielle :

- L'usine numérique 4.0 se caractérise par une communication (M2M) continue et instantanée entre les différents outils et postes de travail intégrés dans les chaines de production et d'approvisionnement.
- Des outils de simulations et de traitements de données puissants pour créer des répliques virtuelles, former les ouvriers et techniciens, faciliter la maintenance…
- Une usine économique d'énergie et matières premières grâce à une meilleure coordination des besoins et disponibilités…

I.3 Les systèmes automatisés :

Dans un système automatique, les opérations programmées s'exécutent et s'enchaînent sans l'intervention de l'utilisateur. Un opérateur suit l'évolution du système et contrôle le bon déroulement du cycle de fonctionnement. Il assure la programmation, le démarrage et l'arrêt du système (en cas de problème).

Le système automatisé est composé d'une partie commande et d'une partie opérative.

I.3.1 Echange d'information :

- Le système attend les consignes de l'opérateur (personne qui va faire fonctionner le système).
- L'opérateur donne les consignes à la partie commande du système.
- La partie commande traduit ces consignes de l'opérateur en ordre.
- Ces ordres sont alors transmis à la partie opérative du système qui va les exécuter.
- La partie opérative rend compte à la partie commande de son état de fonctionnement.
- La partie commande signale à l'opérateur de l'état du système.

I.3.2 Avantages et inconvénients de l'automatisation [4] :

Avantages :

- Eliminer les tâches répétitives.
- Simplifier le travail de l'humain.
- Augmenter la sécurité.
- Accroitre la productivité.
- Economiser les matières premières et l'énergie.
- S'adapter à des contextes particuliers.
- Maintenir la qualité.

Inconvénients :

- Incidence sur l'emploi (chômage) : la mise en place d'une machine se substituant à 10 salariés n'aboutit pas la création de 10 emplois.
- Coût de maintenance.
- Les pannes.
- Investissement pour l'achat des machines.

I.3.3 Structure et organisation générale d'un système automatisé :

La figure suivante montre un schéma général d'un système automatisé.

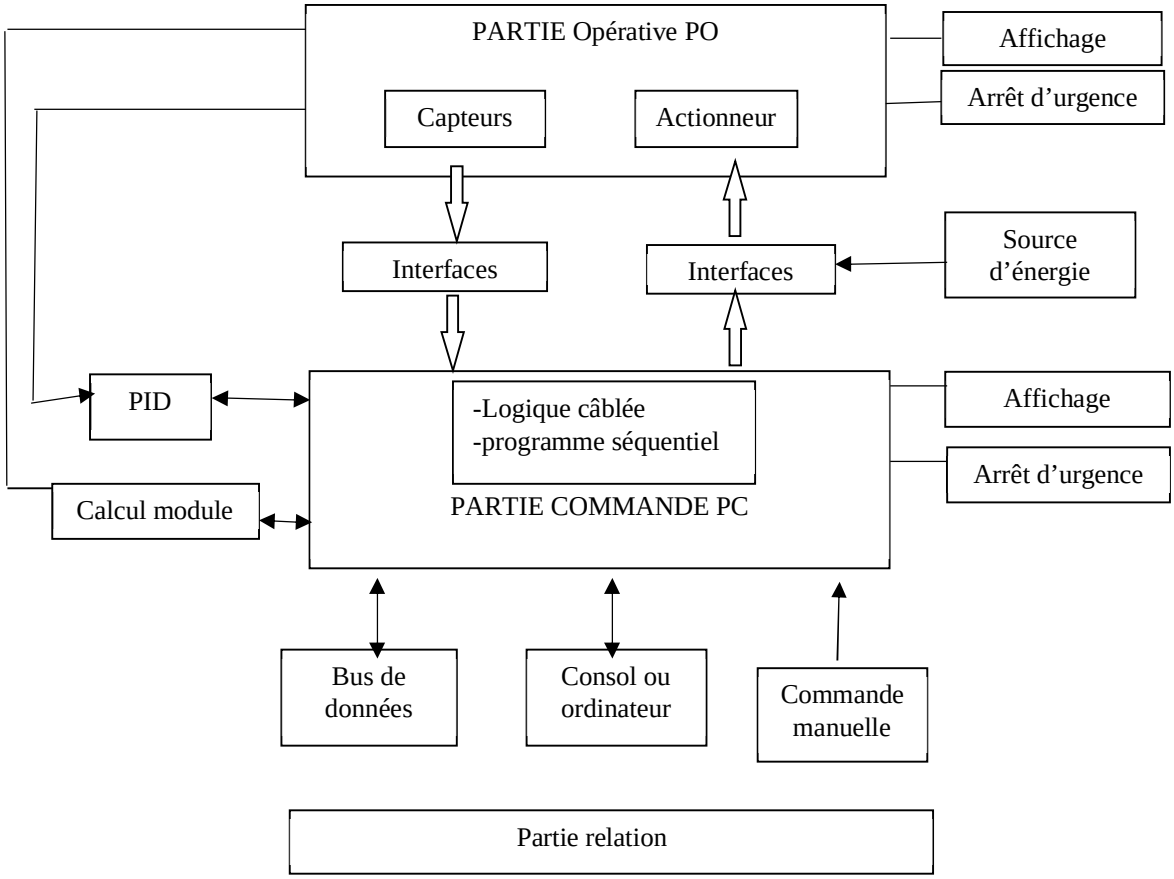

Figure I.2 Structure d'un système automatisé

I.3.3.1 Partie Opérative (PO) [5] :

- Elle exécute les ordres qu'elle reçoit de la partie commande grâce aux Actionneurs.
- Elle possède des Capteurs qui permettent de recueillir des informations.
- Elle reçoit des messages et envois des consignes vers la partie commande.

Elle comporte les éléments suivants :

1- Pré-actionneur : est un constituant dont le rôle est de distribuer, sur ordre de la partie commande, l'énergie utile aux actionneurs. Les pré-actionneurs les plus utilisés sont les contacteurs (pour les moteurs électriques) et les distributeurs (pour les vérins pneumatiques).

Distributeur pneumatique

Distributeur électropneumatique

Contacteur de puissance

Figure I.3 Les prés-actionneurs. [5]

2- Actionneur (moteur...) : Objet technique qui transforme l'énergie d'entrée qui lui est appliqué en une énergie de sortie (généralement mécanique) utilisable par un Effecteur pour fournir une action définie.

Moteur électrique

Moteur pneumatique

Vérin pneumatique

Figure I.4 Les actionneurs. [5]

3- Effecteur : qui agit sur la matière d'œuvre (pales de ventilateurs…), (tout organe en contact avec la matière d'œuvre).

4- Capteur : est un élément de prélèvement et de codage d'informations sur un processus ou sur l'environnement du système. Il convertit une grandeur physique (position, vitesse, …) en une information appelée compte-rendu et compréhensible par la Partie commande.

Capteur de pression industriel

Capteur de position

Détecteurs de fumée

Figure I.5 Les Capteurs. [5]

I.3.3.2 Partie Commande (PC) :

Elle joue le rôle ‘cerveau’ du système, elle pilote la partie opérative et reçoit des informations venant des capteurs.

La partie commande peut-être mécanique, électronique, ou autre. Sur de gros systèmes elle peut se composer de trois parties : un ordinateur, un logiciel et une interface.

I.3.3.3 Partie Relation (PR) ou pupitre de commande :

Le pupitre de commande permet à l’opérateur de commander le système (marche, arrêt, départ cycle …). Il permet également de visualiser les différents états du système à l’aide de voyants, et par le dialogue ou d’Interface Homme-Machine (IHM).

I.3.3.4 Interfaces :

Les interfaces assurent une compatibilité entre les signaux qui circulent entre la partie commande et la partie opérative. On en distingue deux types :

- Celles qui permettent un changement de niveau d’énergie : relais instantané, Contacteurs auxiliaires…
- Celles qui permettent un changement de type d’énergie : interface électro- pneumatiques, contacts à pression...etc.

I.4 Automates Programmables Industrielles (API) :

I.4.1 Historique :

Les Automates Programmables Industriels 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 nommée automates programmables. Ils n’étaient rentables que pour des installations d’une certaine complexité, mais la situation a changé très vite, ce qui a rendu les systèmes câblés obsolètes.

De nombreux modèles d'automates sont aujourd'hui disponibles ; depuis les nano automate bien adapté 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. [6]

I.4.2 Définition :

L'automate programmable industriel est un dispositif électronique programmable destiné à la commande de processus industriels par un traitement séquentiel des donner. Il envoie des ordres vers la partie puissante ou bien opérative (contient des actionneurs) à partir des pré-actionneurs en fonction des données d'entrée (capteur ou boutons poussoirs) de la partie commande ou Système de Contrôle / Commande (SCC), qui sont dirigés par un programme informatique. [7]

I.4.3 Structure d'un automate programmable industriel :

I.4.3.1 Description générale :

Un automate programmable industriel se présente sous la forme d'un ou plusieurs profilés supports (racks) dans lesquels viennent s'enficher les différents modules fonctionnels [8]:

- L'alimentation 110/220 VCA ou 24 VCC
- L'unité centrale de trainement à base de microprocesseur,
- Des cartes d'entrées/sorties logiques (TOR),
- Des cartes d'entrées/sorties analogiques (ANA),
- Des cartes de comptage rapide,
- Des cartes de communication (CP),
- Des cartes spécifiques pour : réseaux, asservissement, régulation commande d'axe….

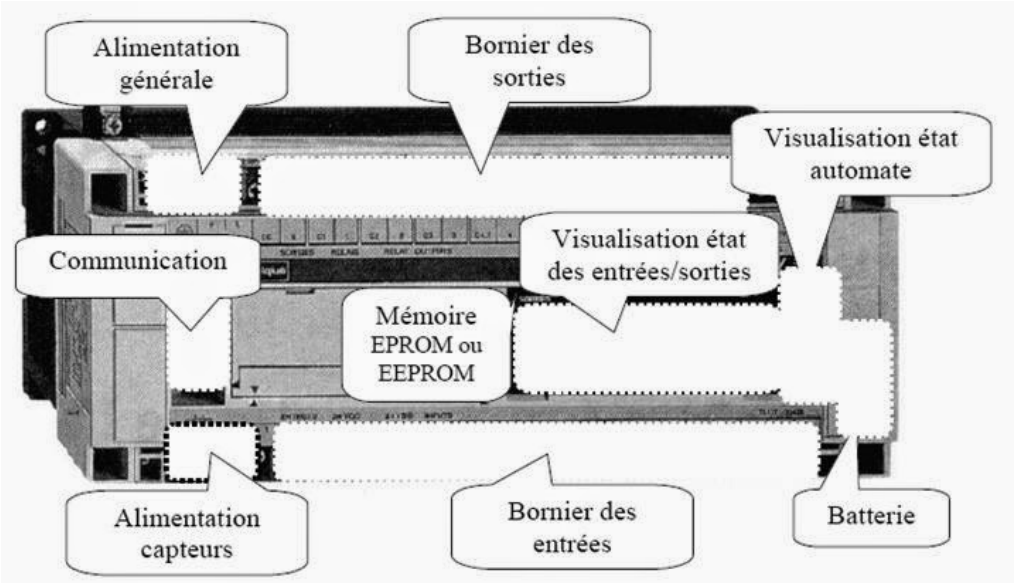

Figure I.6 Structure détaillée d'un API. [8]

I.4.3.2 La structure interne d'API :

La structure interne d'API se présente comme suit :

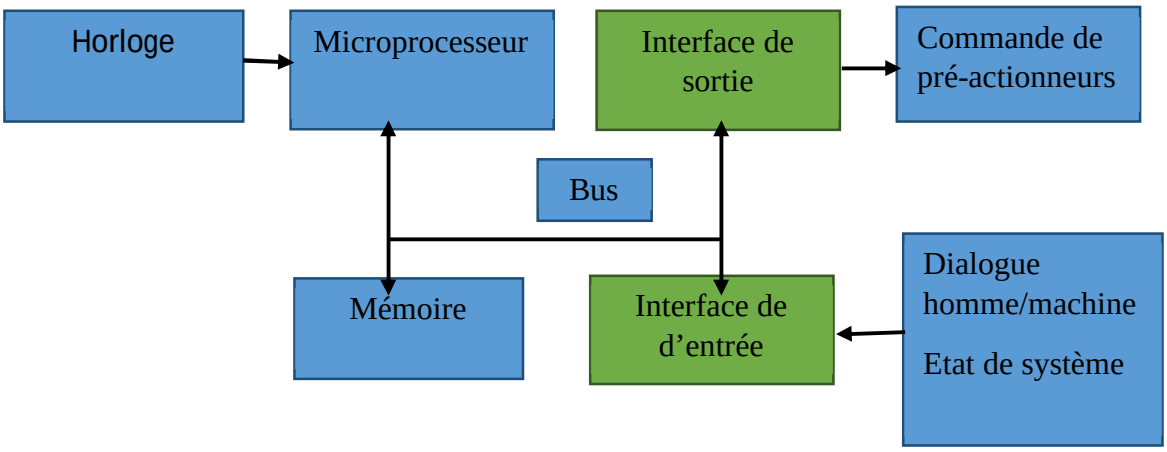

Figure I.7 Structure interne d'un API

L'automate programmable reçoit les informations relatives à l'état du système et puis commande les pré-actionneurs suivant le programme inscrit dans sa mémoire.

Un API donc se compose de trois grandes parties :

1- Le processeur [9] **:**

Le processeur a pour rôle principal le traitement des instructions qui constituent le programme de fonctionnement de l'application. Mais en dehors de cette tâche de base, il réalise également d'autres fonctions :

- Gestion des entrées/sorties.
- Surveillance et diagnostique de l'automate par une série de tests lancés à la mise sous tension ou cycliquement en cours de fonctionnement.
- Dialogue avec le terminal de programmation aussi bien pour l'écriture et la mise au point du programme qu'en cours d'exploitation pour des réglages ou des vérifications de données.

Le processeur est organisé autour d'un certain nombre de registres, ce sont des mémoires rapides permettant la manipulation des informations qu'elles retiennent, ou leur combinaison avec des informations extérieures.

Les principaux registres existants dans un processeur sont :

- **L'accumulateur :** C'est le registre où s'effectuent les opérations du jeu d'instruction, les résultats sont contenus dans ce registre spécial.

- **Le registre d'instruction :** Il reçoit l'instruction à exécuter et décode le code opération. Cette instruction est désignée par le pointeur.

- **Le registre d'adresse :** Ce registre reçoit, parallèlement au registre d'instruction, la partie opérande de l'instruction. Il désigne le chemin par lequel circulera l'information lorsque le registre d'instruction validera le sens et ordonnera le transfert.

- **Le registre d'état :** C'est un ensemble de positions binaires décrivant, à chaque instant, la situation dans laquelle se trouve précisément la machine.

2- Les piles [9] **:**

Une organisation spéciale de registres constitue une pile, ces mémoires sont utilisées pour contenir le résultat de chaque instruction après son exécution. Ce résultat sera utilisé ensuite par d'autres instructions, et cela pour faire place à la nouvelle information dans l'accumulateur.

3- Zone de mémoires [9] **:**

Un système de processeur est accompagné par un ou plusieurs types de mémoires. Elles permettent :

- De stocker le système d'exploitation dans des ROM ou PROM,
- Le programme dans des EEPROM,
- Les données système lors du fonctionnement dans des RAM. Cette dernière est généralement secourue par pile ou batterie. On peut, en règle générale, augmenter la capacité mémoire par adjonction de barrettes mémoires type PCMCIA.

4- Interface d'entrées/sorties [10] **:**

Ils assurent le rôle d'interface entre la CPU et le processus, en récupérant les informations sur l'état de ce dernier et en coordonnant les actions.

Plusieurs types de modules sont disponibles sur le marché selon l'utilisation souhaitée :

- **Modules TOR (Tout Ou Rien) :** l'information traitée ne peut prendre que deux états (vrai/faux, 0 ou 1 ...). C'est le type d'information délivrée par une cellule photoélectrique, un bouton poussoir ...etc.

- **Modules analogiques :** l'information traitée est continue et prend une valeur qui évolue dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (débitmètre, capteur de niveau, thermomètre...etc.).

- Modules spécialisés : l'information traitée est contenue dans des mots codes sous forme binaire ou bien hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent.

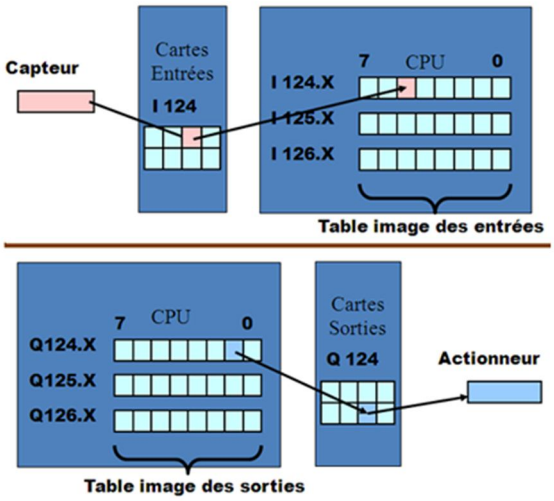

Figure I.8 Les interfaces d'entrées/sorties [10].

- Cartes d'entrées :

Elles sont destinées à recevoir l'information en provenance des capteurs et adapter le signal en le mettant en forme, en éliminant les parasites et en isolant électriquement l'unité de commande de la partie opérative.

