



Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de génie électrique

MÉMOIRE DE MASTER

Sciences et Technologies
Automatique
Automatique et informatique industriel

Réf. : Entrez la référence du document

Présenté et soutenu par :
Raissi Oussama et Larbi Bounab

Le : dimanche 26 juin 2022

Etude et simulation d'un protocole de communication pour l'automatisation industriel

Jury :

| | | | |
|----------------------|-----|----------------------|-----------|
| Dr. Megherbi Hassina | MCA | Université de Biskra | Président |
| Dr. Abada Khaled | MAA | Université de Biskra | Examineur |
| Dr. Achour SAADOUNE | MAA | Université de Biskra | Encadreur |



Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de génie électrique

MÉMOIRE DE MASTER

Sciences et Technologies
Automatique
Automatique et informatique industriel

Etude et simulation d'un protocole de communication pour l'automatisation industriel

Le : ...26 JUIN 2022....

Présenté par :

Raissi Oussama.
Larbi Bounab.

Avis favorable de l'encadreur :

Achour SAADOUNE

Signature Avis favorable du Président du Jury

Cachet et signature

Résumé :

Ce travail voit et étudier les protocoles de communication généralement et particulièrement le protocole profinet et son application dans le domaine industriel et le concrétiser dans notre projet pour les deux méthodes de communication et simuler en utilisant l'API avec le portail TIA et son simulateur qui y est installé.

ملخص:

يقوم هذا العمل بمشاهدة ودراسة بروتوكولات الاتصالات عامة وخاصة بروتوكول profinet وتطبيقه في مجال الصناعة وإدراكه وتطبيقه في مشروعنا بطريقتين للاتصال، والمحاكاة باستخدام API مع بوابة TIA ومحاكيها المثبت PLCsim

Remerciements:

Tout d'abord, je remercie Dieu Tout-Puissant et Miséricordieux de m'avoir donné la force, Courage et patience pour accomplir ce modeste travail.

Après, la réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma gratitude.

Je voudrais tout d'abord adresser toute ma reconnaissance à la directrice de ce mémoire, Dr Achour SAADOUNE, qui fut la première à me faire découvrir le sujet qui a guidé mon mémoire. Pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion.

Je désire aussi remercier les professeurs de l'université, qui m'ont fourni les outils nécessaires à la réussite de mes études universitaires.

Je voudrais exprimer ma reconnaissance envers les amis et collègues qui m'ont apporté leur soutien moral et intellectuel tout au long de ma démarche.

Un grand merci à mon amie pour ses conseils concernant mon style d'écriture, ils ont grandement facilité mon travail.

Dédicace :

Je dédie cet ouvrage

A ma maman qui m'a soutenu et encouragé durant ces années d'études.

Qu'elle trouve ici le témoignage de ma profonde reconnaissance.

A mes frères, mes grands-parents et Ceux qui ont partagé avec moi tous les moments d'émotion lors de la réalisation de ce travail. Ils m'ont chaleureusement supporté et encouragé tout au long de mon parcours.

A ma famille, mes proches et à ceux qui me donnent de l'amour et de la vivacité.

A tous mes amis qui m'ont toujours encouragé, et à qui je souhaite plus de succès.

A tous ceux que j'aime

Contents

| | |
|--|----|
| Introduction général : | 1 |
| I. Chapitre : système automatisé et l'automate programmable industriel. | 2 |
| I.1 Introduction : | 3 |
| I.2 Les systèmes automatisés:..... | 3 |
| I.2.1 Définition: | 3 |
| I.2.2 Schéma d'un système automatisé :..... | 4 |
| I.2.3 Structure d'un système automatisé :..... | 4 |
| I.2.4 Objectifs de l'automatisation :..... | 8 |
| I.2.5 Les avantages de l'automatisation:..... | 9 |
| I.2.6 Types de systèmes d'automatisation : | 9 |
| I.2.7 Différents types de command :..... | 11 |
| I.3 Automate programmable (API) :..... | 12 |
| I.3.1 Définition et historique: | 12 |
| I.3.2 L'avènement de l'api :..... | 13 |
| I.3.3 Avantages de l'automate programmable industriel:..... | 13 |
| I.3.4 Type des API :..... | 14 |
| I.3.5 Nature des informations traitées par l'automate :..... | 15 |
| I.3.6 Architecture et structure des automates:..... | 16 |
| I.3.7 Traitement du programme automate: | 18 |
| I.3.8 Choix de l'API :..... | 19 |
| I.4 Conclusion :..... | 19 |
| II. Chapitre: Les Réseaux Industriels de Communication | 21 |

| | | |
|---------|--|----|
| II.1 | Introduction : | 22 |
| II.2 | Le modèle OSI : | 22 |
| II.2.1 | Définition: | 22 |
| II.3 | Différents types d'interconnexion de réseaux : | 26 |
| II.4 | Topologies des réseaux: | 27 |
| II.5 | Les bus et réseaux de terrain en automatisme industriel: | 29 |
| II.5.1 | Système de bus : | 29 |
| II.5.2 | Les avantages d'une telle mise en réseau: | 29 |
| II.5.3 | Système de bus industriel et réseaux de terrain: | 29 |
| II.6 | Hierarchie d'automatisation : | 34 |
| II.7 | Critères de comparaison entre RLI : | 36 |
| II.7.1 | Les critères techniques : | 37 |
| II.7.2 | Les critères stratégiques : | 37 |
| II.8 | Conclusion : | 38 |
| III. | Chapitre : Profinet..... | 39 |
| III.1 | Introduction : | 40 |
| III.2 | PROFINET en un coup d'œil | 41 |
| III.3 | Raisons d'utiliser PROFINET : | 42 |
| III.4 | Canaux de communication PROFINET : | 42 |
| III.5 | Composants Profinet (stations): | 44 |
| III.5.1 | I/O device : | 45 |
| III.5.2 | I/O contrôleurs : | 46 |
| III.5.3 | I/O Superviseur : | 46 |

| | | |
|----------|---|----|
| III.6 | Descriptions des appareils :..... | 47 |
| III.7 | Adressage : | 47 |
| III.8 | Profinet DCP et DHCP : | 48 |
| III.9 | LLDP et Profinet :..... | 50 |
| III.10 | Fonctions : | 50 |
| III.10.1 | Données cycliques échangé : | 50 |
| III.10.2 | Échange de données acyclique :..... | 51 |
| III.10.3 | Diagnostic appareil/réseau :..... | 51 |
| III.11 | Fonctions optionnelles : | 52 |
| III.12 | Facilité du câblage : | 53 |
| III.13 | Conclusion :..... | 54 |
| IV. | Logiciel et le travail pratique : | 55 |
| IV.1 | Introduction : | 56 |
| IV.2 | Définition de tia portal :..... | 56 |
| IV.3 | Les langages des API :..... | 56 |
| IV.4 | La vue du logiciel tia portal : | 57 |
| IV.5 | La vue du projet :..... | 58 |
| IV.6 | Création et programmation d'un projet : | 58 |
| IV.6.1 | Création :..... | 58 |
| IV.6.2 | Configuration de matériaux : | 59 |
| IV.6.3 | Adressage des E/S : | 60 |
| IV.6.4 | Programmation de projet :..... | 61 |
| IV.6.5 | Structure du programme : | 62 |

| | | |
|--------|--|----|
| IV.6.6 | Exécution du programme :..... | 63 |
| IV.6.7 | Simulation du programme :..... | 65 |
| IV.7 | Programmation et communication avec deux méthodes :..... | 66 |
| IV.7.1 | Feux de circulation : | 66 |
| IV.7.2 | Des fonctions mathématiques : | 84 |
| IV.8 | Conclusion :..... | 92 |
| | Conclusion général :..... | 93 |
| | Bibliographie et webographie :..... | 94 |