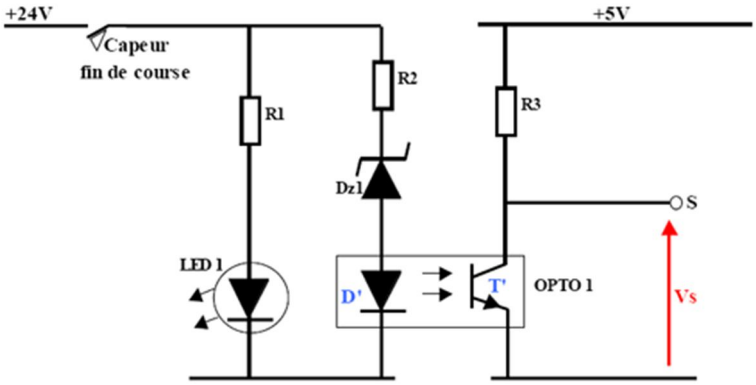

Figure I.9 Exemple d'une carte d'entrée typique d'un API. [10]

- Cartes de sorties :

Elles sont destinées à commander les pré-actionneurs et éléments des signalisations du système et adapter les niveaux de tensions de l'unité de commande à celle de la partie opérative du système en garantissant une isolation galvanique entre ces dernières

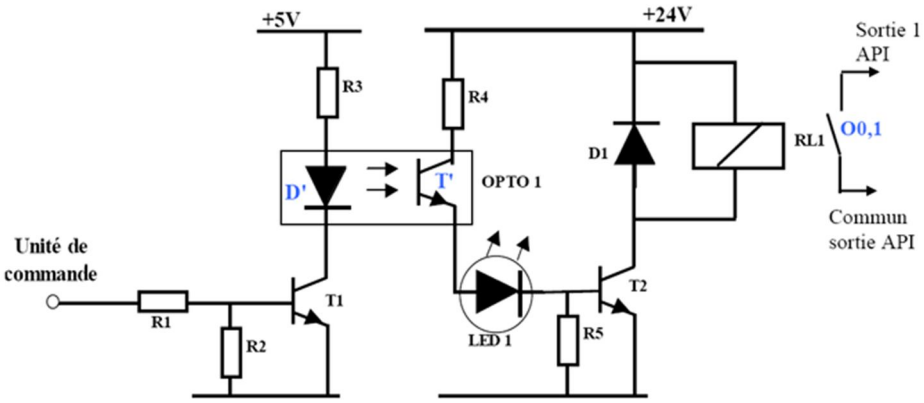

Figure I.10 Exemple d'une carte de sortie typique d'un API. [10]

I.4.4 Langages de programmation des API [11] :

Il existe plusieurs types de langages de programmation :

- **Grafcet :** Ce langage de programmation de haut niveau permet la programmation aisée de tous les procédés séquentiels.
- **Schéma FBD** : Le langage FBD permet de programmer graphiquement à l'aide de blocs, représentant des variables, des opérateurs ou des fonctions. Il permet de manipuler tous les types de variables.
- **Schéma Ladder :** Ce langage graphique est essentiellement dédié à la programmation d'équations booléennes.
- **Texte structuré :** Ce langage est un langage textuel de haut niveau. Il permet la programmation de tout type d'algorithme plus ou moins complexe.
- **Liste d'instructions :** Ce langage textuel de bas niveau est un langage à une instruction par ligne. Il peut être comparé au langage assembleur.

L'automate programmable industriel traduit le langage de programmation en langage compréhensible directement par le microprocesseur. Ce langage est propre à chaque constructeur, il est lié au matériel mis en œuvre.

Chaque instruction du programme est composée :

- de l'opération à effectuer (la nature de l'opération est codée 1 ou 0).
- de la variable sur laquelle l'opération va être effectué (variable de sortie, variable d'entrée, variable interne…)
- de la nature de la variable (binaire, numérique, texte, …)

Chaque instruction est écrite dans une partie de la mémoire appelée adresse.

I.4.5 Critères de choix d'un automate [12] :

Le choix d'un automate programmable est généralement basé sur :

- Nombre d'entrées/sorties : le nombre de cartes peut avoir une incidence sur le nombre de racks dès que le nombre d'entrées/sorties nécessaires devient élevé.
- Type de processeur : la taille mémoire, la vitesse de traitement et les fonctions spéciales offertes par le processeur permettront le choix dans la gamme souvent très étendue.
- Fonctions ou modules spéciaux : certaines cartes (commande d'axe, pesage ...) permettront de "soulager" le processeur et devront offrir les caractéristiques souhaitées (résolution, ...).
- Fonctions de communication : l'automate doit pouvoir communiquer avec les autres systèmes de commande (API, supervision ...) et offrir des possibilités de communication avec des standards normalisés (Profibus ...).

I.4.6 Domaines d'utilisation des automates programmables industriels :

Pour les raisons qui viennent d'être évoquées, les API s'adressent à des applications que l'on trouve dans la plupart des secteurs industriels. Ces machines fonctionnent dans les principaux secteurs et dans de l'enseignement. Parmi ces applications on trouve [13] :

- Mécanique et automobile.
- Industries chimiques
- Industries pétrolières
- Industries agricoles et alimentaires
- Transport et manutention

I.5 Les automatismes centralisés et décentralisés :

I.5.1 Les automatismes centralisés [14] :

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

- Gérer les demandes d'exécution et d'état de l'automatisme (image des entrées).
- Elaborer les 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 (voir figure I.11) géraient tout un ensemble de fonctions qui n'avaient pas forcement 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, citions :

- 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.

Un exemple d'une structure d'automatismes centralisés est illustré par la figure suivante :

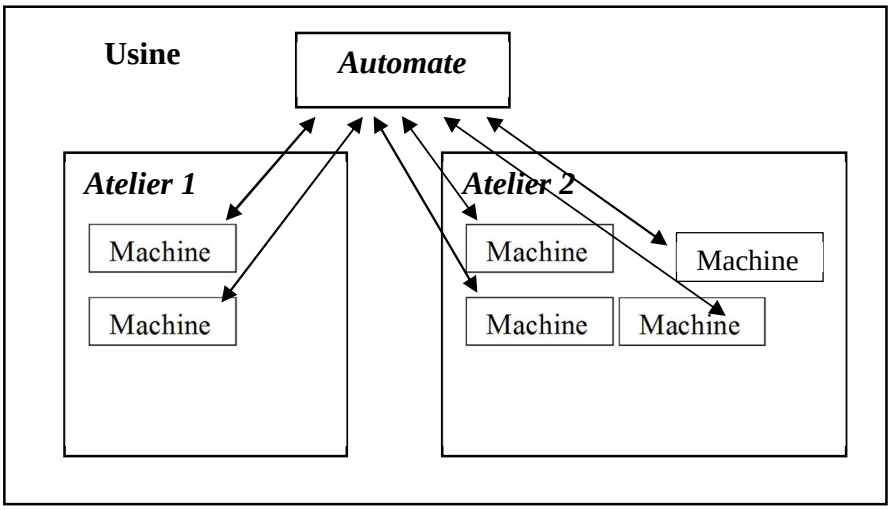

Figure I.11 Les automatismes centralisés.

I.5.2 Les automatismes décentralisés [14] :

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

La figure montre un exemple d'une structure des automates décentralisés.

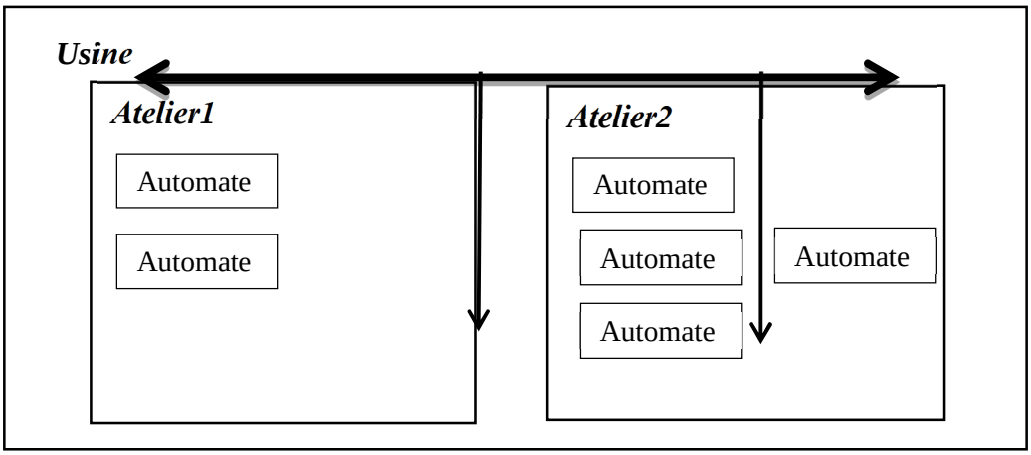

Figure I.12 Les automatismes décentralisés

I.6 Conclusion :

L'objectif du premier chapitre est d'avoir une idée sur l'industrie 4.0 et les systèmes automatisés et les notions de bases sur les API. Nous avons notamment mis en évidence le rôle de l'automatisme qui a pour objectif d'améliorer la productivité, la qualité, la sécurité et tout variable qui influence positivement à l'entreprise.

II.1 Introduction :

Dans ce chapitre, nous allons présenter l'utilité de la Plateforme Virtuelle d'Expérimentation Robotique (V-REP), ensuite la plateforme d'assemblage robotisée qu'on veut concevoir dans notre projet et la complexité de sa réalisation sont expliquées. Enfin, la nécessité d'utiliser les techniques Hardware-In-The-Loop (HIL)pour sa validation avant de passer à l'implémentation réelle est démontrée.

II.2 La Plateforme Virtuelle d'Expérimentation Robotique V-REP :

II.2.1 Description :

La plateforme V-REP est un simulateur de robot 3D basé sur une architecture de contrôle distribuée : les programmes de contrôle (ou scripts) peuvent être directement attachés aux objets de la scène et s'exécuter simultanément de manière filetée ou non filetée. Cela rend le V-REP très polyvalent et idéal pour les applications multirobots, et permet aux utilisateurs de modéliser des systèmes robotiques de la même manière qu'en réalité - où le contrôle est la plupart du temps également distribué. V-REP vous permet d'éditer et de simuler des systèmes robotiques complets ou des sous-systèmes (capteurs, mécanismes, etc.). Il offre une multitude de fonctionnalités qui peuvent être facilement intégrées et combinées grâce à une API et une fonctionnalité de script exhaustive. [15]

Le logiciel V-REP simule un grand nombre de lois physiques pour se rapprocher de la réalité. Voici une vue typique de l'application V-REP.

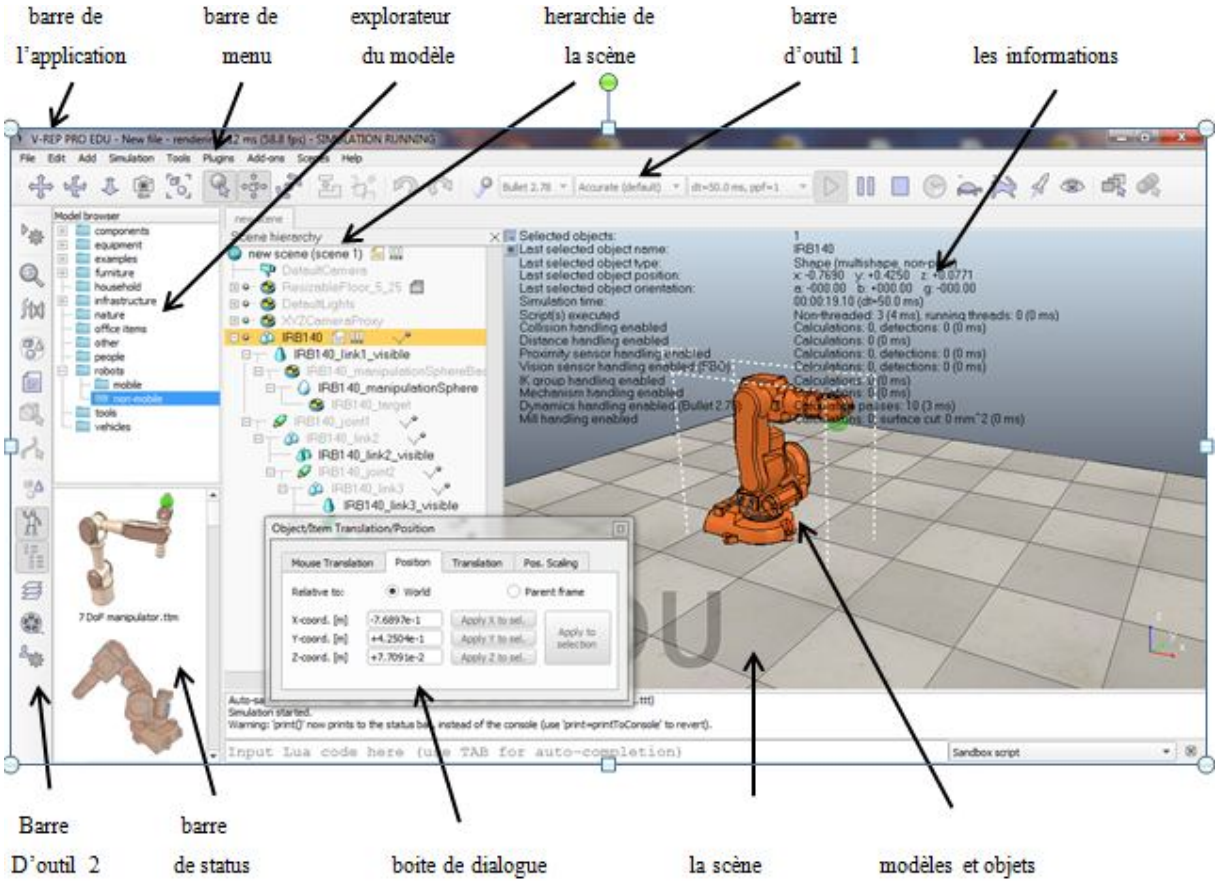

Figure II.1 L'application V-REP

L'application V-REP est composée de plusieurs éléments. Ses principaux éléments sont :

Une fenêtre d'application : la fenêtre d'application est la fenêtre principale de l'application. Il est utilisé pour afficher, éditer, simuler et interagir avec une scène.

Boîtes de dialogue : à côté de la fenêtre de l'application, l'utilisateur peut également modifier et interagir avec une scène en ajustant les paramètres ou les paramètres de la boîte de dialogue. Chaque boîte de dialogue regroupe un ensemble de fonctions connexes ou de fonctions qui s'appliquent à un même objet cible.

Barre d'application : la barre d'application indique le type de licence de votre copie V-REP, le nom de fichier de la scène actuellement affichée, le temps utilisé pour une passe de rendu (une passe d'affichage) et l'état actuel du simulateur (état de simulation ou type du mode d'édition actif).

Barre de menu : la barre de menu permet d'accéder à presque toutes les fonctionnalités du simulateur. La plupart du temps, les éléments de la barre de menu activent une boîte de

dialogue. Le contenu de la barre de menu est sensible au contexte (c'est-à-dire qu'il dépend de l'état actuel du simulateur).

Barres d'outils : les barres d'outils présentent des fonctions auxquelles on accède souvent (par exemple, changer le mode de navigation, sélectionner une autre page, etc.). Certaines fonctions de la barre d'outils 1 et toutes les fonctions de la barre d'outils 2 sont également accessibles via la barre de menus ou le menu contextuel.

Modèle de navigateur : le navigateur de modèle est visible par défaut, mais peut être basculé avec son bouton de barre d'outils correspondant. Il affiche dans sa partie supérieure une structure de dossier de modèle V-REP, et dans sa partie inférieure, les vignettes des modèles contenus dans le dossier sélectionné.

Hiérarchie des scènes : la hiérarchie des scènes est visible par défaut, mais peut être basculée avec le bouton correspondant de la barre d'outils. Il affiche le contenu d'une scène (c'est-à-dire tous les objets de scène composant une scène). Depuis les objets de scène sont construits dans une structure de type hiérarchique, la hiérarchie des scènes affiche un arbre de cette hiérarchie, et les éléments individuels peuvent être développés ou réduits. Un double clic sur une icône ouvre / ferme une boîte de dialogue de propriétés liée à l'icône cliquée. Un double-clique sur un nom d'objet permet de l'éditer.

Texte d'information : le texte d'information affiche des informations relatives à la sélection actuelle d'un objet/élément et à l'exécution d'états ou de paramètres de simulation. L'affichage de texte peut être basculé avec l'un des deux petits boutons en haut à gauche d'une page. L'autre bouton peut être utilisé pour basculer un fond blanc, donnant un meilleur contraste en fonction de la couleur de fond d'une scène.

Barre d'état : la barre d'état affiche les informations relatives aux opérations effectuées, aux commandes et affiche également les messages d'erreur de l'interpréteur « Lua ».

II.2.2 L'utilisation de logiciel V-REP :

Le logiciel V-REP peut être utilisé pour la surveillance à distance, pour le contrôle du matériel, pour le prototypage et la vérification rapides, pour le développement rapide d'algorithmes/ajustement de paramètres, pour le double contrôle de sécurité, pour la formation en robotique, pour les simulations d'automatisation industrielle, etc.

II.2.3 Le langage de programmation de V-REP :

Le langage de script sur la plateforme V-REP est « Lua », qui est un langage de programmation par extension conçu pour prendre en charge la programmation procédurale générale.

II.3 Fonctionnement de la cellule :

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 lorsque celle-ci 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.

II.3.1 Présentation d'un modèle de la cellule et son fonctionnement :

La cellule robotisée (voir figure II.2) va contenir un convoyeur circulaire qui contient quatre palettes et quatre capteurs photoélectriques et autour de lui il y a quatre postes de travail qui contiennent des robots. Dans chaque poste, il se trouve un stock de pièces chargées sur deux autres convoyeurs contiennent aussi des capteurs et un robot qui placera la pièce sur la palette après l'avoir prise du convoyeur.

Dans cette application, on veut réaliser un fonctionnement automatisé de cette plateforme d'assemblage. En appuyant sur un bouton poussoir qui mettra le convoyeur circulaire 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 prendre la pièce et la mettre sur son espace du travail où il fera sa tâche d'assemblage.

Lorsque l'assemblage (pièce/palette) est fait. Un signal est donné par le capteur pour que le convoyeur circulaire marche de nouveau.

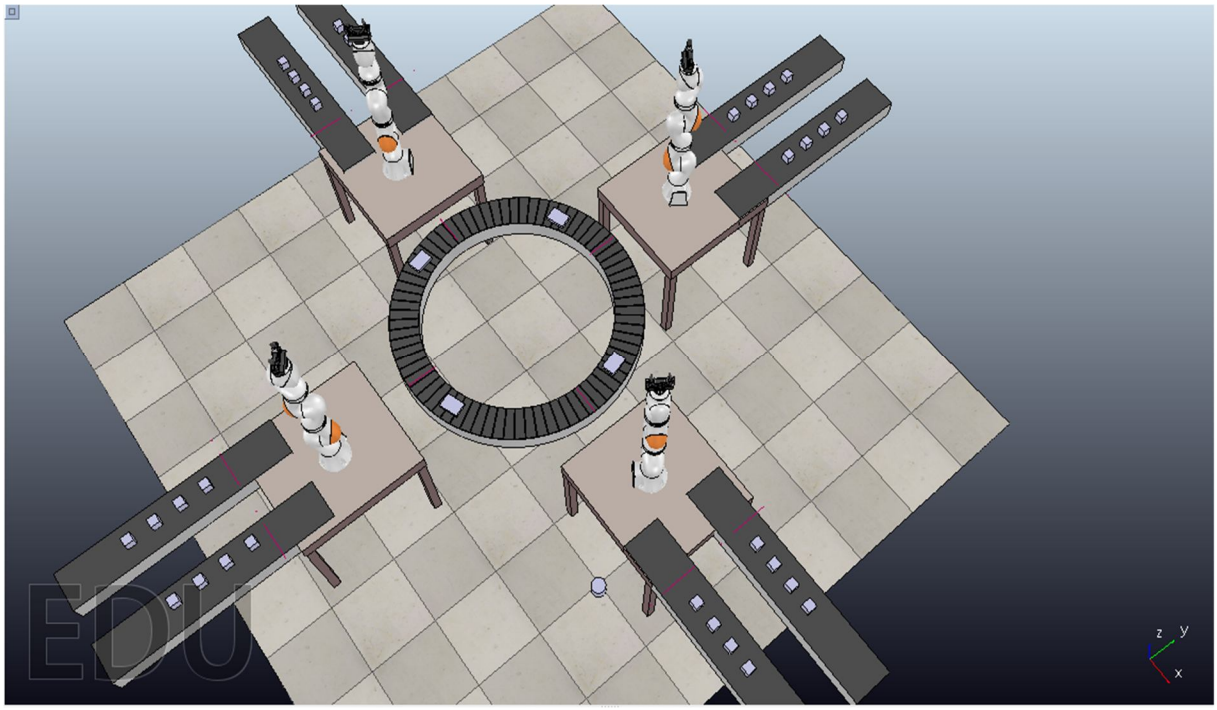

Figure II.2 Un model représentant de la cellule

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 pièce du convoyeur simple par le robot et accomplissement de son assemblage avec la palette.
- Redémarrage du convoyeur circulaire après avoir reçu un signal donné par le capteur.
- Pour que le robot puisse faire sa tâche, le bras manipulateur descendra à un point précis puis il marchera quelque pas jusqu'à l'arrivé à la pièce pour la tenir avec sa pince (gripper). Il retournera au point de descendre et il tournera de 180° après il déplacera directement pour mettre la pièce à sa place.

II.4 Les techniques de validation en environnement émulé (Hardware-In-The Loop) :

En général, la conception des systèmes automatisés nécessite d'abord une analyse puis des tests de validation pour passer 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 environnements, mais commençons d'abord par expliquer comment et pourquoi les utiliser ?

II.4.1 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. [16]

II.4.2 L'intérêt de la simulation HIL [16] :

- Elle est considérée comme un outil très puissant car elle permet de tester une partie physique du système dans un environnement virtuel complètement contrôlé.
- Elle permet aussi de tester des éléments en simulation dont les modèles ne paraissent pas assez précis pour subvenir au besoin de la simulation.
- Il est possible d'effectuer des essais sur les effets de défauts et pannes des actionneurs, capteurs et ordinateurs sur l'ensemble du système.
- Elle permet l'économisassions de temps et d'argents dans le processus de développement.
- Enfin, selon la complexité des cas et du matériel disponible, réaliser plusieurs émulations peut faciliter différentes validations partielles permettant une validation finale rapide.

II.4.3 Utilisation de HILS (Hardware In The Loop Simulation):

Le HILS est devenue une technique indispensable pour la vérification et la validation des systèmes dans diverses disciplines, telles que, l'automobile, l'aéronautique et de l'environnement industriel.

Dans le contexte de l'automobile : par exemple, le HILS permet aux développeurs de valider de nouvelles solutions matérielles et logicielles automobiles, en respectant les exigences de qualité et de délais de mise sur le marché des restrictions. Dans un simulateur typique HIL, la dynamique du moteur est émulée à partir de modèles mathématiques et exécutés par un processeur en temps réel dédié. En outre, une unité d'E / S permet de connecter des capteurs et actionneurs du véhicule (qui présente en général un haut degré de non-linéarité). Enfin, l'unité de contrôle électronique (ECU) en cours de test est connectée au système et stimulé par une série de manœuvres de véhicules exécutées par le simulateur.

Dans l'électronique de puissance : la simulation HIL utilisée pour réduire le cycle de développement augmente l'efficacité et améliore la fiabilité et la sécurité de ces systèmes pour un grand nombre d'applications. Dans l'industrie, l'approche HIL présente de nombreux avantages, car la simulation est réalisée avec le programme réel en cours d'exécution dans le contrôleur réel. Cela peut être très important, surtout lorsque différentes échelles de temps sont considérées d'un point de vue réel, principalement en ce qui concerne l'évolution réelle du contrôleur et de son calendrier, en cours d'exécution dans des conditions réelles de travail. Dans ce contexte, le modèle du système de fabrication est en cours d'exécution dans le PC et contrôlé par PLC réel.

II.4.4 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.

La partie sur laquelle on se focalise est la partie des contrôleurs (les PLC). On veut tester les contrôleurs dans des situations qui reflètent la réalité et cela sans utiliser le matériel réel ou le système physique (l'installation) dont on ne dispose pas de la majorité des éléments constituants à l'heure actuelle. Mais cela est toujours possible grâce aux techniques de validation en environnement émulé. Il nous suffit alors de se disposer de ces contrôleurs et d'un logiciel de simulation sur lequel on peut concevoir un modèle représentant 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 en choisir un, qui est dédié aux systèmes de production et d'assemblage, qui dispose des éléments constituants de la cellule dans sa bibliothèque et bien sûr qui a la capacité de communiquer en temps réel avec d'autres logiciels et matériels.

Notre réalisation a été réalisée au CDTA (Centre de Développement de Technologie Avancée). Donc on a choisi un logiciel qui répond aux exigences qu'on avait précisées auparavant, il s'agit du logiciel de simulation 3D nommé V-REP (Virtual Robot Experimentation Platform) qu'on a déjà présenté. C'est lui le logiciel dont on a besoin par la suite.

II.5 La simulation :

Avant de passer à la programmation du V-REP, on explique d'abord c'est quoi une simulation, et à quoi elle sert ?

II.5.1 Définition de la simulation :

La simulation est une technique de modélisation largement utilisée dans l'évaluation de performances des systèmes informatiques et réseaux de communication. Il s'agit d'implanter un modèle simple du système à l'aide d'un programme de simulation adéquat. Cela permet de modéliser des situations très complexes que l'on ne peut résoudre analytiquement. De plus, le comportement transitoire des systèmes peut être évalué alors que les modèles analytiques sont généralement utilisés pour étudier le comportement stationnaire d'un système. [17]

II.5.2 Domaines d'application de la simulation :

Les applications de la simulation sont incomptables. 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.5.3 Les avantages et les inconvénients de la simulation [18] :

Avantages :

- Observations des états du système.
- Etudes des points de fonctionnement d'un système.
- Etudes de l'impact des variables sur les performances du système.
- Etude d'un système sans les contraintes matérielles.

Inconvénient :

- La conception de modèles peut nécessiter des compétences spéciales.
- Résultats pas forcément généralisable.

- 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 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.

II.6 La mise en place et la programmation de la plateforme simulée :

Après avoir choisi le logiciel simulateur V-REP pour la conception de la plateforme, on passera par la suite à la mise en place.

On va suivre les étapes suivantes :

Importer les éléments nécessaires du logiciel V-REP

1. Capteurs :

Un capteur est un dispositif permettant de détecter un phénomène physique sous la forme d'un signal, généralement électrique. Le capteur se différencie du détecteur et du senseur par sa possibilité de délivrer une grandeur physique directement utilisable pour une mesure ou une commande.

Le type de capteurs qu'on a utilisé, ce sont des détecteurs photoélectriques, ils permettent la détection d'objets de toutes natures.

2. Convoyeurs :

On a utilisé un convoyeur cercle pour le transport des palettes et huit autres convoyeurs simples d'une taille et de forme réglable pour les pièces ou on a inséré des capteurs qui font le rôle de la détection photoélectrique.

La figure ci-dessous représente le programme de convoyeur simple.

```
function sysCall_init()
    forwarder=sim.getObjectHandle('ConveyorBelt_forwarder')
    proximity_sensor3=sim.getObjectHandle('sensor1#3')
end

function sysCall_cleanup()

end

function sysCall_actuation()
    beltVelocity=sim.getScriptSimulationParameter(sim.handle_self,"conveyorBeltVelocity")

psensor3,dis,pt,detectOH=sim.readProximitySensor(proximity_sensor3)

if(psensor3 > 0) then
      beltVelocity=0
       sim.setIntegerSignal('b',detectOH)

     end
```

Figure II.3 Le programme du simple convoyeur.

La figure ci-dessous représente le programme de convoyeur circulaire.

```
function sysCall_init()
    pathHandle=sim.getObjectHandle("CircularConveyorPath")
    sim.setPathTargetNominalVelocity(pathHandle,0) -- for backward compatibility

 proximity_sensor1=sim.getObjectHandle('sensor1')
 proximity_sensor2=sim.getObjectHandle('sensor2')
 proximity_sensor3=sim.getObjectHandle('sensor3')
 proximity_sensor4=sim.getObjectHandle('sensor4')

end

function sysCall_actuation()
         beltVelocity=sim.getScriptSimulationParameter(sim.handle_self,"circularConveyorBeltVelocity")
         psensor1=sim.readProximitySensor(proximity_sensor1)

 if(psensor1 > 0) then
      beltVelocity=0

         end
         psensor2=sim.readProximitySensor(proximity_sensor2)

if(psensor2 > 0) then
      beltVelocity=0
           end
psensor3=sim.readProximitySensor(proximity_sensor3)

if(psensor3 > 0) then
      beltVelocity=0
           end
psensor4=sim.readProximitySensor(proximity_sensor4)

if(psensor4 > 0) then
      beltVelocity=0
           end

local dt=sim.getSimulationTimeStep()
```

Figure II.4 Le programme du simple convoyeur circulaire.

Les instructions qu'on a utilisées pour les programmes de convoyeur simple et circulaire à base de script Lua sont :

- Sim.getScriptSimulationParameter : Récupère le paramètre d'un script principal ou d'un script enfant de sa liste de paramètres de simulation.
- Sim.getObjectHandle : Récupère un descripteur d'objet en fonction de son nom.
- Sim.readProximitysensor : Lit l'état d'un capteur de proximité.
- Sim.getIntegerSignel : Obtient la valeur d'un signal entier.

3. Robots industriels :

Un robot est une machine capable d'effectuer des taches et de manipuler des objets selon un programme de façon automatique. Ils sont généralement utilisés pour remplacer les humains dans des situations ou ces derniers sont incapables d'effectuer le travail, des situations plus dangereuses, de haute précision ou répétitives.

Les robots sont de plus en plus utilisés en industries, car ils sont très rapides, et précis

Les robots qu'on va utiliser est le LBR IIWA (Leicht Bau Roboter Intellegent Industrial Work Assistant)7 R800 de KUKA à 6 degrés de liberté.

Le LBR IIWA est le premier robot sensitif et donc apte à la collaboration homme-robot (homme et machine travaillent main dans la main). L'homme commande et surveille la production, le robot se charge des tâches corporelles fatigantes. Les deux apportent des compétences spécifiques, c'est un principe décisif pour Industrie 4.0 fabriqué en série. LBR signifie « Leicht Bau Roboter » (robot léger), IIWA signifie « Intelligent Industrial Work Assistant » (assistance intelligent de travail industriel). La base pour des processus de production nouveaux et sûrs pour l'avenir et pour plus de rentabilité et une efficacité plus élevée sont bien assurés par ce robot. [19]

La figure ci-dessous représente le programme de la pince du robot

```
function sysCall_actuation()
closing=sim.getIntegerSignal('c')
p1=sim.getJointPosition(j1)
p2=sim.getJointPosition(j2)
if (closing==1) then
    if (p1<p2-0.008) then
        sim.setJointTargetVelocity(j1,-0.01)
        sim.setJointTargetVelocity(j2,-0.04)
    else
        sim.setJointTargetVelocity(j1,-0.04)
        sim.setJointTargetVelocity(j2,-0.04)
    end
else
    if (p1<p2) then
        sim.setJointTargetVelocity(j1,0.04)
        sim.setJointTargetVelocity(j2,0.02)
    else
        sim.setJointTargetVelocity(j1,0.02)
        sim.setJointTargetVelocity(j2,0.04)
    end
end
```

Figure II.5 Programme de la pince de robot sur V-REP.

Les instructions qu'on a utilisé pour le programme de la pince sont :

- Sim.getJointPosition : Récupère la position intrinsèque d'un joint.
- Sim.setJointTargetVelocity : Définit la vitesse cible intrinsèque d'une articulation non sphérique.

La figure si dessous représente le programme de position de robot réel.

```
initialeprise={-69.31*math.pi/180,27.6*math.pi/180,10.1*math.pi/180,-111.13*math.pi/180,-6.48*math.pi/180,41.56*math.pi/180,0}
Prise1P1={-89.06*math.pi/180,52.34*math.pi/180,10*math.pi/180,-66.23*math.pi/180,-8.51*math.pi/180,62.01*math.pi/180,-30*math.pi/
Prise1P2={-89.08*math.pi/180,60.03*math.pi/180,10.01*math.pi/180,-66.6*math.pi/180,-10.18*math.pi/180,54.1*math.pi/180,-25*math.p
Prise2P1={-48.47*math.pi/180,43.68*math.pi/180,10.1*math.pi/180,-82.67*math.pi/180,-8.25*math.pi/180,54*math.pi/180,25*math.pi/18
Prise2P2={-48.46*math.pi/180,52.15*math.pi/180,10*math.pi/180,-83.1*math.pi/180,-11*math.pi/180,45.29*math.pi/180,25*math.pi/180}
InitialeDepot={115.11*math.pi/180,27.6*math.pi/180,10.1*math.pi/180,-111.13*math.pi/180,-6.48*math.pi/180,41.56*math.pi/180,0}
Depot1P1={106.31*math.pi/180,55*math.pi/180,10*math.pi/180,-61.34*math.pi/180,-8.8*math.pi/180,64*math.pi/180,0}
Depot1P2={106.29*math.pi/180,57.31*math.pi/180,10*math.pi/180,-61.78*math.pi/180,-9.28*math.pi/180,61.22*math.pi/180,0}
Depot2P1={107.88*math.pi/180,37.39*math.pi/180,10*math.pi/180,-94.12*math.pi/180,-7.74*math.pi/180,48.74*math.pi/180,0}
Depot2P2={107.88*math.pi/180,40.38*math.pi/180,10*math.pi/180,-94.57*math.pi/180,-8.73*math.pi/180,45.40*math.pi/180,0}
Depot3P1={117.1*math.pi/180,52.66*math.pi/180,10*math.pi/180,-65.83*math.pi/180,-8.81*math.pi/180,61.68*math.pi/180,0}
Depot3P2={117.07*math.pi/180,55*math.pi/180,10*math.pi/180,-66.32*math.pi/180,-9.35*math.pi/180,58.89*math.pi/180,0}
Depot4P1={121.9*math.pi/180,36.87*math.pi/180,10*math.pi/180,-95.06*math.pi/180,-7.87*math.pi/180,48.19*math.pi/180,0}
Depot4P2={121.9*math.pi/180,39.88*math.pi/180,10*math.pi/180,-95.51*math.pi/180,-8.91*math.pi/180,44.81*math.pi/180,0}
Repos={0,0,0,0,0,0,0}
sim.rmlMoveToJointPositions(jointHandles,-1,currentVel,currentAccel,maxVel,maxAccel,maxJerk,initialeprise,targetVel)
```

Figure II.6 Programme de position de robot.

La figure si dessous représente la tâche du robot.

```
if(var ==1)then
if(signal1>0) then
sim.rmlMoveToJointPositions(jointHandles,-1,currentVel,currentAccel,maxVel,maxAccel,maxJerk,Prise1P1,targetVel)
sim.rmlMoveToJointPositions(jointHandles,-1,currentVel,currentAccel,maxVel,maxAccel,maxJerk,Prise1P2,targetVel)
sim.wait(5)
sim.setIntegerSignal('c',1)
sim.setObjectParent(cube1[j],connector,false)
sim.rmlMoveToJointPositions(jointHandles,-1,currentVel,currentAccel,maxVel,maxAccel,maxJerk,Prise1P1,targetVel)
sim.rmlMoveToJointPositions(jointHandles,-1,currentVel,currentAccel,maxVel,maxAccel,maxJerk,initialeprise,targetVel)
sim.rmlMoveToJointPositions(jointHandles,-1,currentVel,currentAccel,maxVel,maxAccel,maxJerk,InitialeDepot,targetVel)
sim.rmlMoveToJointPositions(jointHandles,-1,currentVel,currentAccel,maxVel,maxAccel,maxJerk,Depot1P1,targetVel)
sim.rmlMoveToJointPositions(jointHandles,-1,currentVel,currentAccel,maxVel,maxAccel,maxJerk,Depot1P2,targetVel)
sim.setIntegerSignal('c',0)
sim.setObjectParent(cube1[j],-1,false)
j=j+1
sim.wait(5)
sim.rmlMoveToJointPositions(jointHandles,-1,currentVel,currentAccel,maxVel,maxAccel,maxJerk,Depot1P1,targetVel)
sim.rmlMoveToJointPositions(jointHandles,-1,currentVel,currentAccel,maxVel,maxAccel,maxJerk,InitialeDepot,targetVel)
sim.rmlMoveToJointPositions(jointHandles,-1,currentVel,currentAccel,maxVel,maxAccel,maxJerk,initialeprise,targetVel)
end
client:send("F")
sim.setIntegerSignal('P1',1)
end
```

Figure II.7 Le programme de la tâche du robot.