List des figures :

| | |
|---|----|
| Figure 1 Schéma d'un système automatisé. | 4 |
| Figure 2 les parties de système automatisé. | 5 |
| Figure 3 pré-actionneurs..... | 7 |
| Figure 4 des actionneurs..... | 7 |
| Figure 5 Des capteurs. | 8 |
| Figure 6 chaîne de production (ensemble de processus). | 10 |
| Figure 7 un processus. | 11 |
| Figure 8 automate compact. | 14 |
| Figure 9 Automate modulaire..... | 15 |
| Figure 10 API..... | 16 |
| Figure 11 cycle de traitement du programme automate. | 18 |
| Figure 12 Le modèle OSI. | 23 |
| Figure 13 Les couches de modèle OSI..... | 24 |
| Figure 14 Constitution d'une trame. | 26 |
| Figure 15 Différents types d'interconnexions de réseaux. | 27 |
| Figure 16 Les topologies des réseaux. | 29 |
| Figure 17 Infrastructure Ethernet prenant en charge plusieurs protocoles d'application et périphériques | 31 |
| Figure 18 Exemple d'un câblage TCP / IP, Industriel Ethernet, Profinet et Profibus. | 33 |
| Figure 19 utilisation de Profibus..... | 34 |
| Figure 20 exemple de la hiérarchie à respecter. | 34 |
| Figure 21 Les niveaux de communication préconisés par Siemens..... | 35 |
| Figure 22 Les niveaux de communication retenus par Schneider Electric..... | 36 |

| | |
|--|----|
| Figure 23 PROFINET satisfait à toutes les exigences de la technique d'automatisation. | 42 |
| Figure 24 A layer 2 Ethernet frame. | 43 |
| Figure 25 paquet non en temps réel avec des données de protocole supplémentaires. | 44 |
| Figure 26 Voies de communication pour PROFINET IO. | 45 |
| Figure 27 Les appareils et contrôleurs PROFINET échangent régulièrement des données cycliques, mais les superviseurs ne se connectent que lorsqu'ils ont besoin d'informations spécifiques. | 46 |
| Figure 28 Attribution de nom. | 48 |
| Figure 29 Extrait de la séquence de démarrage de Profit à l'aide de DCP à partir du contrôleur I/O. | 49 |
| Figure 30 communication en temps réel avec surveillance du temps de cycle. | 51 |
| Figure 31 Modèle de diagnostic pour signaler les défauts avec différentes priorités. | 52 |
| Figure 32 Plusieurs contrôleurs lus les mêmes entrées sur un appareil. | 52 |
| Figure 33 Accès par plusieurs contrôleurs à différents modules dans un dispositif. | 53 |
| Figure 34 vue de logiciel. | 57 |
| Figure 35 vue de projet. | 58 |
| Figure 36 création de projet étapes. | 59 |
| Figure 37 configuration de matériel. | 60 |
| Figure 38 : adressage E/S. | 60 |
| Figure 39 création d'un nouveau bloc de programmation. | 61 |
| Figure 40 exemple de structure de programme. | 63 |
| Figure 41 TIA portal en vue. | 64 |
| Figure 42 le bouton de visualisation. | 65 |
| Figure 43 simulateur de step7. | 65 |
| Figure 44 mode d'exécution. | 65 |

| | | |
|-----------|---|----|
| Figure 45 | vue de zone de travail..... | 67 |
| Figure 46 | adressage d'Ethernet de CPU_1. | 68 |
| Figure 47 | adressage d'Ethernet de CPU_2. | 68 |
| Figure 48 | tableaux de variables. | 70 |
| Figure 49 | le programme dans l'OB CPU_1..... | 72 |
| Figure 50 | le programme dans l'OB CPU_2..... | 74 |
| Figure 51 | DB pour CPU_1..... | 75 |
| Figure 52 | DB pour CPU_2..... | 76 |
| Figure 53 | paramètre de la liaison de fonction PUT..... | 77 |
| Figure 54 | paramètre de bloc de fonction PUT..... | 77 |
| Figure 55 | donne Access à PUT/GET..... | 79 |
| Figure 56 | DB de CPU_2..... | 79 |
| Figure 57 | DB de CPU_1..... | 80 |
| Figure 58 | PUT instruction..... | 80 |
| Figure 59 | configuration instruction PUT..... | 81 |
| Figure 60 | compilation avec succès CPU_1..... | 82 |
| Figure 61 | compilation avec succès CPU_2..... | 83 |
| Figure 62 | vue de réseau et d'adresses..... | 83 |
| Figure 63 | adressage de CPU_1..... | 85 |
| Figure 64 | adressage de CPU_2..... | 86 |
| Figure 65 | tableau des variables CPU_1..... | 86 |
| Figure 66 | tableau des variables CPU_2..... | 87 |
| Figure 67 | OB1 de CPU_1..... | 88 |
| Figure 68 | OB1 de CPU_2..... | 88 |
| Figure 69 | définir CPU_2 comme un esclave de CPU_1..... | 89 |
| Figure 70 | création des zones de transfert..... | 90 |
| Figure 71 | CPU_1 bien communiqué avec le simulateur..... | 91 |

Figure 72 CPU_2 bien communiqué avec le simulateur. 91

Figure 73 vue de réseau et d'adresses..... 92

List d'abréviation :

ISO : Organisation internationale de normalisation

OSI : Interconnexion des systèmes ouverts

MAC : Contrôle d'accès au support

API : L'automate programmable industriel

E/S : Entrées / Sorties

T.O.R : Tout ou rien

RT : temps réel

IRT : temps réel isochrone (IRT)

CBA : Component Based Automation

TIA : Totally Integrated Automation

PROFINET: Process Field Net.

PROFIBUS: Process Field Bus.

OB : Organization block.

FB : Function block.

DB : Data Block.

FC : Function.

CPU: Central Processing Unit.

Introduction général :

Les réseaux industriels sont désormais vitaux dans le monde de l'automatisation

Bien gérer l'établissement. La mise en réseau industrielle apporte une grande flexibilité systèmes de contrôle/commande, ils réduisent les coûts de câblage, il offre des possibilités L'actualité du contrôle et de la supervision des installations, pour les équipes d'exploitation Comme maintenance, production ou gestion.

Dans notre étude, nous essaierons de créer un système de feux de signalisation intelligents avec deux automates utilisant le protocole Profinet, à travers cela et dans tout le chemin, nous verrons quel est le système d'automatisation, les automates et leurs pièces et composants, à côté du réseau et Ethernet évoluant dans le domaine de la communication de l'industrie et son application, ses méthodes et son processus, ayant des idées sur les protocoles à utiliser en leur sein. Après cette précision dans le protocole Profinet, y compris la définition, les avantages et l'application dans les systèmes attachés aux autres protocoles, nous communiquons finalement les deux automates qui exécutent notre programme à l'aide de Profinet.

I. Chapitre : système automatisé et l'automate programmable industriel.

I. Chapitre : système automatisé et l'automate programmable industriel.

I. Chapitre : système automatisé et l'automate programmable industriel.

I.1 Introduction :

Un système c'est un ensemble d'éléments en interaction dynamique, organisé en fonction d'un but, L'automatisation est d'équiper une installation pour qu'elle remplisse sa fonction de totalement ou partiellement sans intervention humaine et l'automate programmable, L'objectif de l'automatisation des systèmes est de produire, en ayant recours le moins possible à l'homme, des produits de qualité et ce pour un coût le plus faible possible.

Les automates ont commencé comme des remplacements de logique de relais dans l'industrie automobile. Ils se sont étendus pour inclure des entrées et des sorties analogiques. Les API étaient à l'origine limités à l'automatisation discrète (ou d'usine). L'ajout de capacités analogiques leur a permis d'être utilisés beaucoup plus largement.

I.2 Les systèmes automatisés:

I.2.1 Définition:

Simple ou complexes, les systèmes automatisés sont partout dans notre environnement quotidien, Un système est dit automatisé s'il exécute toujours le même cycle de travail pour lequel il a été programmé après avoir reçu les consignes d'un opérateur, il est composé de:

❖ Chaîne d'information / La partie commande (PC):

Elle donne les ordres et reçoit les informations de l'extérieur ou de la partie opérative. Elle peut se présenter sous 3 manières différentes : un boîtier de commande, un microprocesseur (cerveau électronique), ou un ordinateur

❖ Chaîne d'énergie / La partie opérative (PO):

I. Chapitre : système automatisé et l'automate programmable industriel.

C'est la partie d'un système automatisé qui effectue le travail. Autrement dit, c'est la machine. C'est la partie qui reçoit les ordres de la partie commande et qui les exécute. Elle comporte les capteurs et les actionneurs. [3]

I.2.2 Schéma d'un système automatisé :

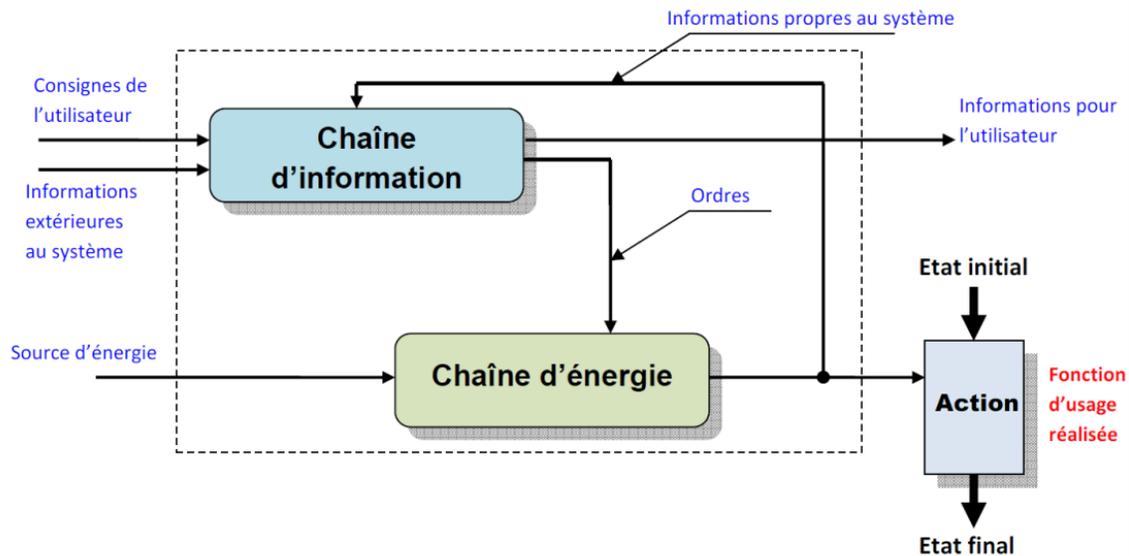


Figure 1 Schéma d'un système automatisé.

I.2.3 Structure d'un système automatisé :

Un système automatisé peut, pour faciliter l'analyse, se représenter sous la forme d'un schéma identifiant trois parties (P.O ; P.C ; P.P) du système et exprimant leurs interrelations (Informations, Ordres, Comptes rendus, Consignes). [1]

I. Chapitre : système automatisé et l'automate programmable industriel.

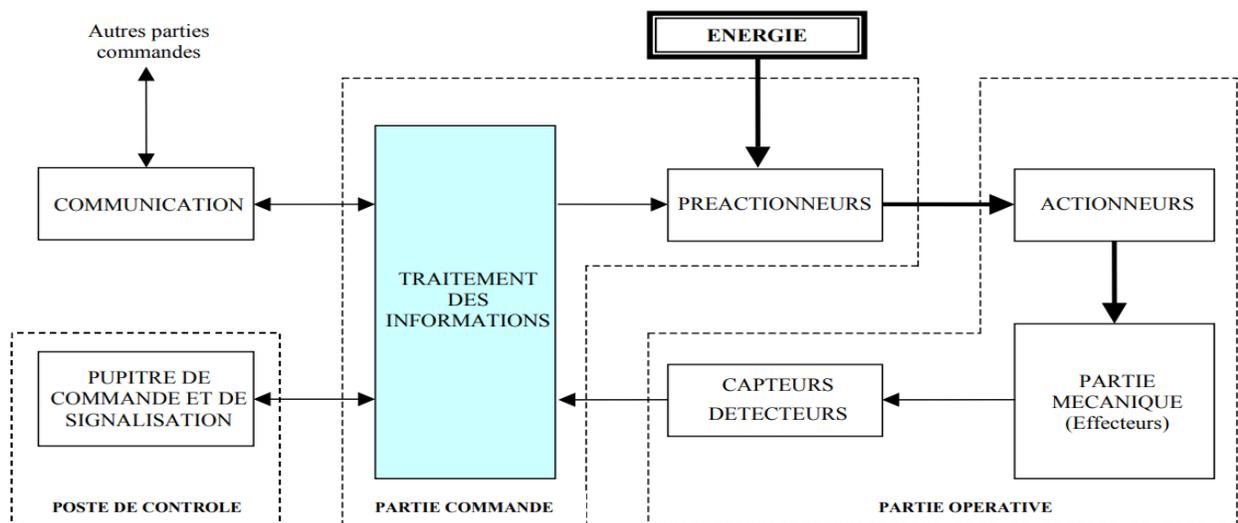
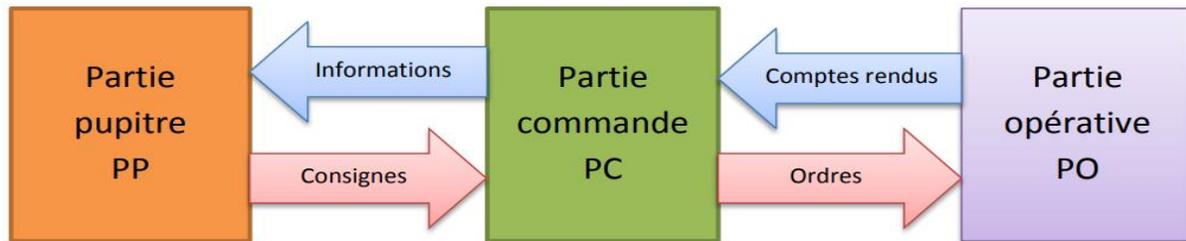


Figure 2 les parties de système automatisé.