Les instructions qu'on a utilisées pour le programme de taches de robot sont :

- Sim.setThreadIsFree : Les threads créés par V-REP ne s'exécutent jamais simultanément, ils se comportent plutôt comme des coroutines.
- Sim.rmlMoveJointPositions : Déplace (actionne) plusieurs articulations en même temps.
- Sim.getObjectParent : Récupère le handle de l'objet parent d'un objet.
- Sim.wait : Attend un certain temps.

Le problème qui se pose 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.7 La communication socket [20] :

Le logiciel V-REP 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 ?

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 (Transmission Control Protocol). 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 (Protocole de datagramme utilisateur.). Ce mode nécessite l'adresse de destination à chaque envoi, et aucun accusé de réception n'est donné.

II.8 Conclusion :

Dans ce chapitre nous avons présenté le logiciel V-REP qui a pour but de réaliser la plateforme virtuelle d'expérimentation. Nous avons aussi développé bien l'idée de notre projet qui consiste essentiellement à créer une plateforme virtuelle sur V_REP et la commander par un PLC en expliquant le fonctionnement qu'on veut établir dans la cellule d'assemblage robotisée. Dans le chapitre qui se suit on s'intéressera à la partie physique du système qui est le PLC utilisé.

III.1 Introduction :

Dans ce chapitre, on présentera la partie matérielle (physique) de système étudié, cette partie constituant des contrôleurs utilisés pour le pilotage de la cellule S7-1500 CPU 1515 et leur fonction avancée, en plus on présentera aussi, le logiciel de programmation associé TIA PORTAL V15 utilisé dans le cadre de ce projet.

III.2 Objectifs du projet :

Le but principal de ce projet de fin d'étude est la réalisation d'un programme via " TIA Portal V15 " qui est le dernier logiciel d'ingénierie de SIEMENS pour l'automatisation d'une cellule robotisée, puis tester ce programme dans un automate programmable industriel S7-1500, afin de vérifier son bon fonctionnement.

III.3 Automate Siemens S7-1500 :

L'automate Siemens S7-1500 sorti le 27 novembre 2012 est un contrôleur pour les machines de moyenne et haute de gamme. Cette nouvelle génération de contrôleurs est caractérisée par une haute performance avec une grande efficacité. Il comporte une multitude de fonctions intégrées en standard, y compris le Motion Control, les fonctions sécurité pour garantir une sécurité maximale en production et en développement. Les fonctions de diagnostics configurables permettent de superviser l'état de l'installation, son intégration dans portail TIA permet de concevoir simplement des projets en optimisant les coûts de développement. [21]

Figure III. 1 Automate Siemens S7-1500. [21]

Initialement, le portefeuille d'automates comprendra les trois types de CPU 1511, 1513 et 1516 pour la gamme de la puissance moyenne, chaque CPU est également disponible en version F (sécurité) pour les applications de sécurité, avec des caractéristiques de performance graduées.

Celles-ci diffèrent, par exemple, du nombre d'interfaces, performances de bit et la taille de la mémoire d'affichage et des données. Selon les tâches d'automatisation à effectuer, la CPU dans la configuration centrale peut être complété par 32 modules supplémentaires, par exemple avec les nouveaux modules de communication ou des modules IO.

Simatic S7-1500 a été orientée vers la performance et l'efficacité. En ce qui concerne la performance globale, la technologie, la sécurité et les performances du système ont été considérablement améliorées. Afin d'accroître l'efficacité, de nouveaux développements ont été réalisés spécifiquement dans les domaines de la conception et de la gestion, le diagnostic du système et le logiciel l'ingénierie portail TIA Portal. Le Simatic S7-1500 a un bus de fond de panier avec un temps de réponse inférieur à 500ms.

III.3.1 Les différents composants de notre automate S7-1500 [21] :

- **L'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**.**
- **Unité Central CPU 1515-2PN :** La CPU 1515-2PN version 2.1est dotée d'une mémoire de programme et de données moyennes qui convient pour des applications contenant non seulement une périphérie centralisée, mais aussi PROFINET IO qui sont des structures d'automatisation décentralisées.

Figure III.2 CPU 1515-2PN. [21]

La CPU 1515-2PN contient deux interfaces qui sont :

- L'interface intégrée : qui se présente sous forme d'un commutateur à 2 ports.
- L'interface PROFINET intégrée : qui est une interface supplémentaire avec adresse IP séparée qui peut être utilisée par exemple pour la séparation des réseaux, pour le raccordement d'autres périphériques PROFINET 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.

- **Les modules d'entrés sorties :** Les modules d'entrées/sorties TOR (Tout ou Rien) 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 tels que :
  - **Module d'entrées :** Il permet à l'automate de recevoir des informations provenant des capteurs (Tout ou Rien).
  - **Module de sortie :** Le module de sorties assure le raccordement de l'automate aux différents actionneurs et pré-actionneurs tels que les moteurs et les relais.

L'emplacement sur le châssis de l'automate S7-1500 de la CPU et des différents modules d'entrées/sorties est présenté par la figure suivante :

Figure III.3 Constitution de l'Automate S7-1500

III.4 Automate ET-200SP [22] :

L'automate ET-200SPest un système d'E/S décentralisé qui permet un raccordement facile à un automate central via la liaison PROFINET**.**

SIMATIC ET-200SP est un système d'E/S décentralisées multifonctions qui convient pour différents champs d'application. Ce système est approuvé pour un degré de protection IP 20 et conçu pour être utilisé dans une armoire de commande. La figure ci-dessous montre l'automate ET-200SP :

Figure III.4 L'automate ET-200SP. [22]

III.4.1 Les différents composants de notre automate ET-200SP [22] :

L'alimentation

Pour une alimentation performante de l'installation, une entrée d'alimentation stabilisée est ajoutée à l'armoire, il s'agit de SITOP PSU100S (DC 24V/10A). (Figure III.5).

Figure III.5 L'alimentation SITOP PSU100S [22]

Les modules d'entrées sorties

- Le module d'entrer numérique.
- Le module de sortie numérique.

Unité central CPU 1512-SP1PN

La CPU 1512SP-1 PN est dotée d'une mémoire de programme et de données. Elle possède notamment une interface PRFINET qui a trois ports : le port 1 et le port 2 se trouvent sur le BusAdapter (Adaptateur de bus) et le port 3 se trouve sur la CPU. On peut aussi raccorder ce système d'E/S à une PG / un PC ou un appareil HMI via le port 3.

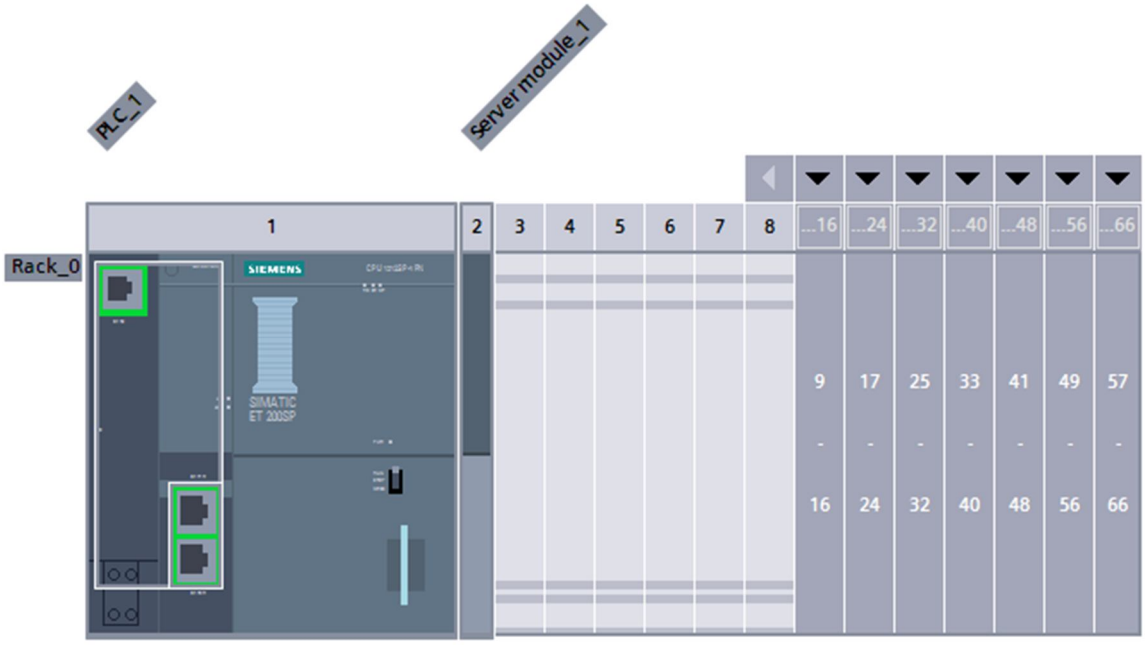

Figure III.6 Constitution d'un automate ET-200SP.

III.5 Interface Homme-Machine « IHM » SIMATIC IFP2200 [20] :

L'interface homme-machine (IHM) est l'interface utilisateur qui relie l'opérateur au dispositif de commande d'un système industriel. 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 Display Port ou DVI-D au PC industriel, qui peut être éloigné de 30 mètres. La figure ci-dessous indique une SIMATIC IFP2200 (Industrial Flat Panel):

Figure III.7 L'interface homme-machine IHM SIMATIC IFP2200.[20]

Caractéristiques du SIMATIC IFP2200 :

- Robuste façade en aluminium
- Peut être livré comme moniteur ou pupitre tactile
- Installation déportée possible jusqu'à 5 m de l'IPC
- Interface DVI-D et interface DisplayPortV1.1
- Prise en charge du multi-monitorage
- Rétro éclairage réglable par logiciel
- Alimentation 24 V CC
- Jusqu'à 16 millions de couleurs.

III.6 TIA Portal (Totally Integrated Automation):

Totally Integrated Automation apporte une réponse optimale à toutes les exigences et offre un concept ouvert vis à vis des normes internationales et de systèmes tiers. Avec ses principaux caractéristiques et robustesse, Le TIA Portal accompagne l'ensemble du cycle de vie d'une machine ou d'une installation. L'architecture système complète offre des solutions complètes pour chaque segment d'automatisation sur la base d'une gamme de produits complète. [6]

III.6.1 Description du logiciel TIA Portal :

La plateforme « Totally Intergrated Automation Portal » est le nouvel environnement de travail Siemens qui permet de mettre en œuvre des solutions d'automatisation avec un système d'ingénierie intègre comprenant les logiciels SIMATIC Step7 et SIMATIC WinCC.

III.6.2 Les avantage du logiciel TIA portal [6] :

- Programmation intuitive et rapide : avec des éditeurs de programmation nouvellement développés SCL, CONT, LOG, LIST et GRAPH.

- Efficacité accrue grâce aux innovations linguistiques de STEP7 : programmation symbolique uniforme, Calculate Box, ajout de blocs durant le fonctionnement, et bien plus encore.
- Performance augmentée grâce à des fonctions intégrées : simulation avec PLCSIM, télémaintenance avec Télé Service et diagnostic système cohérent.
- Technologie flexible : Fonctionnalité motion control évolutive et efficace pour les automates S7-1500 et S7-1200.
- Sécurité accrue avec Security Integrated : Protection du savoir-faire, protection contre la copie, protection d'accès et protection contre la falsification.
- Environnement de configuration commun avec pupitres IHM et entraînements dans l'environnement d'ingénierie TIA Portal

III.6.3 Vue du portail et vue du projet :

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

- Vue du portail : elle est axée sur les tâches à exécuter et sa prise en main est très rapide.

- 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. [23]

- **Vue du portail :** Chaque portail permet de traiter une catégorie de tâche (action), la fenêtre affiche la liste des actions peuvent être réalisées pour la tache sélectionnée. La vue de portail TIA PORTAL est représentée par la figue ci-dessous.

Figure III.8 Vue du portail

- **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 figue suivante montre la vue de projet TIA PORTAL

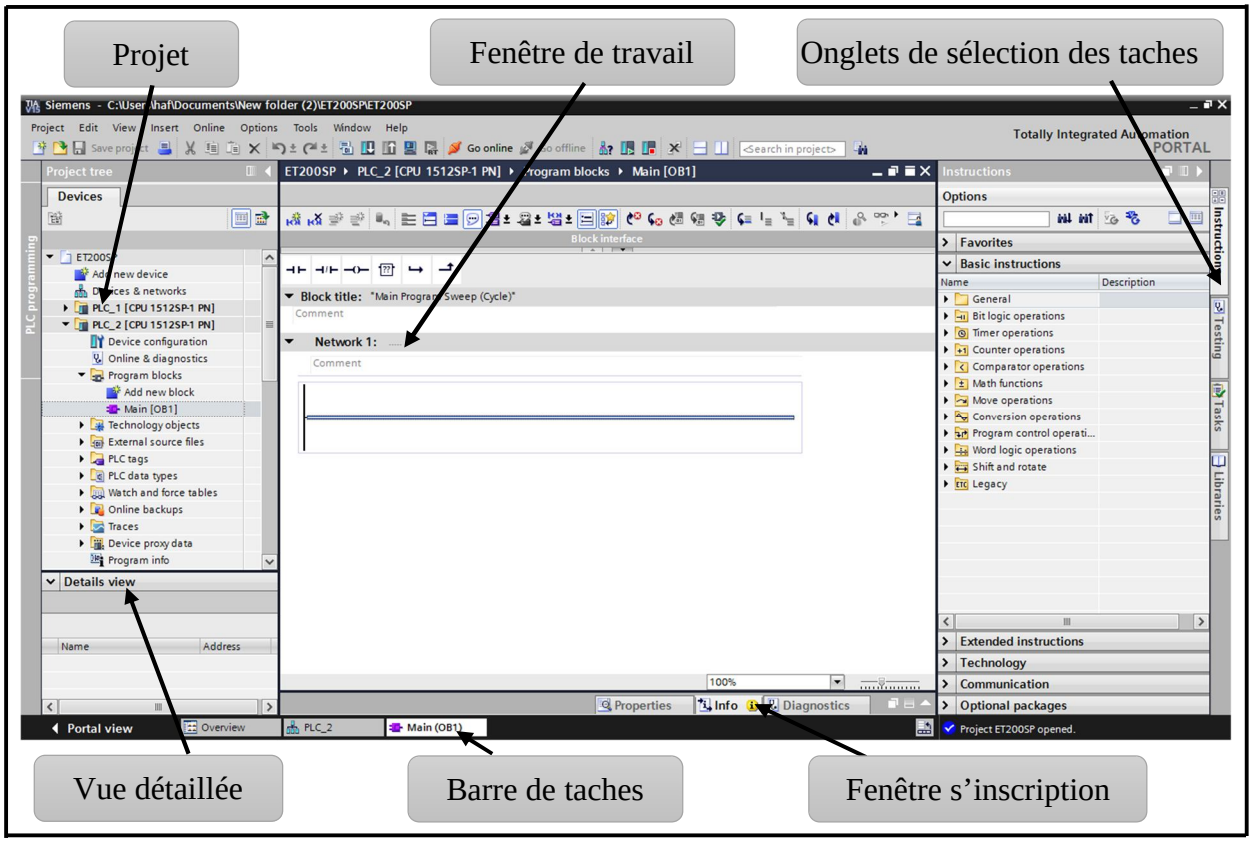

Figure III.9 Vue du projet

- 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 hommes machines (IHM)
- La fenêtre d'inspection permet de visualiser des informations complémentaires sur objet sélectionné où 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 et de redimensionner, réorganiser, désancrer les différentes fenêtres.[23]

III.6.4 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 indiquée comment créer un projet.