❖ La partie commande (PC) :

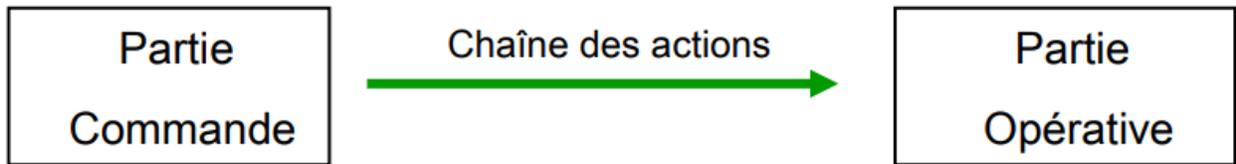
Elle est en générale composée d'ordinateurs, de mémoires et de programmes. Elle transmet les ordres aux actionneurs à partir :

- du programme qu'elle contient ;
- des informations reçues par les capteurs ;
- des consignes données par l'utilisateur ou l'opérateur.

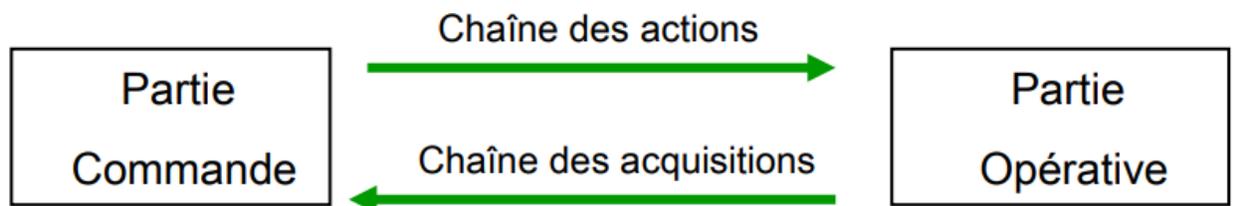
❖ Mode de command :

- Mode de commande directe (ou boucle ouverte) :

I. Chapitre : système automatisé et l'automate programmable industriel.



- Mode de commande avec compte-rendu d'exécution (ou boucle fermée) :



❖ La partie opératives (PO):

C'est la partie visible du système, Elle consomme de l'énergie électrique, pneumatique (air) ou hydraulique (eau ou huile) et elle comporte les éléments mécaniques du mécanisme avec :

- Des pré-actionneurs lesquels reçoivent des ordres de la partie command. [1]

I. Chapitre : système automatisé et l'automate programmable industriel.



Figure 3 pré-actionneurs

- Des actionneurs, qui transforment l'énergie reçue en énergie utile (vérin, moteur, voyant,...)



Figure 4 des actionneurs.

- Des capteurs, qui transforment les variations de grandeurs physiques en signaux électriques (température, luminosité, présence, position,...)
 - Capteurs mécaniques, pneumatiques ou électriques.
 - Capteurs magnétiques montés sur les vérins.
 - Capteurs pneumatiques à chute de pression. [1]

I. Chapitre : système automatisé et l'automate programmable industriel.



Figure 5 Des capteurs.

I.2.4 Objectifs de l'automatisation :

❖ Visant le personnel :

Améliorer ses conditions de travail en supprimant les tâches les plus pénibles et en augmentant la sécurité.

❖ Visant le produit :

Améliorer sa faisabilité, sa qualité par rapport au cahier des charges, sa fiabilité dans le temps.

❖ Visant l'entreprise :

Améliorer sa compétitivité (en diminuant les coûts de production), sa productivité, la qualité de production, la capacité de contrôle, de gestion, de planification. [1]

I. Chapitre : système automatisé et l'automate programmable industriel.

I.2.5 Les avantages de l'automatisation:

- ❖ Meilleure rentabilité
 - Accélère le cycle de production
 - Augmente la qualité du produit
 - Permet des économies de temps et de coût du personnel
 - Réduit l'impact environnemental grâce au fonctionnement efficace des ressources des systèmes (par ex. faible consommation de matériaux et d'énergie)
 - Augmente la flexibilité
- ❖ Augmente la fiabilité, la sécurité et le cycle de vie des systèmes de production
 - Transport d'objets lourds comme les éléments de carrosserie dans la construction automobile au moyen d'aide au levage
 - Inspection automatisée et analyse d'image (par ex. images infrarouges ou radiographies)
 - Précision améliorée et évitement des erreurs
- ❖ Conditions de vie et de travail améliorées
 - Soulage les êtres humains d'un travail mentalement peu exigeant, monotone, pénible, dangereux
 - Utilisation commode. [2]

I.2.6 Types de systèmes d'automatisation :

- ❖ Automatisation de la production:
 - Objectif : contrôle des processus

I. Chapitre : système automatisé et l'automate programmable industriel.

- Se base sur le PLC (contrôleur logique programmable)
- Capture les données du capteur toutes les 10 à 100 ms
- Se divise en production série et individuelle
- Ici, les produits finaux sont créés à partir de nombreuses matières premières, matériels et pièces achetées en externe.
- De nombreux processus de production et d'assemblage sont fréquemment requis.
- Les processus de production sont décrits à l'aide des plannings de travail (spécification des étapes de production et d'assemblage) et des listes de pièces (indiquant les composants individuels d'un produit). [2]

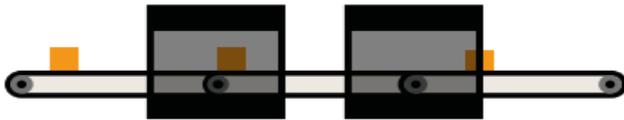


Figure 6 chaîne de production (ensemble de processus).

❖ Automatisation de processus :

- Objectif : régulation des processus.
- Se basait auparavant purement sur PCS (système de contrôle de processus), de nos jours se base également sur PLC à des niveaux de complexité faible à moyen dans certains cas.
- En raison de la structure généralement très décentralisée, ils sont appelés systèmes de contrôle distribués (DCS).

I. Chapitre : système automatisé et l'automate programmable industriel.

- Capture des données du capteur toutes les 100 ms à plusieurs secondes.
- Essentiellement l'automatisation des opérations liées au processus et des réactions chimiques telles que le mélange, le chauffage, la séparation ou la synthèse.
- Une distinction est faite entre production continue, discontinue et par campagne. [2]

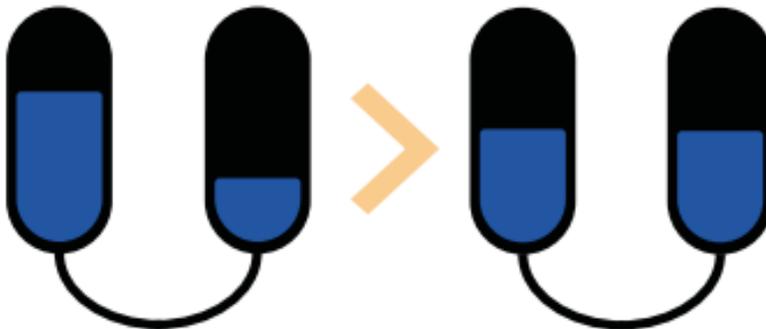


Figure 7 un processus.

I.2.7 Différents types de command :

- ❖ Système automatisé combinatoire:

Ces système n'utilisent aucun mécanisme de mémorisation (ils n'ont pas de mémoire) et à une combinaison des entrées correspond une seule combinaison des sorties. La logique associée est appelée logique combinatoire. Les outils utilisés pour les concevoir sont l'algèbre de Boole, les tables de vérité, les tableaux de KARNAUGH.

- ❖ Système automatisé séquentiel:

I. Chapitre : système automatisé et l'automate programmable industriel.

Ces systèmes sont les plus répandus sur le plan industriel. Le déroulement du cycle s'effectue étape par étape. À une situation des entrées peut correspondre plusieurs situations de sorties. La sélection d'une étape ou d'une autre dépend de la situation antérieure du dispositif. La logique associée est appelée « logique séquentielle ».

Elle peut être:

- avec commande pneumatique ou électrique : logique câblée.
- avec commande électronique : logique programmée. [1]

I.3 Automate programmable (API) :

I.3.1 Définition et historique:

L'ingénierie de contrôle a évolué au fil du temps. Dans le passé, les humains étaient la principale méthode de contrôle d'un système. Plus récemment, l'électricité a été utilisée pour le contrôle et les premiers contrôles électriques étaient basés sur des relais. Ces relais permettent d'allumer et d'éteindre l'alimentation sans interrupteur mécanique. Il est courant d'utiliser des relais pour prendre des décisions de contrôle logiques simples l'avènement de l'api a commencé en 1970 et est devenu le choix le plus courant pour les contrôles de fabrication.

Un Api ou PLC (Programmable Logic Controller) en anglais est une machine électronique programmable par du personnel non informatique et destiné à piloter en milieu industriel et dans le temps véritables procédés industriels constitués de composants électroniques, un automate programmable est un calculateur logique, ou ordinateur, doté d'un ensemble d'instructions délibérément réduit, destiné au contrôle et à la surveillance des processus en temps réel fabricants Son objectif principe est de rendre l'ensemble du mécanisme du type " laisser-faire-seul" : le système contrôle son

I. Chapitre : système automatisé et l'automate programmable industriel.

sorties, décide et agit sur ses entrées pour maintenir le fonctionnement comme prévu par l'utilisateur. C'est le principe de l'automatisation.

I.3.2 L'avènement de l'api :

- ❖ problèmes de contrôle de relais :
 - la complexité de la configuration du circuit.
 - la fiabilité du contact des contacts de relais est vulnérable.
 - dispositif de contrôle plus grand.
 - modifier le contenu du contrôle et compléter ces difficultés.

- ❖ demandes de la société GM :
 - programmer modifier et changer la séquence devrait être facile.
 - la maintenance doit être facile et utiliser la méthode plug-in.
 - plus petit que la commande de relais.
 - fiabilité supérieure à la commande de relais ...etc. [4]

I.3.3 Avantages de l'automate programmable industriel:

- Rentable pour Contrôler des systèmes complexes.
- Flexible et peut être réappliqué pour contrôler d'autres systèmes.
- Sauvegarde les programmes.
- Suivi temps réel l'évolution de l'automatisme.
- Simplification du câblage.
- La fiabilité.

I. Chapitre : système automatisé et l'automate programmable industriel.

- E La maintenance et le dépannage possible par des techniciens de formation électromécanique.

I.3.4 Type des API :

❖ Compact :

Les automates compacts permettent de commander des sorties en T.O.R et gèrent parfois des fonctions de comptage et de traitement analogique.



Figure 8 automate compact.

❖ Modulaire :

Les automates modulaires permettent de réaliser de nombreuses autres fonctions grâce à des modules intelligents que l'on dispose sur un ou plusieurs racks.

I. Chapitre : système automatisé et l'automate programmable industriel.



Figure 9 Automate modulaire.

I.3.5 Nature des informations traitées par l'automate :

Les informations peuvent être de type:

❖ **Tout ou rien (T.O.R.):**

L'information ne peut prendre que deux états (vrai/faux, 0 ou 1 ...). C'est le type d'information délivrée par un détecteur, un bouton poussoir ...

❖ **Analogique:**

L'information est continue et peut prendre une valeur comprise dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (pression, température ...)

❖ **Numérique:**

L'information est contenue dans des mots codés sous forme binaire ou bien hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent. [5]

I. Chapitre : système automatisé et l'automate programmable industriel.

I.3.6 Architecture et structure des automates:

- ❖ Aspect extérieur :
 - Rack
 - Module d'alimentation (ps : power supply)
 - Unité central
 - Les modules des entres et sortie TOR
 - Les modules des entres et sortie analogique
 - Pile de sauvegarde
 - Le Coupleur
 - Connexion au 24V cc
 - LED de signalisation d'état et de défauts
 - Carte mémoire
 - Porte de connexion Ethernet [5]



Figure 10 API.

I. Chapitre : système automatisé et l'automate programmable industriel.

❖ Aspect interne :

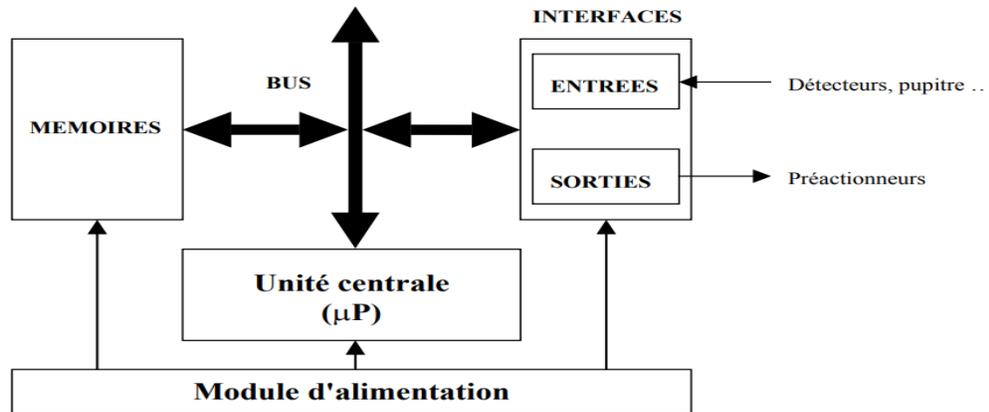


Figure: structure interne d'un api

- Unité centrale de traitement (cpu) : à base de microprocesseur, elle réalise toutes les fonctions logiques, arithmétiques et de traitement numérique (transfert, comptage, temporisation ...)
- Le bus interne : il permet la communication de l'ensemble des blocs de l'automate et des éventuelles extensions
- Mémoires : Elles permettent de stocker le système d'exploitation (ROM ou PROM), le programme (EEPROM) et les données système lors du fonctionnement (RAM). Cette dernière est généralement secourue par pile ou batterie.
- Interface d'entrée : elle permet de recevoir les informations du S.A.P. ou du pupitre et de mettre en forme (filtrage, ...) ce signal tout en l'isolant électriquement.
- Interface de sortie : elle permet de commander les divers pré-actionneurs et éléments de signalisation du S.A.P. tout en assurant l'isolement électrique. [5]

I.3.7 Traitement du programme automate:

Tous les automates fonctionnent selon le même mode opératoire :

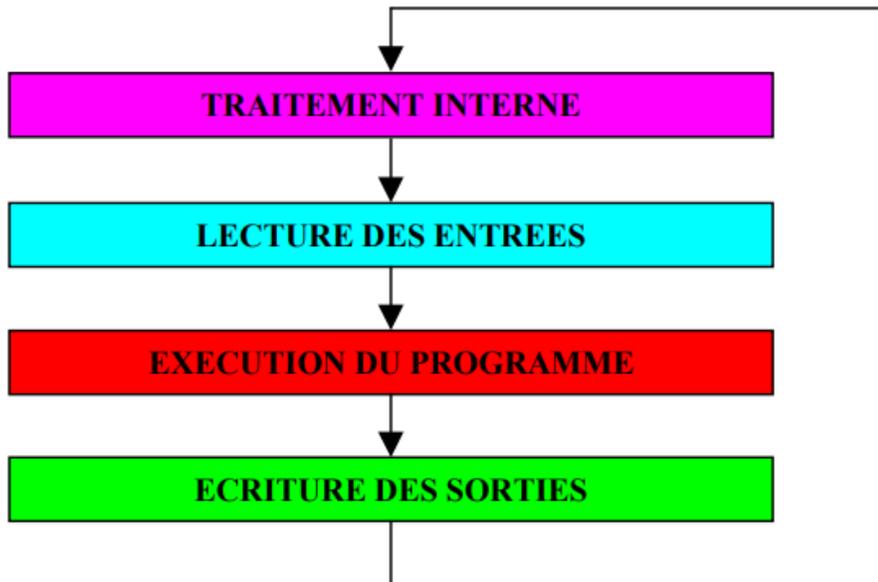


Figure 11 cycle de traitement du programme automate.