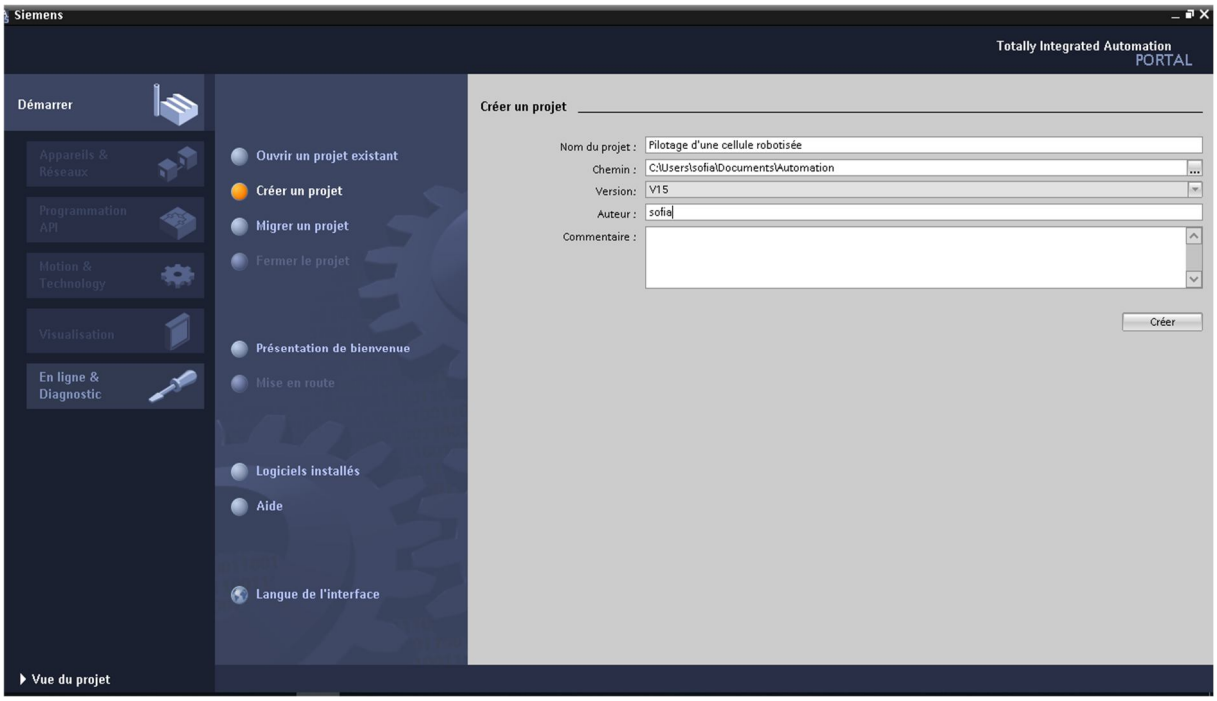

Figure III.10 Création d'un projet

III.6.5 Configuration matériel (hardware) [23] :

Une configuration matérielle est nécessaire pour :

- Les paramètres ou les adresses préréglés d'un module.
- Configurer les liaisons de communication.

Une fois un projet est 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 qu'on peut les 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 première méthode de la configuration et paramétrage du matériel est représenté par la figure suivante.

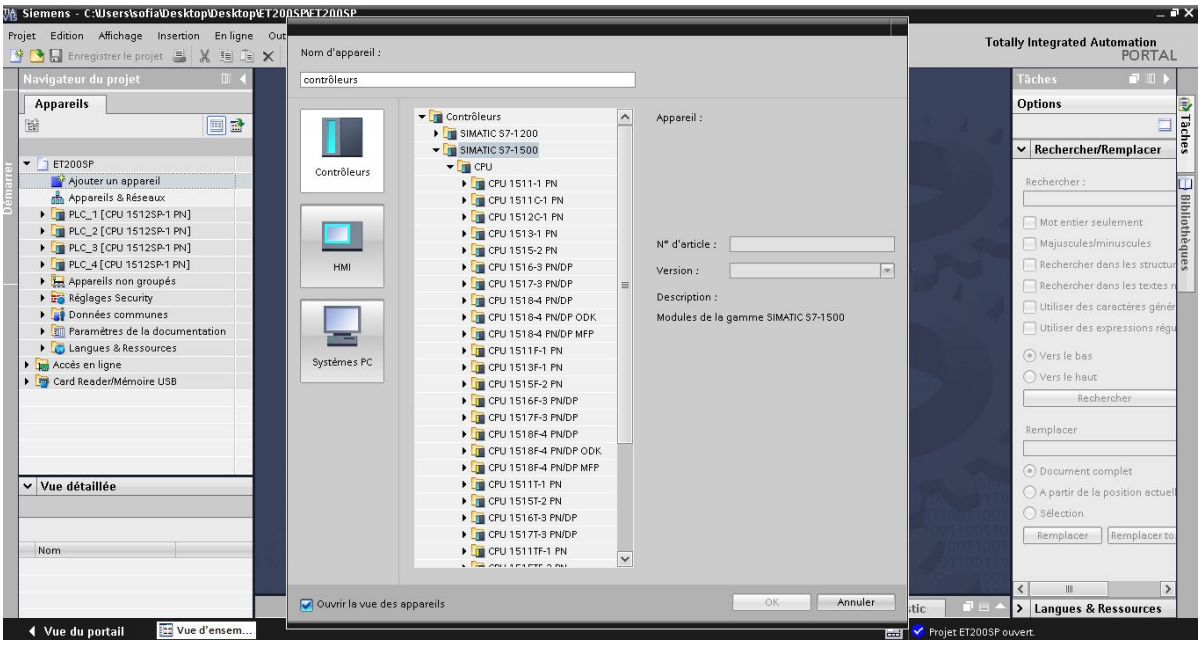

Figure III.11 Première methode de configuration et paramétrage du matériel

Les modules complémentaires de l'API peuvent être ajoutés en utilisant le catalogue. Si on veut ajouter un écran où 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 représente la deuxième méthode de la configuration et du paramétrage du matériel.

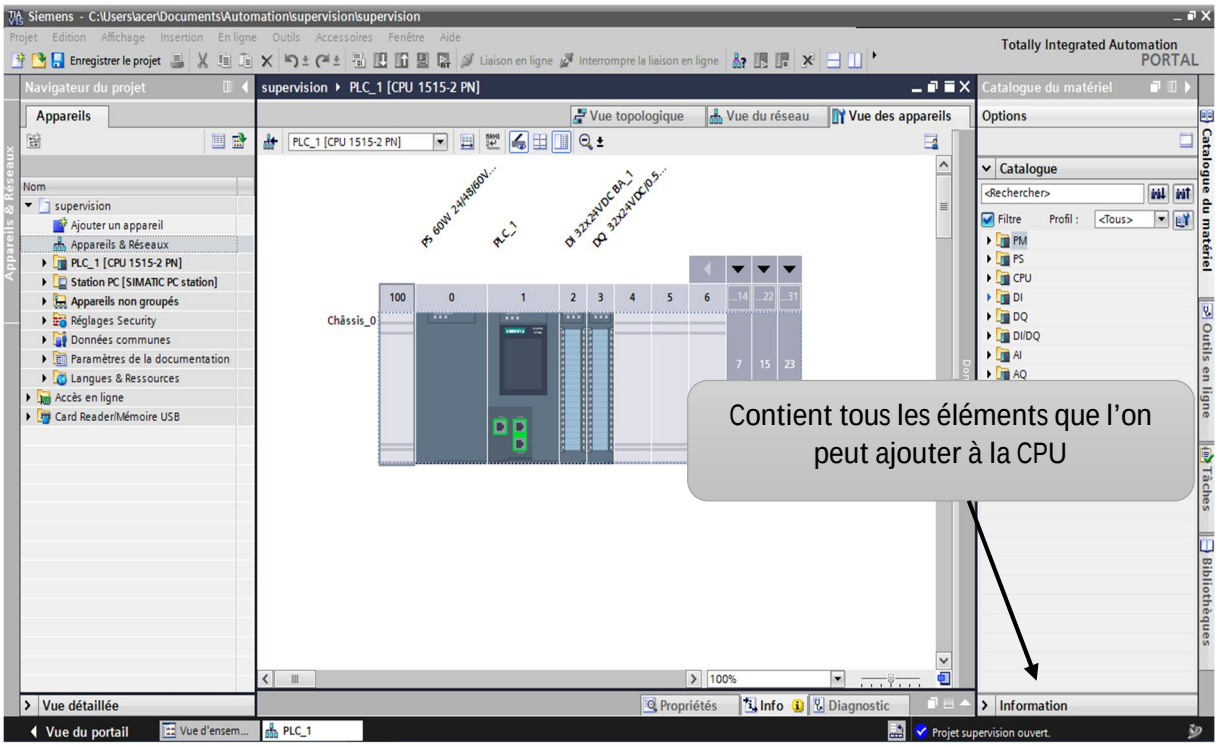

Figure III.12 Deuxième méthode de configuration et paramétrage du matériel

III.6.6 Adresse Ethernet de la CPU :

Toujours dans les propriétés de la CPU, il est possible de définir son adresse Ethernet, un double clic sur le connecteur Ethernet de la station fait apparaitre la fenêtre d'inspection permettant de définir ses propriétés.

Pour établir une liaison entre la CPU et la console de programmation, il faut affecter aux deux appareils des adresses appartenant au même réseau. On utilisera comme adresse pour l'automate 192.168.0. N° de l'automate. L'adresse Ethernet de la CPU est représentée dans la figure ci-dessous.

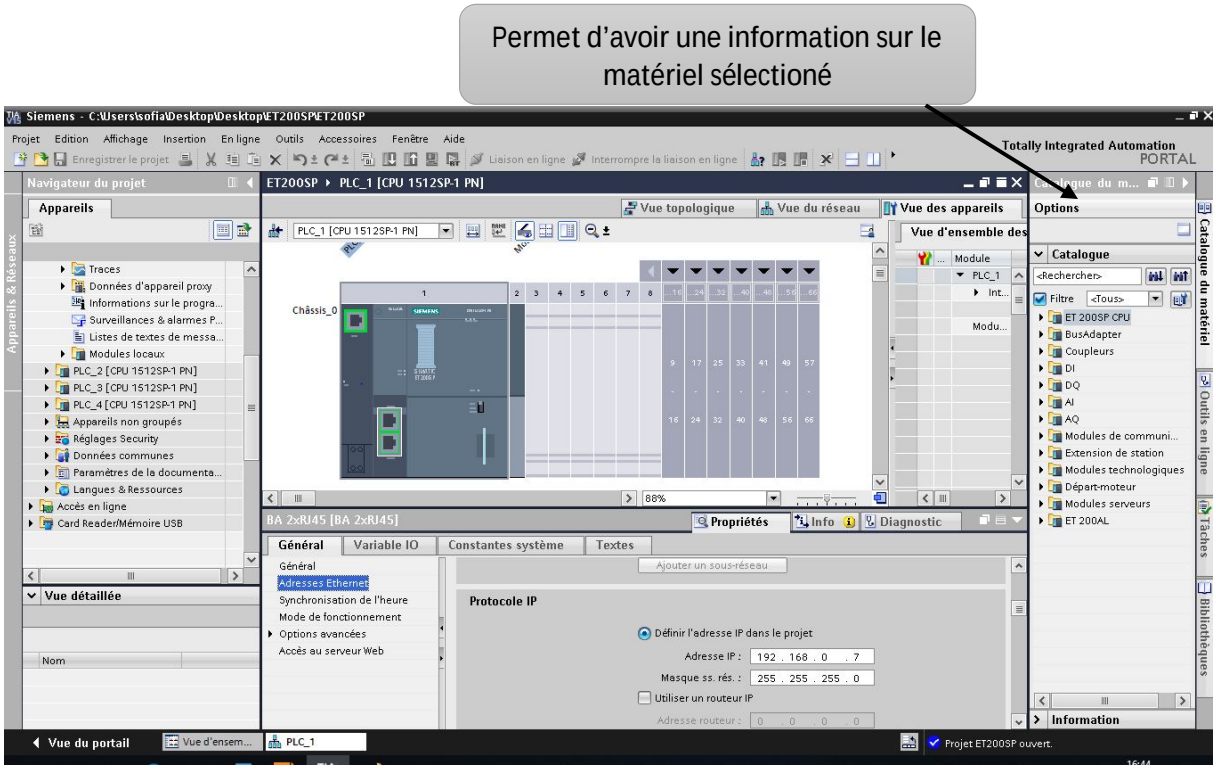

Figure III.13 Adresse Ethernet de la CPU

III.6.7 Compilation et chargement de la configuration matérielle :

Une fois la configuration matérielle réalisée, il faut la compiler et la charger dans l'automate, la compilation se fait à l'aide de l'icône **« compiler »** de la barre de tâche. On sélectionne l'API dans le projet puis cliquer sur l'icône **« compiler ».** En utilisant cette manière, on effectue une compilation matérielle et logicielle. Une autre solution pour compiler est de faire un clic droit sur l'API dans la fenêtre du projet et de choisir l'option Compiler « Configuration matérielle et logicielle », La figure ci-dessous montre l'étape de compilation et chargement de la configuration matérielle.

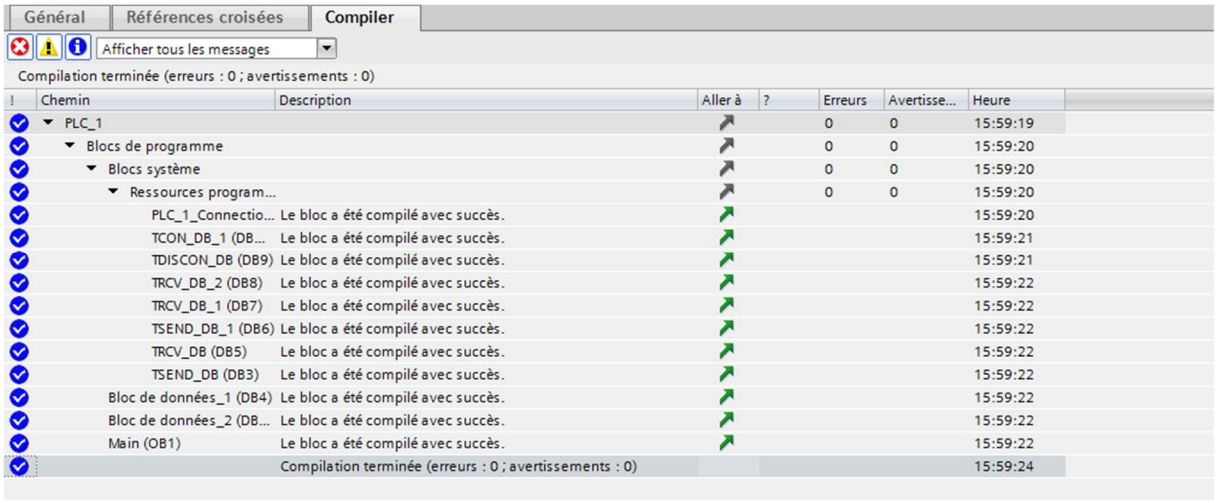

Figure III.14 Compilation de la configuration matérielle

Pour charger la configuration dans l'automate, on effectue un clic sur l'icône **« charger dans l'appareil »**. La fenêtre ci-dessous s'ouvre et vous devez faire le choix du mode de connexion (PN/IE, Profibus, MPI). Si vous choisissez le mode PN/IE, l'API doit posséder une adresse IP, la figure suivante indique la compilation et le chargement en mode de connexion PN/IE.

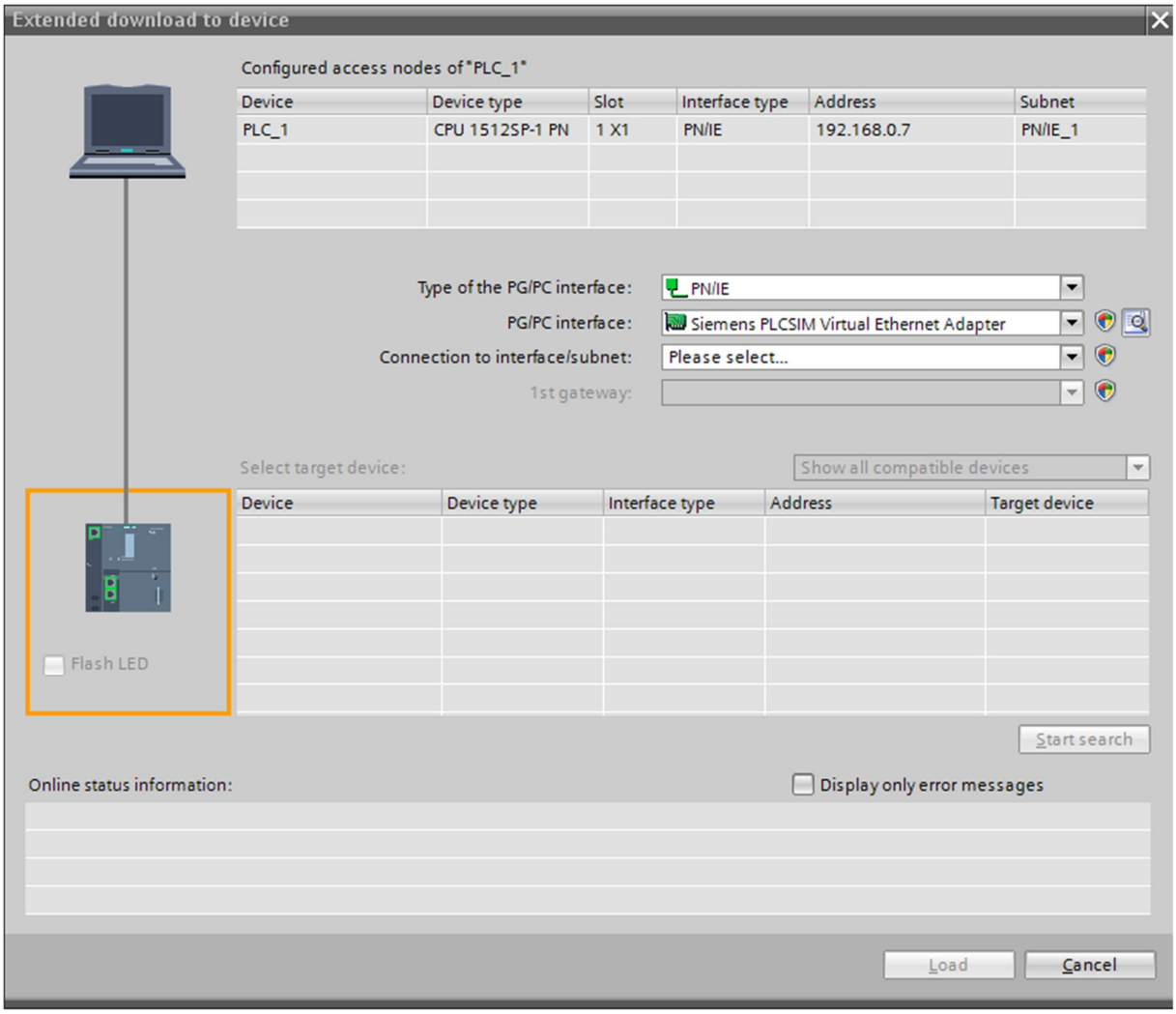

Figure III.15 Compilation et chargement en mode de connexion PN/IE.