❖ **Traitement interne :**

L'automate effectue des opérations de contrôle et met à jour certains paramètres systèmes (détection des passages en RUN / STOP, mises à jour des valeurs de l'horodateur, ...).

❖ **Lecture des entrées :**

L'automate lit les entrées (de façon synchrone) et les recopie dans la mémoire image des entrées.

❖ **Exécution du programme :**

I. Chapitre : système automatisé et l'automate programmable industriel.

L'automate exécute le programme instruction par instruction et écrit les sorties dans la mémoire image des sorties.

❖ **Ecriture des sorties :**

L'automate bascule les différentes sorties (de façon synchrone) aux positions définies dans la mémoire image des sorties. [5]

I.3.8 Choix de l'API :

- Le type des entrées/sorties nécessaire.
- Nombres d'entrées/sorties intégrés.
- Qualité de processeur pour le traitement.
- Capacité de la mémoire.
- Nombre de compteurs.
- Nombre de temporisateur
- Détermination de l'emplacement de l'A.P.I

I.4 Conclusion :

L'avènement des automates fait savoir aux groupes industriels que leur développement conduira à une révolution industrielle qui conduira à son tour résolument à de nouveaux systèmes puissants créatifs, et nous avons discuté des types d'entre eux, de leur composant, de l'architecture et de la nature des données et des entrées ils traitent. Faire un système d'automatisation complet

II. Chapitre : Les Réseaux Industriels de communication.

II. Chapitre: Les Réseaux Industriels de Communication

II. Chapitre : Les Réseaux Industriels de communication.

II.1 Introduction :

Avec l'adoption progressive de standards communs par les deux mondes, la frontière entre automatisme et informatique tombe. La communication entre ces deux mondes converge grâce à l'adoption de protocoles standard mondiaux Ethernet et TCP/IP, mais aussi grâce à la prise en compte de mécanismes normés tels qu'OPC. Ces nouvelles technologies, associées aux Extranet, Intranet et Internet, autorisent un accès aux données de l'automatisme en temps réel, en tout lieu, à toute personne autorisée. Cette convergence est renforcée par l'arrivée de nombreux constituants d'automatismes basés sur cette technologie. Le concept "Transparent Factory" de Schneider Electric s'appuie sur toutes ces nouvelles technologies de l'information et de la communication (NTIC).

Un certain nombre de tendances ont poussé les utilisateurs vers une plus grande mise en œuvre de l'Ethernet industriel. Ceux-ci inclus: Contrôle : de centralisé à distribué, Réseaux : de plusieurs à un, Connexions : du filaire au sans fil, Technologies de l'information : de l'ignorance à l'exploitation, Réseaux d'usines : du propriétaire à l'ouvert. [10]

II.2 Le modèle OSI :

II.2.1 Définition:

Un protocole de communication est une spécification de plusieurs règles pour un type de communication particulier.

Le modèle OSI (Open System Interconnexion) a été créé par l'ISO (Organisation internationale de normalisation) qui a édité la norme ISO 7498 dans le but d'offrir une base commune à la description de tout réseau informatique. [9]

II. Chapitre : Les Réseaux Industriels de communication.

- ❖ **Fonctionnellement:** l'ensemble des sept couches est divisible en deux familles bien distinctes : la première, constituée des couches 1 à 4, offre les services de communication (transfert des données), la seconde constituée des couches 5 à 7, les services d'application (utilisation des données). [7]

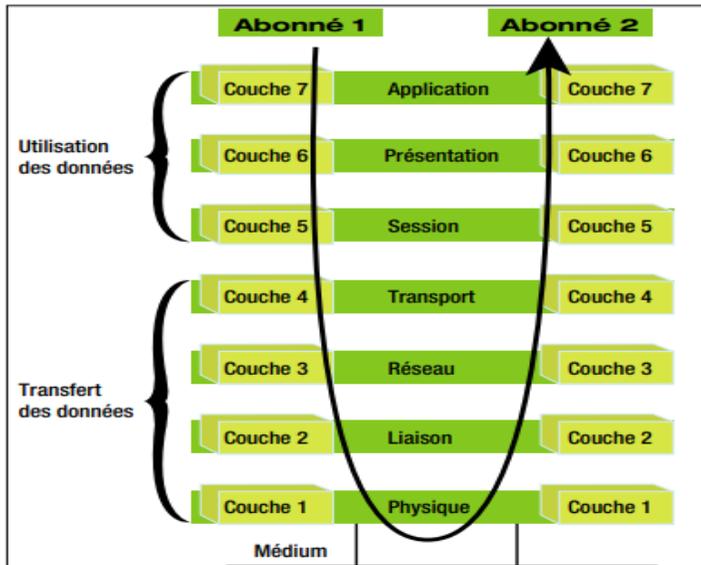


Figure 12 Le modèle OSI.

II. Chapitre : Les Réseaux Industriels de communication.

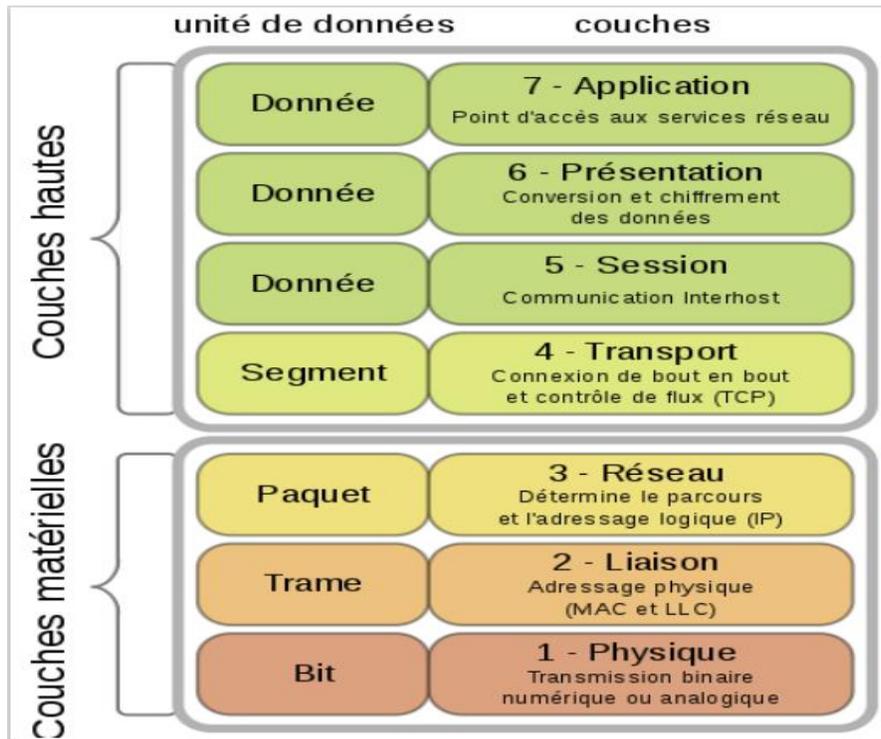


Figure 13 Les couches de modèle OSI.

- ❖ Chaque couche supporte un protocole indépendamment des autres couches,
- ❖ Chaque couche procure des services à la couche immédiatement supérieure,
- ❖ Chaque couche requiert les services de la couche immédiatement inférieure,
- ❖ La couche 1 décrit le médium (le support de communication),
- ❖ La couche 7 procure des services à l'utilisateur ou à une application.

Rappel succinct des différentes couches :

- La couche 1 (couche physique) : Elle décrit les règles mécaniques et électriques d'accès au média (représentation physique des données, encodage, vitesse et type de transmission, type de média et topologie (maillage)).

II. Chapitre : Les Réseaux Industriels de communication.

- La couche 2 (la liaison de données) : Elle détecte et corrige les erreurs de transmission, établit les connexions logiques entre les entités.
 - La couche 3 (le réseau) : Elle définit les mécanismes du routage c'est à dire de transfert des paquets d'information d'une station à l'autre (possède un répertoire des adresses des points du réseau, gère l'adresse réseau de son système, définit en collaboration avec les autres nœuds les circuits de dialogue utilisés, établit la relation entre la destination des données et les liaisons de son système).
 - La couche 4 (le transport) : C'est la frontière entre le monde de la transmission (couches 1, 2 et 3) et celui de l'application (couches 5, 6 et 7). Elle définit les protocoles permettant de garantir les transferts de messages "longs" avec un maximum de sécurité (détection de perte d'information, segmentation des messages, contrôle des erreurs, contrôle de flux).
 - La couche 5 (la session) : Elle offre les moyens d'organiser et de synchroniser le dialogue entre abonnés.
 - La couche 6 (la représentation) : Elle présente les données dans un format reconnaissable par l'application (conversion de codes ou de formats de données, sélection de la syntaxe, compression et cryptage des données).
 - La couche 7 (l'application) : Son rôle est de fournir aux applications résidentes tous les moyens de dialogue. Il s'agit par exemple d'une interface utilisateur.
- ❖ **La trame (frame):** La trame est l'ensemble des informations transmises en un seul bloc via un réseau. Elle est également appelé paquet. Chaque trame respecte la même organisation de base et contient des informations de contrôle, telles que les

II. Chapitre : Les Réseaux Industriels de communication.

caractères de synchronisation, les adresses de station, une valeur de contrôle d'erreur, ainsi qu'une quantité variable de données. [9]

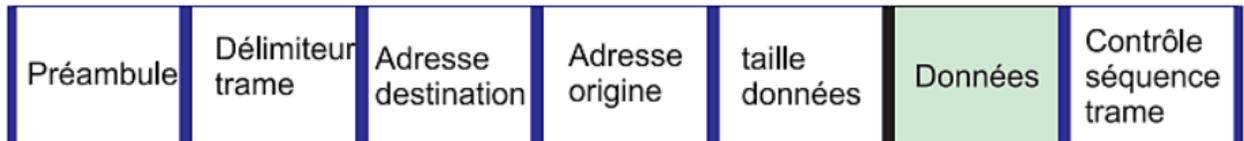


Figure 14 Constitution d'une trame.

II.3 Différents types d'interconnexion de réseaux :

- ❖ **Gateway** : Unité fonctionnelle qui permet l'interconnexion de deux réseaux d'architecture différente.
- ❖ **Routeur** : Il crée une segmentation logique de réseaux. Il assure le passage de l'information entre deux sous-réseaux logiques distincts en choisissant le meilleur chemin. C'est la couche réseau qui assure ce routage. Il n'est pas transparent, il faut donc l'adresser pour le traverser.
- ❖ **Switch** : Il transmet les données reçues sur un port, seulement vers le port sur lequel la station destinatrice est connectée. Il assure la prolongation du support au-delà des limites en distance du standard (segment) en réalisant une remise en forme des signaux. Il supprime les collisions et les paquets non valides et réduit la charge moyenne sur le réseau entier. Syn : Bridge.
- ❖ **Hub** : Les données reçues sur un port sont envoyées à tous les autres ports. Le hub ne possède pas de mémoire interne et diffuse les collisions ; plus il y a d'équipements, plus il y a de collisions et plus la charge est importante. Pour de plus longues distances, utiliser des switches. Synonyme : répéteur. [9]

II. Chapitre : Les Réseaux Industriels de communication.

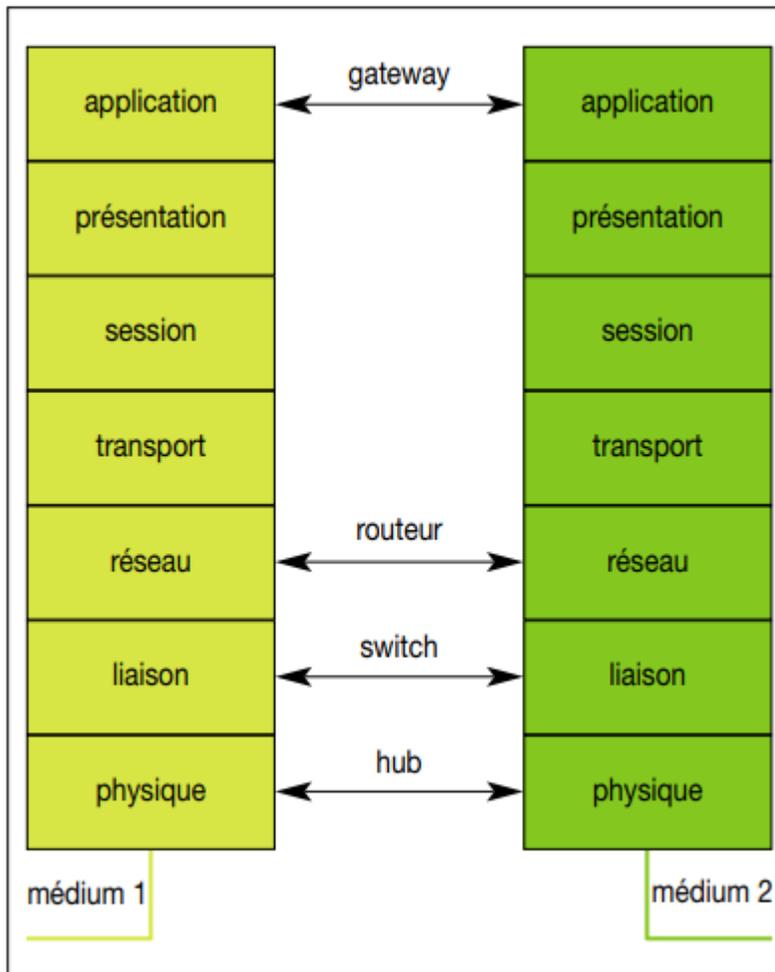


Figure 15 Différents types d'interconnexions de réseaux.