Pour une première connexion ou pour charger l'adresse IP désirée dans la CPU, il est plus facile de choisir le mode de connexion MPI et de relier le PC à la CPU via le **« PC Adapter ».** Si le programme trouve un appareil, ce dernier figurera dans la liste en bas de la fenêtre. La touche **« Clign. DEL »** permet de faire clignoter une LED sur la face avant de l'appareil afin de s'assurer que l'on est connecté à l'appareil désiré. La compilation et le chargement en mode de connexion MPI est indiquée dans la figure ci-dessous.

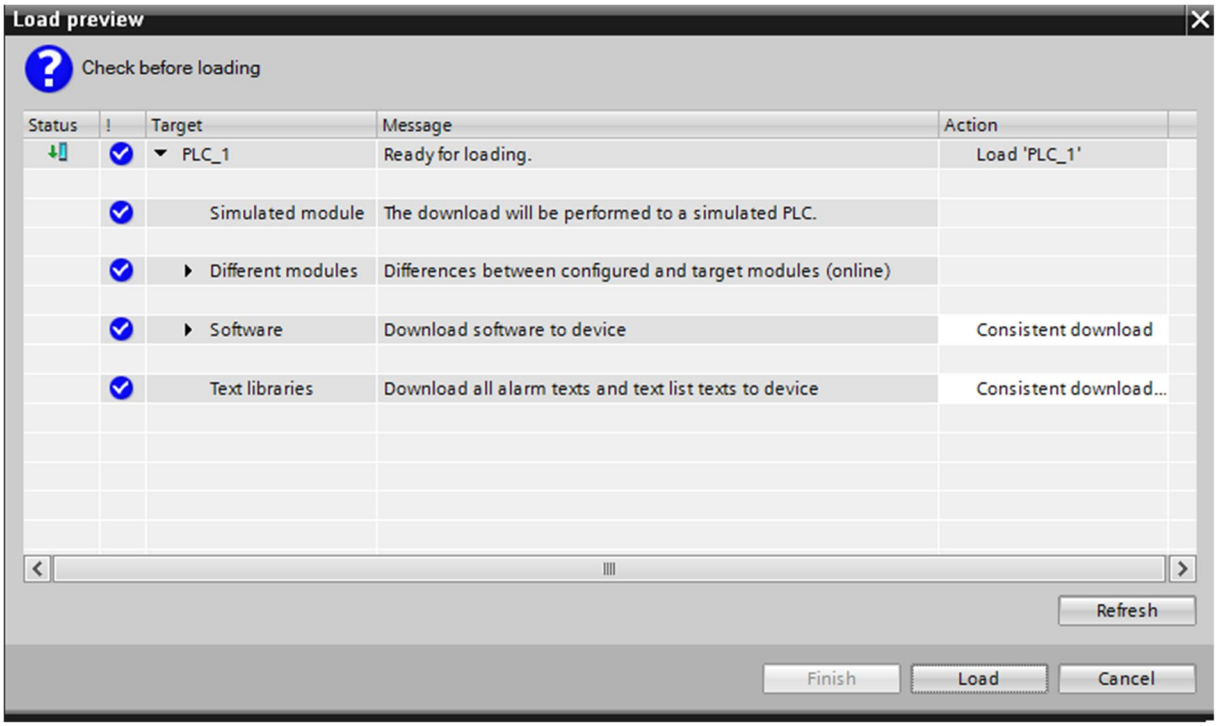

Figure III.16 Compilation et chargement en mode de connexion MPI.

Une fois la configuration terminée, on peut charger le tout dans l'appareil, des avertissements/confirmations peuvent être demandés lors de cette opération. Si des erreurs sont détectées, elles seront visibles via cette fenêtre. Le programme ne pourra pas être chargé tant que les erreurs persistent. L'automaticien se doit de les corriger en modifiant le programme où la configuration matérielle,

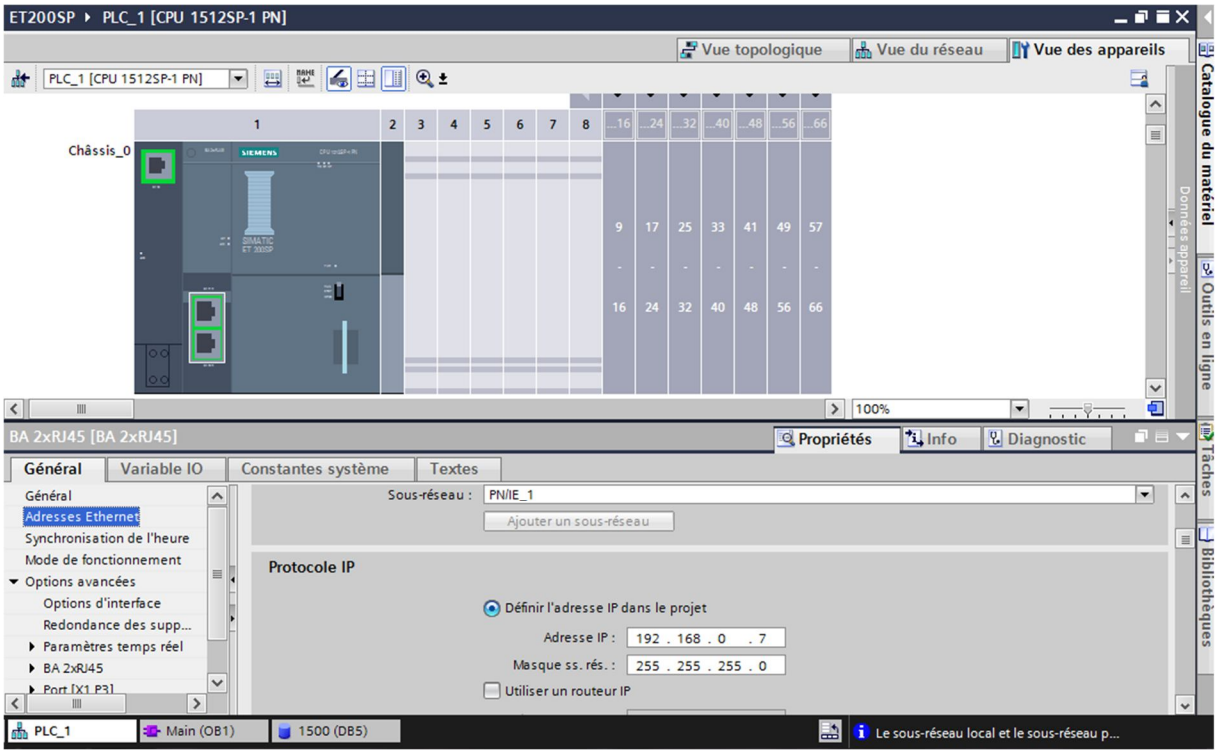

Figure III.17 Confirmation de chargement de la configuration matérielle.

III.6.8 PLCSIM Advanced V2.0 :

L'automatisation peut commencer à être testée pendant la phase de conception. L'interaction entre conception et automatisation permet des tests initiaux et une mise en service virtuelle sans recourir à une machine réelle.

Ce programme peut simuler un véritable dispositif de la série PLC S7-1200, S7-1500, etc. Cependant, les caractéristiques avancées du S7-1200, S7-1500 ne sont pas encore simulées. Par conséquent, le SIMATIC S7-PLCSIM Advanced est né et nous permet de créer des contrôleurs virtuels pour simuler les contrôleurs S7-1500 et ET-200SP et fournir une simulation complète des fonctions.[24]

III.7 Compatibilité avec TIA Portal et les versions du microprogramme de la CPU :

Le microprogramme utilisé dans PLCSIM Advanced V2.0 correspond à celui d'une CPU S7-1500 V2.5. Les versions suivantes sont compatibles [25] :

PLCSIM Advanced	TIA Portal	Supported CPU firmware version
V1.0 SP1	V14 to V15.1	V1.8 to V2.0
V2.0	V14 to V15.1	V1.8 to V2.5
2.0 SP1	V14 to V15.1	V1.8 to V2.6

Tableau III.1 Compatibilité avec TIA PORTAL et les versions du microprogramme de la CPU

III.8 Réseau PROFINET [26] :

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. Nous donnons ci-après la définition de la technologie Ethernet et le sigle TCP/IP :

- **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.
- **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.

III.8.1 Spécification sur le PROFINET IO [26] :

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 S7-400on 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 : qui est utilisé pour le paramétrage, la configuration et les opérations de lecture / écriture acycliques.
- Le canal RT ou Real Time : qui est utilisé pour le transfert de données cyclique standard et les alarmes. Cette communication contourne l'interface TCP/IP standard pour accélérer l'échange des données avec les automates programmables.
- Le canal, Isochro Real Time (IRT) ou temps réel isochrone : c'est un canal à très haut débit utilisé pour les applications Motion Control.

III.9 Conclusion :

L'automate programmable (S7-1500 et ET-200SP) adoptée pour le pilotage de notre cellule et le logiciel de programmation utilisé pour la commande (TIA Portal V15) sont présentés dans ce chapitre.

IV.1 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.

Afin de piloter notre cellule robotisée, nous allons réaliser un programme que nous allons implanter dans l'automate centrale S7-1500 et les quatre ET200SP décentralisés grâce au logiciel de conception et d'automatisation TIA PORTAL V15 de SIEMENS.

Dans ce chapitre nous allons définir les types de communication exploités, ensuite décrire l'implantation du programme d'automatisation élaboré à partir de l'analyse fonctionnelle et on terminera par la présentation du système superviseur de la plateforme.

IV.2 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 ET200SP. On va adopter une architecture de pilotage centralisée, tel que l'automate S7-1500 qui sera l'automate central sur lequel on établira le programme complet qui commandera la cellule, mais on va utiliser les systèmes d'E/S (ET200SP) décentralisées avec leurs fonctionnalités de CPU pour des raisons qu'on expliquera par la suite.

Pour connecter les quatre ET 200 SP à l'automate central, on va les mettre sous un réseau PROFINET. La figure suivante représente la connexion des différents éléments de la partie commande sous un réseau PROFINET.

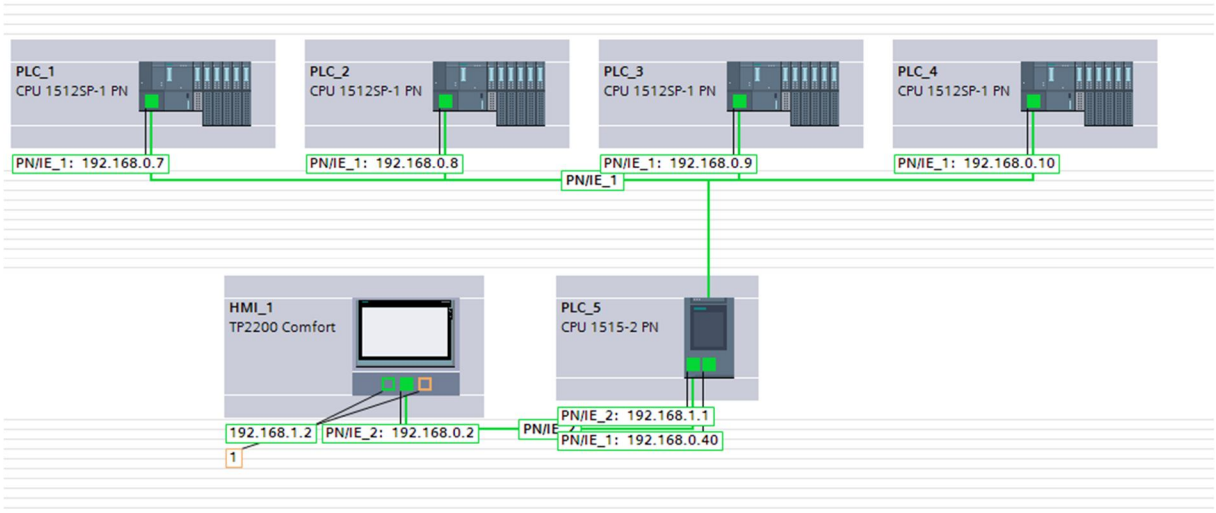

Figure IV.1 Schéma représentant la connexion des différents éléments de la partie commande sous un réseau PROFINET

Remarque : Pour connecter ces appareils sous le même réseau, il suffit de le configurer 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 II. 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.3 Communication SOCKET :

Notre but c'est de faire communiquer la partie commande avec la plateforme virtuelle conçue sur V-REP via une communication socket du type client/serveur. Donc deux programmes sont nécessaires, un sur V-REP l'autre sur le PLC (ET 200SP), de telle sorte qu'un d'eux sera le serveur et l'autre sera le client, c'est le client qui demandera la connexion en premier, et puis par la suite l'échange de donnée 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 V-REP, 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 200SP pour créer dessus des interfaces socket, chacun des ET 200 SP est spécifié pour un seul poste.

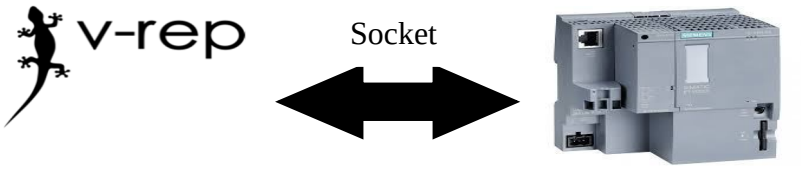

Figure IV.3 Communication socket direct entre VREP et API

IV.4 La création d'une communication socket sur V-REP :

La bibliothèque de commande de V-REP possède les instructions nécessaires 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 :

```
        -- socket--
local socket=require("socket")
local client=socket.tcp()
sim.setThreadIsFree(true)
local result=client:connect('192.168.0.7',2232)
sim.setThreadIsFree(false)
        --fin socket --
```

Figure IV.4 Communication socket sur V-REP (connexion avec l'automate ET 200SP)

```
local returnData=client:receive(1)
 if (returnData==nil) then
      break -- Read error
 end
 if (returnData~=nil) then
 print("Poste 1 Receive",returnData)
 var=tonumber(returnData)
 print("socket",returnData)
 end
 sim.setThreadIsFree(false)
```

Figure IV.5 Communication socket sur V-REP (réception des données)

IV.5 Création du tableau des variables API :

Le tableau des variables API permet de définir la liste des variables qui seront utilisées lors de la programmation comme la désignation de l'ensemble des capteurs photoélectriques, les différents transmetteurs et récepteurs des signaux, les retours de marche et d'arrêt des convoyeurs et du robot, la fin de cycle et les fins des tâches.

Table de variables standard

	Nom	Type de données	Adresse	Réma...	Acces...	Ecritu...	Visibl...	Surveill...	Commentaire
1	conecte	Bool	%M0.0	☐	☑	☑	☑		
2	Tag_1	Bool	%M0.1	☐	☑	☑	☑		
3	rcv	Bool	%M0.2	☐	☑	☑	☑		
4	Tag_2	Bool	%M0.3	☐	☑	☑	☑		
5	send	Bool	%M0.4	☐	☑	☑	☑		
6	Tag_3	Bool	%M0.5	☐	☑	☑	☑		
7	Tag_4	Bool	%M0.6	☐	☑	☑	☑		
8	Tag_5	Bool	%M0.7	☐	☑	☑	☑		
9	capteur3	Bool	%M1.0	☐	☑	☑	☑		
10	départ	Bool	%M1.1	☐	☑	☑	☑		
11	dd	Bool	%M1.2	☐	☑	☑	☑		
12	capteur	Bool	%M1.3	☐	☑	☑	☑		
13	depart	Bool	%M1.4	☐	☑	☑	☑		
14	<Ajouter>			☐	☑	☑	☑		

Figure IV.6 Tableau de variable

IV.6 Ajout des blocs d'organisation (OB) :

Ils commandent le traitement du programme, il est possible par l'intermédiaire des OB de réagir aux événements cycliques, la figure IV.7représente l'ensemble des réseaux de nos blocs OB.

Figure IV.7 Réseaux de blocs OB réalisé.

IV.7 La création d'une interface socket sur l'ET 200 SP :

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

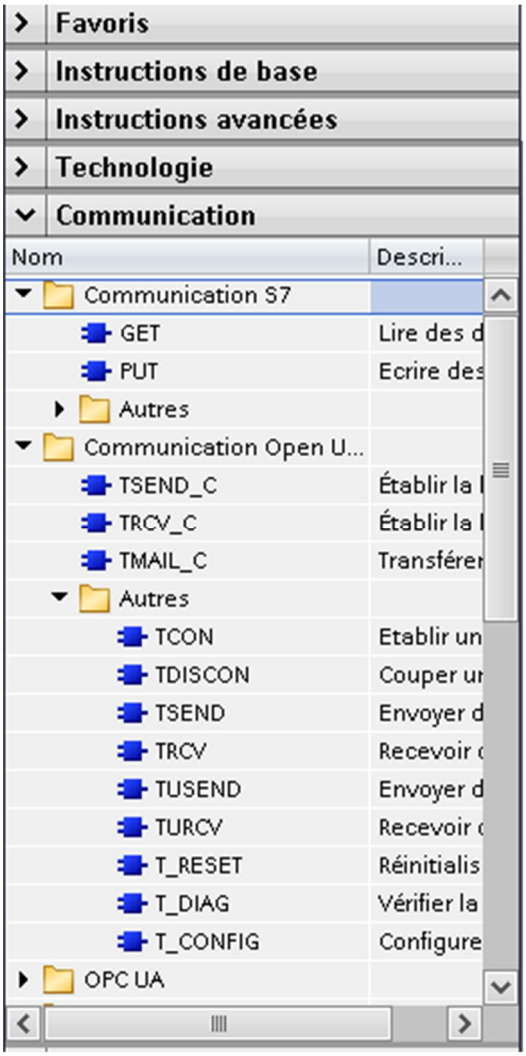

Figure IV.8 La bibliothèque communication.

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

- **TCON :** pour l'établissement de la liaison.[27]

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. La figure ci-dessous représente le bloc d'instance de l'instruction TCON.

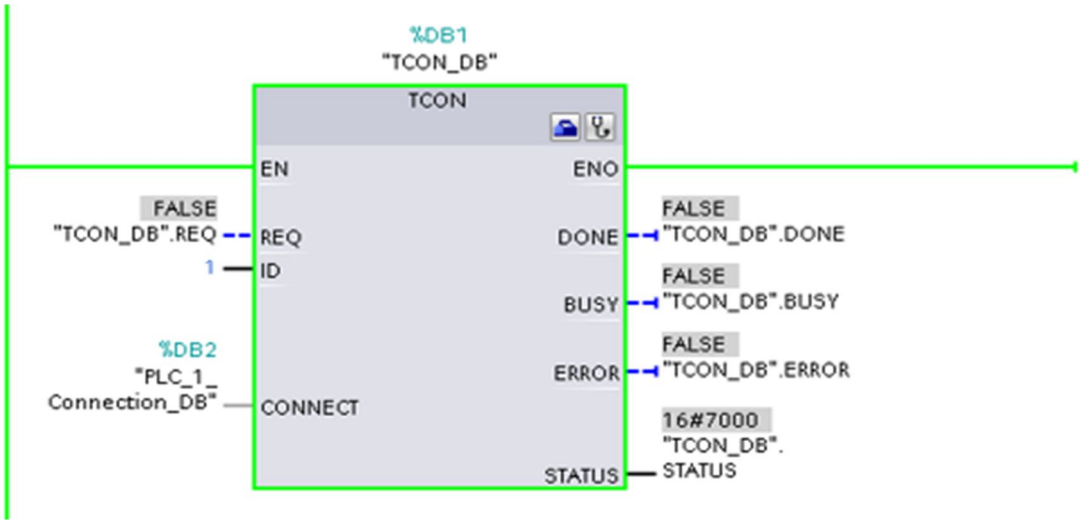

Figure IV.9 Bloc d'instance de l'instruction TCON.

La figure suivante représente le paramètre de liaison de TCON, là où on précise le client, le serveur, l'adresse IP et le port de l'appareil partenaire :

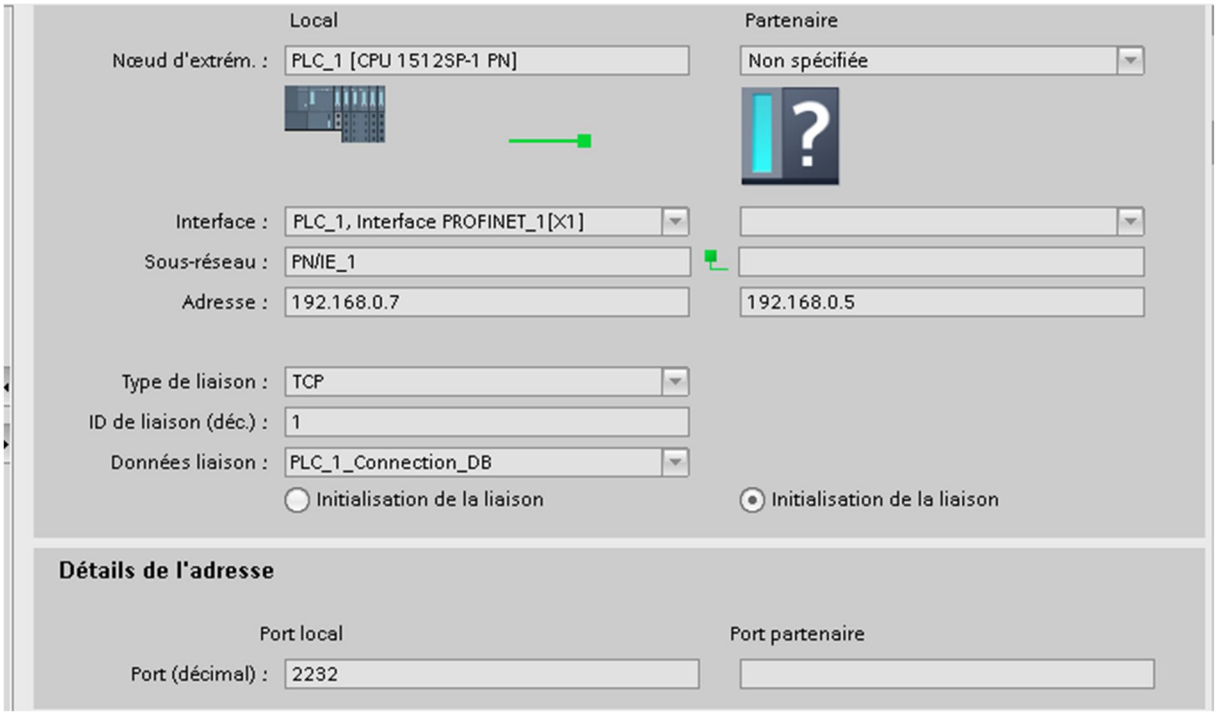

Figure IV.10 Paramétrage de l'instruction TCON.

Dans notre cas, l'appareil partenaire est le PC qui contient le programme V-REP. Donc, on lui donne son adresse IP et le port du client avec qui il est sensé se connecter.

- **TSEND :** pour l'émission des données.[27]

L'instruction "TSEND" permet l'envoi de données via une liaison de communication existante. Et elle s'exécutée de manière asynchrone. La figure ci-dessous représente le bloc d'instance de l'instruction TSEND.

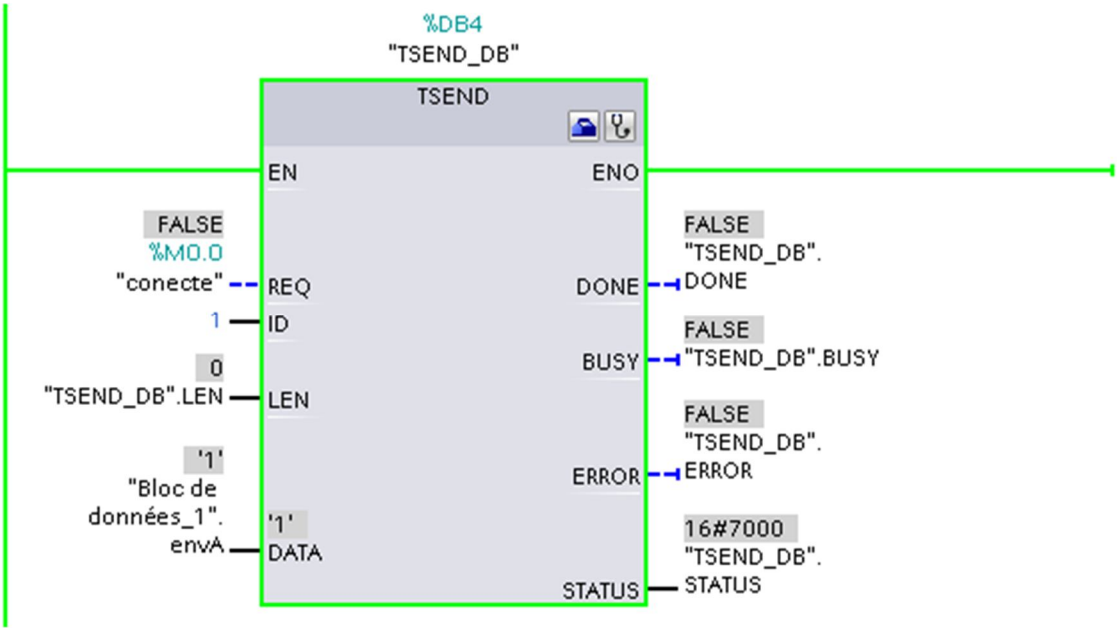

Figure IV.11 Bloc d'instance de l'instruction TSEND.

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.

- **TRCV :** Pour la réception des données.[27]

L'instruction "TRCV" permet la réception de données via une liaison de communication existante. Et elle s'exécutée de manière asynchrone. La figure suivante représente le bloc d'instance de l'instruction TRCV.

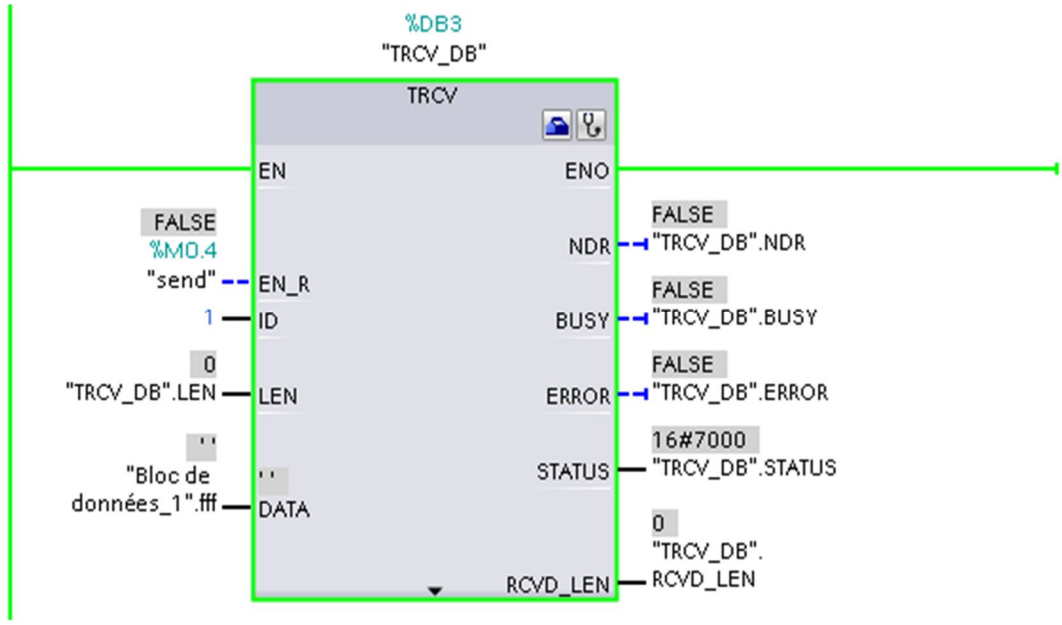

Figure IV.12 Bloc d'instance de l'instruction TRCV.

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 coté émission et du coté réception. Dans notre cas on va envoyer et recevoir des données de type char.

- **TDISCON :** pour la fermeture de liaison.[27]

L'instruction "TDSCON" interrompt une liaison de communication de la CPU a un partenaire de liaison. La figure ci-dessous représente le bloc de l'instruction TDISCON.

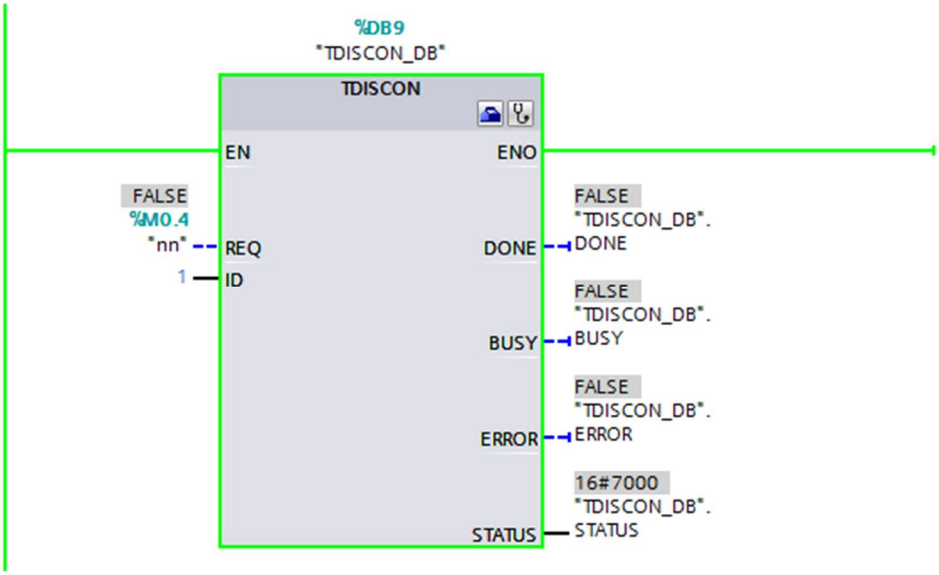

Figure IV.13 Bloc d'instance de l'instruction de déconnexion.

IV.8 La Création d'une communication socket entre Les quatre ET200 SP :

La même chose que la communication socket entre V-REP et ET200 SP, on établit les liaisons par les instructions TCON, TSEND, TRCV, TDISCON. La figure ci-dessous représente les blocs d'instance de TCON des quatre ET 200 SP.

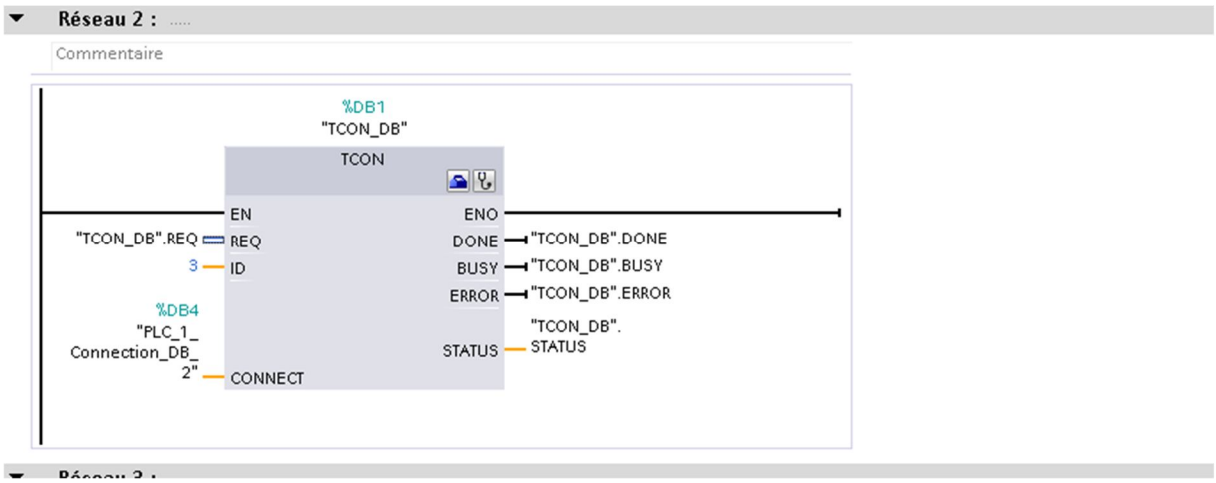

Figure IV.14 Les blocs d'instance de TCON de quatre ET 200 SP.

Dans les paramètres de liaison de TCON, on précise l'adresse IP et le port de serveur d'ET 200 SP partenaire. La figure ci-dessous représente la configuration de l'instruction TCON.

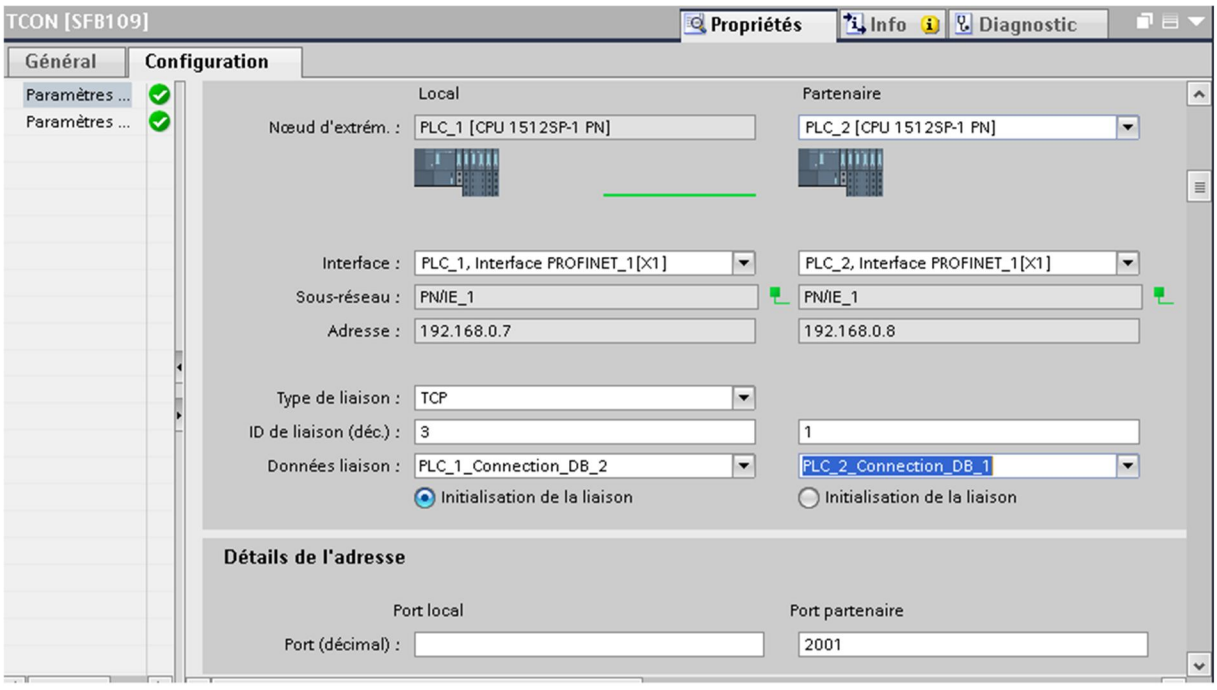

Figure IV.15 Paramétrage de TCON.

IV.9 Communication S7 [28] :

Le S7Comm (S7 Communication) est un mode de communication propre à Siemens. Il permet de faire communiquer des automates Siemens de la famille S7-300 /400/1200/1500, et les ET200 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 distante. 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.

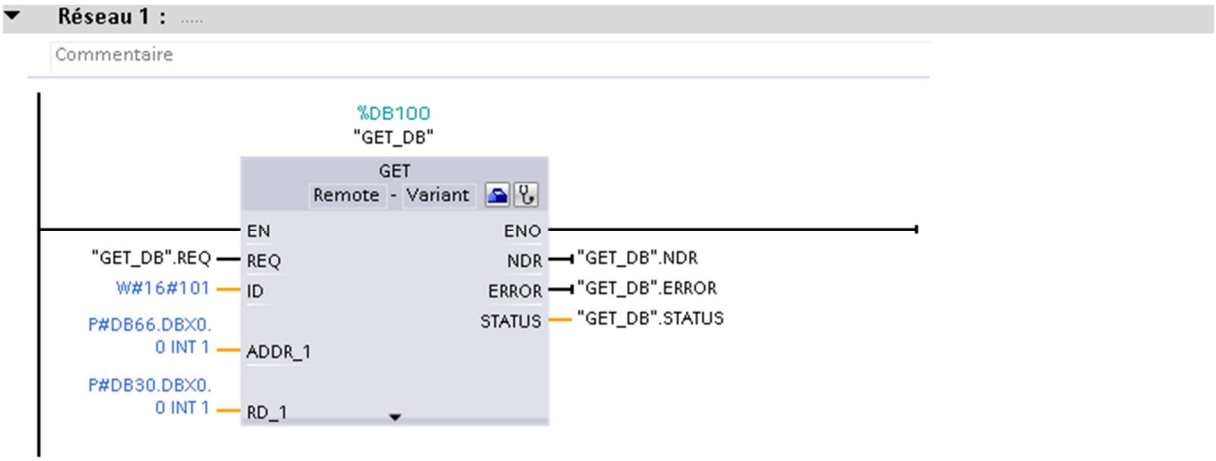

Figure IV.16 L'instruction GET

- 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.

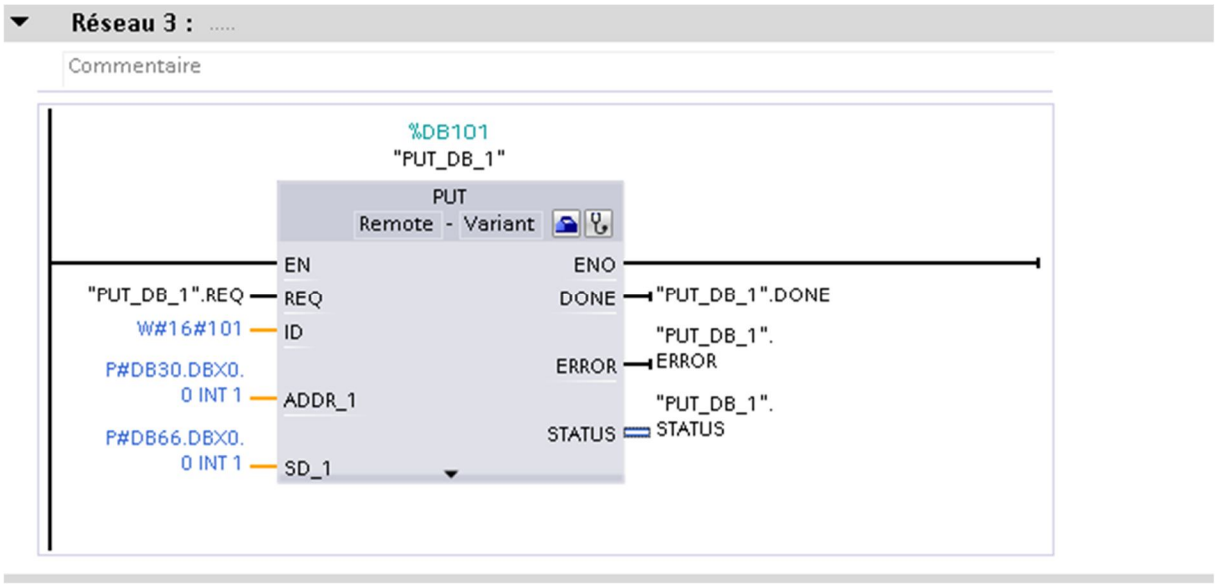

Figure IV.17 L'instruction PUT

La figure suivante représente un schéma qui résume les différents types des communications.

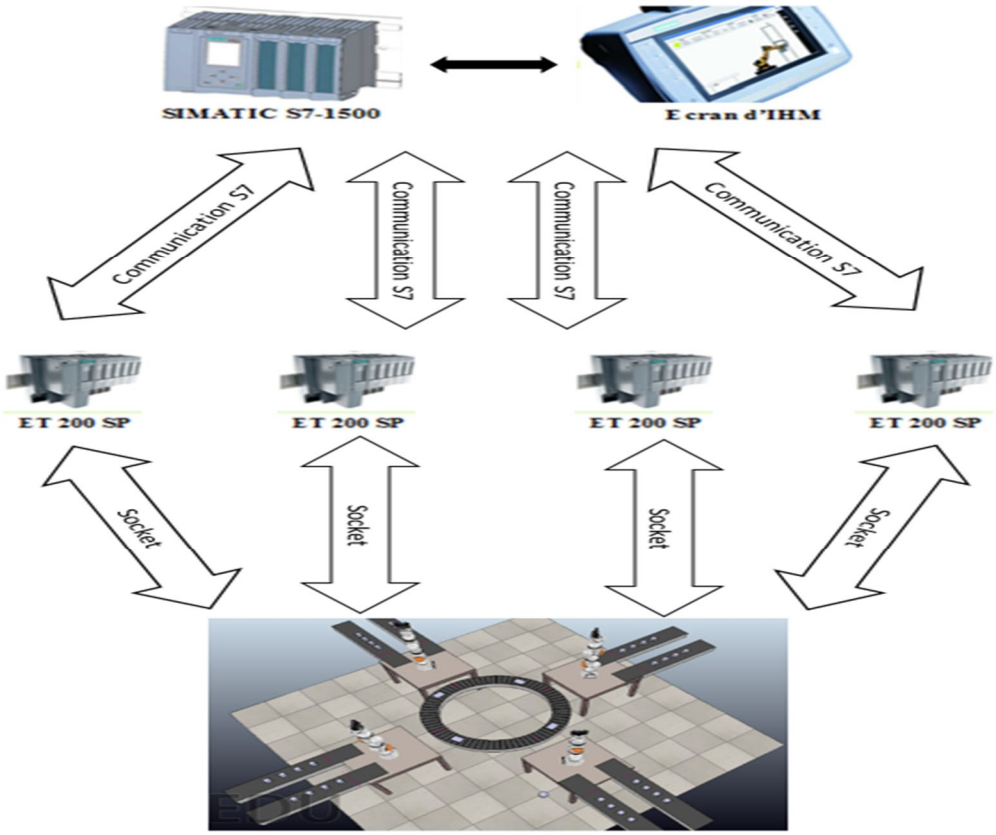

Figure IV.18 Schéma résumant les différents types de communication utilisés.

IV.10 Pilotage de la cellule :

IV.10.1 GRAFCET du système complet :

Ce GRAFCET permet la clarification et la compréhension de différents comportements du système réalisé suivant son cahier de charge.

Entrées	Sorties
Dcy : départ cycle. C1 : capteur 1 de palettes sur le convoyeur circulaire. CP1 : capteur 1 de pièces sur le convoyeur simple. CP2 : capteur 2 de pièces sur le convoyeur simple. Robot-M1 : robot à l'initiale prise. Robot-M2 : robot à l'initial dépôt. Robot-D : robot à la position droite. Robot-G : robot à la position gauche. Pince-F : pince fermée. Pince-O : pince ouverte. PPCC : pièce sur la palette (convoyeur circulaire)	Conv-C : convoyeur circulaire marche. Conv-S : convoyeur simple marche. Robot-DC : robot descendre. Robot-TD : robot tourne droite Robot-TG : robot tourne gauche. F-Pince : fermeture de pince. O-Pince : ouverture de pince. Robot-Re : robot recule.

Tableau IV.1 Les entrées sorties du GRAFCET.

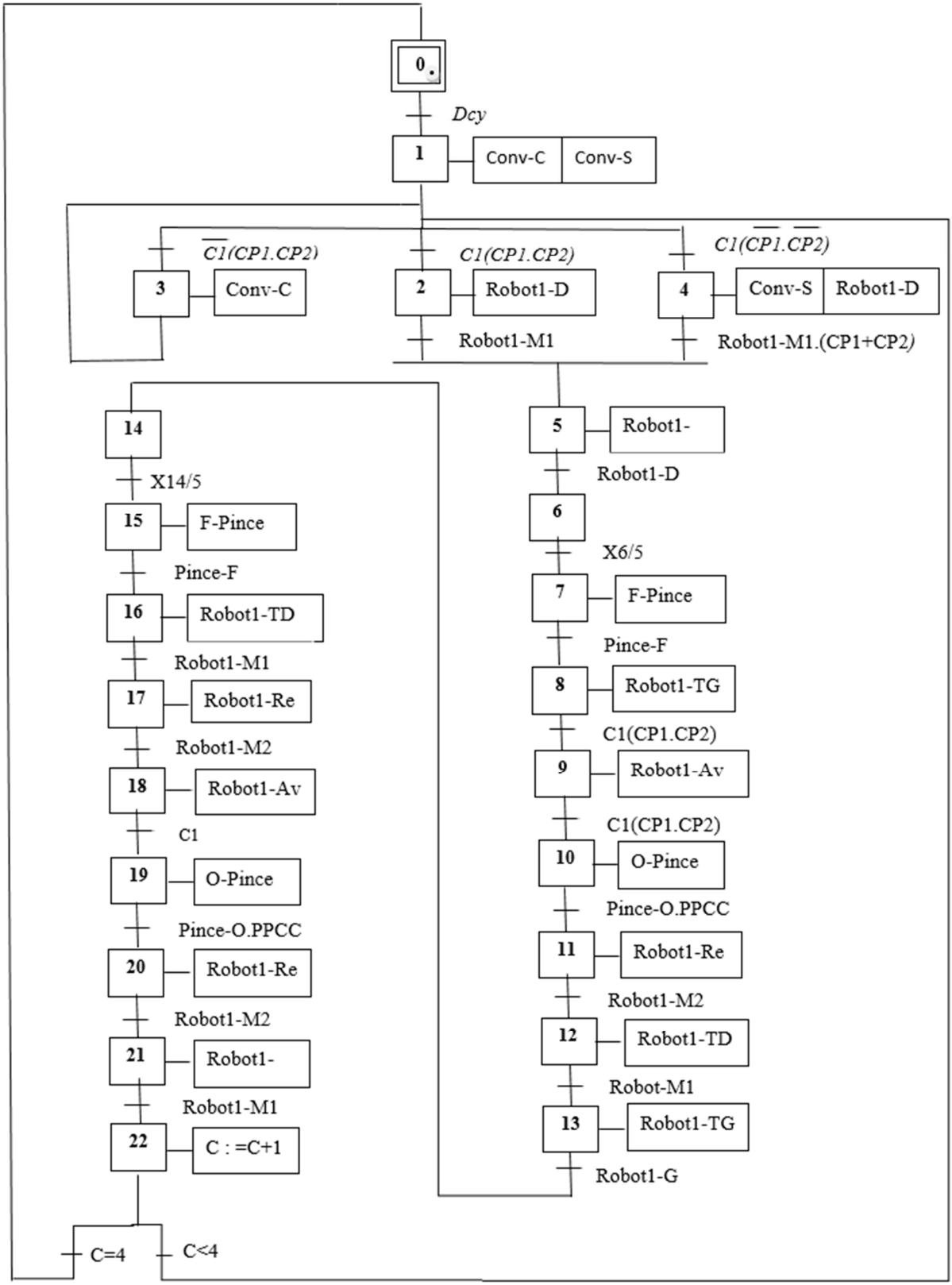

Figure IV.19 GRAFCET du système.

Ce GRAFCET représente le fonctionnement du premier poste. Les autres postes seront identiques à celui-là en changeant les capteurs C1, CP1, CP2 et le Robot1 par le capteur et le robot correspondants au poste.

IV.11 Conclusion :

Dans ce chapitre on a présenté le pilotage de la cellule réalisée 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 on a résumé le déroulement du système par un GRAFCET.

Conclusion générale

L'objectif de ce travail consiste à connaitre de près la démarche de résolution des problèmes et la mise en œuvre de la Quatrième Révolution Industrielles d'une unité de production.

Le pilotage d'une cellule robotisée virtuelle conçue sur le logiciel simulateur V-REP commandé par un automate programmable industriel PLC (S7-1500) en se basant sur les techniques Hardware In The Loop (HIL) 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 tâches de l'automate central.

La prise de connaissance du SIMATIC STEP 7 intégré à TIA Portal V15, nous a permet de programmer le fonctionnement de la plateforme d'assemblage robotisée et de récupérer les états des variables intéressent pour le système.

En effet, tout au long de cette période, nous avons fait face à de nombreux problèmes, et des difficultés majeures autours de la compréhension du système et l'établissement des séquences de son fonctionnement.

Pour atteindre l'objectif de ce projet, nous avons commencé par prendre connaissance de l'installation puis identifier les éléments qui la constituant. Le déplacement sur site nous a nettement aidé à mieux assimiler l'envergure du projet et nous a permis d'avoir un avant-gout des responsabilités qui incombent aux ingénieurs.

Enfin, nous espérons pour les prochaines promotions d'améliorer le travail réalisé sur le pilotage des systèmes robotisés et leur commande par APIs en le rendant rapide et plus précis pour le développement industriel.

[1] S. Deveux, M. Rachline, Introduction à l'automatisme : Schneider Electric, Encyclopédie économie 3000, série haute technologie, ISBN 2-7191-0551-1,2000.

[2] https://www.visiativ-industry.fr/industrie-4-0/

[3] https://connect.factorysystemes.fr/blog/2014/10/09/les-avantages-de-lindustrie-4-0/

[4] C. Vrignon, M. Thenaisie, l'automatisation », ISTIA, 17 octobre 2005.

[5] H. Belkacem, A. Bias, Système de Contrôle Distribué (DCS) avec l'exploitation de l'automate programmable AC800F(ABB), mémoire de Master, université de Mohamed Khider, Biskra, 2012.

[6] S. Mellali, L. Yousfi, Etude de l'automatisation et de la supervision d'un procédé de lavage de filtres Niagara à CEVITAL, mémoire de Master, université Abderrahmane Mira, Bejaia, 2017.

[7] L. Bergougnoux, Automates programmables industriels API, revue technique, université de polythech SIIC, Marseille, 2005.

[8] A. Daghmous, H.Loucif, Commande et Supervision d'un système industriel par Automate programmable, mémoire de Master, université de Larbi Tebessi, Tebessa, 2017 / 2018.

[9] I. Rahmani, D. Ferroug, Etude et automatisation de la banderoleuse au sien de l'unité de conditionnement de lait Candia, mémoire de master, université de Abderrahmane Mira, Bejaia, 2017.

[10] https://www.technologuepro.com/cours-automate-programmable-industriel/Les-automates-programmables-industriels-API.html.

[11] B. Imatouken, Y. SAHEB, Etude et automatisation d'une station de pompage d'eau glacée, mémoire de Master, université Abderrahmane Mira de Bejaia, 2017/2018.

[12] F. Tighzer, S. Messaoudi, Conversion et combinaison de deux automates S5-95U et S7-200 vers un automate S7-300, mémoire de Master, université Abderrahmane Mira de Bejaia, 2015/2016.

[13] ELWE, Système automatisées, bus de terrain, API SIEMENS , systèmes didactiques pour l'enseignement et la formation en science et technique Industriel, 2001.

[14] https://www.edition-ellipses.fr

[15] http://www.coppeliarobotics.com/helpFiles/

[16] T. Letrouve, Structuration de la commande de la simulation au prototype d'un véhicule hybride double parallèle au travers de la représentation énergétique macroscopique, thèse de doctorat, université des Sciences et Technologies, Lille, 2013.

[17] G. Fleury, P. Lacomme, A. Tanguy, Simulation à événements discrets, chapitre1, page

pp.66, 2006.

[18] S. Ehsan, B. Hamdaoui, A Survey on energy-efficient routing techniques with QoS assurances for wireless multimedia sensor networks" IEEE Communications Surveys & Tutorials, 14 (2), p. 265 – 278, 2012.

[19] https://www.kuka.com/fr-fr/produits-et-prestations/syst%C3%A8mes-de-robots/robots-industriels/lbr-iiwa .

[20] H. Ourad, N. Smahi, Pilotage d'une Plateforme d'Assemblage Robotisée Apport des techniques Hardware-in-The-Loop, Mémoire de master, université Mouloud Mammeri, Tizi-Ouzou, 2018.

[21] https://www.automation-sense.com/blog/automatisme/automate-siemens-s71500.html.

[22] https://cache.industry.siemens.com/dl/files/013/90157013/att_895986/v1/et200sp_cpu1512sp_1_pn_manual_fr-FR_fr-FR.

[23] K. Benamsili, K. Ghanem, Automatisation et supervision via TIA PORTAL V13 d'une centrale de production d'air comprimé pour le process de CEVITAL, mémoire de Master, université de Abderrahmane Mira, Bejaia, 2015.

[24] https://cache.industry.siemens.com/dl/files/047/109759047/att_962042/v3/109759047_PLCSIMAdv_SimTable_DOC_V10

[25] https://support.industry.siemens.com/tf/ww/en/posts/document-detailing-which-version-of-tia-portal-supports-which-firmware/202653/?page=0&pageSize=10

[26] https://w3.siemens.com/mcms/automation/fr/industrial/communications/profinet/pages/default.aspx.

[27] https://support.industry.siemens.com/cs/document/109486139/how-you-configure-a-connection-in-the-tia-portal-?dti=0&lc=en-WW

[28]https://cache.industry.siemens.com/dl/files/807/58875807/att_110536/v1/58875807_net_tcon_s7-1500_