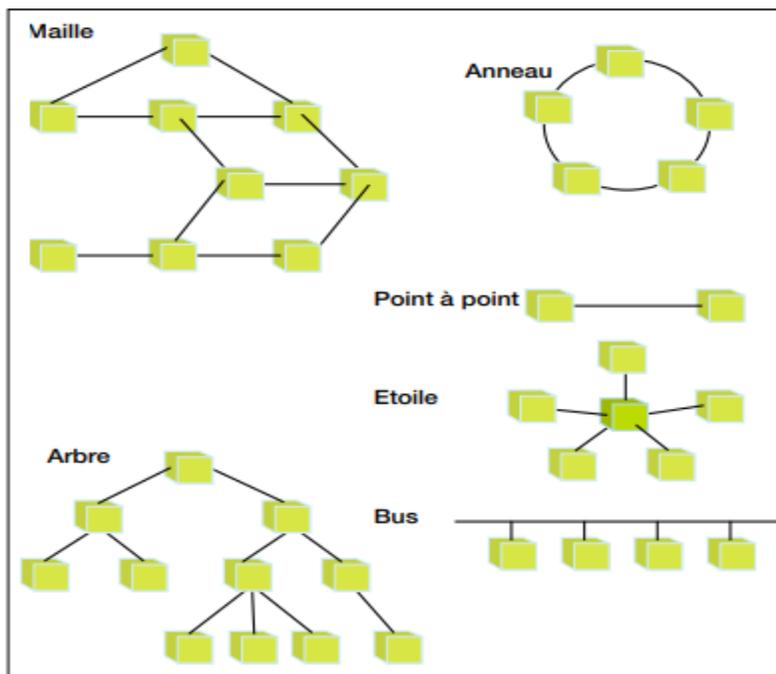
II.4 Topologies des réseaux:

La topologie d'un réseau est caractérisée par le système de câblage du réseau ; c'est la partie physique du réseau.

- ❖ Etoile : La structure en étoile se caractérise par un commutateur central avec des liaisons individuelles vers tous les terminaux du réseau. Le réseau en étoile convient aux applications à forte densité d'équipements et faible rayon d'action, par exemple de petites cellules de fabrication ou des machines de production.

II. Chapitre : Les Réseaux Industriels de communication.

- ❖ Maille : C'est la forme classique d'organisation WAN (World Access Network). Les équipements sont reliés entre eux pour former une toile (Web) d'araignée (comme Internet). Chaque nœud a un rôle de routeur. Pour atteindre un nœud, les chemins sont multiples et choisis en fonction de critères comme la disponibilité d'un nœud ou d'un WAN, la qualité de transmission ou la charge ponctuelle sur un tronçon WAN donné.
- ❖ Anneau : Il suffit de refermer les extrémités d'une ligne par une liaison supplémentaire pour obtenir une structure en anneau. Cette topologie s'utilise pour les systèmes exigeant une haute disponibilité et une protection contre les coupures de ligne ou les défaillances des constituants de réseau.
- ❖ Arbre : Un arbre a plusieurs composants de distribution, en fonction de sa taille, et peut donc être considéré comme une étoile «étendue». Un exemple de cette typologie est le type commun de réseau Ethernet de bureau utilisant des commutateurs comme composant de distribution.



II. Chapitre : Les Réseaux Industriels de communication.

Figure 16 Les topologies des réseaux.

II.5 Les bus et réseaux de terrain en automatisme industriel:

II.5.1 Système de bus :

- Dans les systèmes d'automatisation complexe, les commandes réparties dans l'installation doivent être reliées entre elles. On utilise des systèmes de bus normalisés pour relier en réseau les différentes commandes. [9]

II.5.2 Les avantages d'une telle mise en réseau:

L'échange des données entre les différentes commandes est rapide et simple.

Il est possible de construire des commandes avec des entrées et sorties décentralisées.

Un accès à distance par des systèmes maîtres aux différentes commandes est possible.

Le travail de câblage par rapport aux commandes conventionnelles est nettement inférieur.

Les visualisations, signalisations de défauts, etc. peuvent être traitées et archivées de manière centralisée. [9]

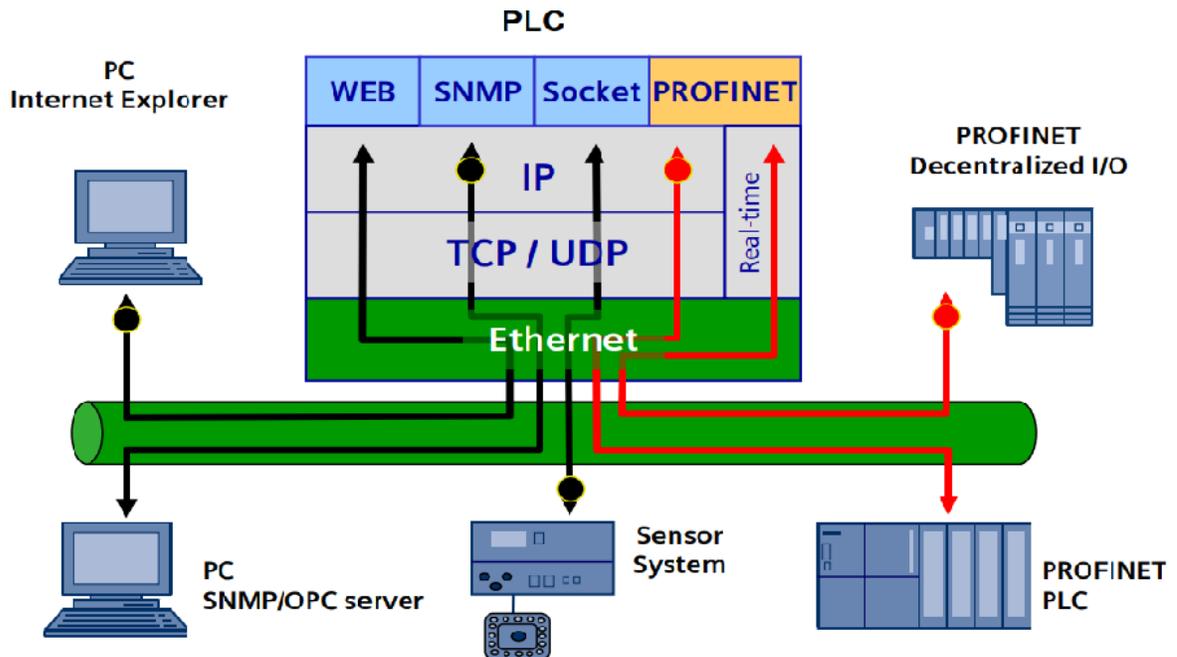
II.5.3 Système de bus industriel et réseaux de terrain:

❖ Ethernet TCP / IP:

Ethernet sans protocole de communication n'est pas un réseau mais un médium normé. Il concerne les couches Liaison (2) et Physique (1) du modèle OSI. Bien qu'il existe une différence entre Ethernet et IEEE 802.3, par abus de langage, Ethernet est assimilé à la norme 802.3. Il est à noter qu'Ethernet TCP/IP est un réseau non déterministe, Aujourd'hui imposé comme standard utilisé dans l'administration. Il est conçu pour l'échange des données dans un réseau local (LAN) et peut également

II. Chapitre : Les Réseaux Industriels de communication.

communiquer avec d'autres réseaux ainsi qu'avec Internet. Pour l'identification des abonnés du réseau, une adresse IP univoque est attribuée à chaque station d'abonné. Le protocole TCP / IP peut échanger des données au-delà des limites de son réseau. Au moyen de commutateurs ou de concentration (hub), il est possible de créer des sous réseaux (subnet).



II. Chapitre : Les Réseaux Industriels de communication.

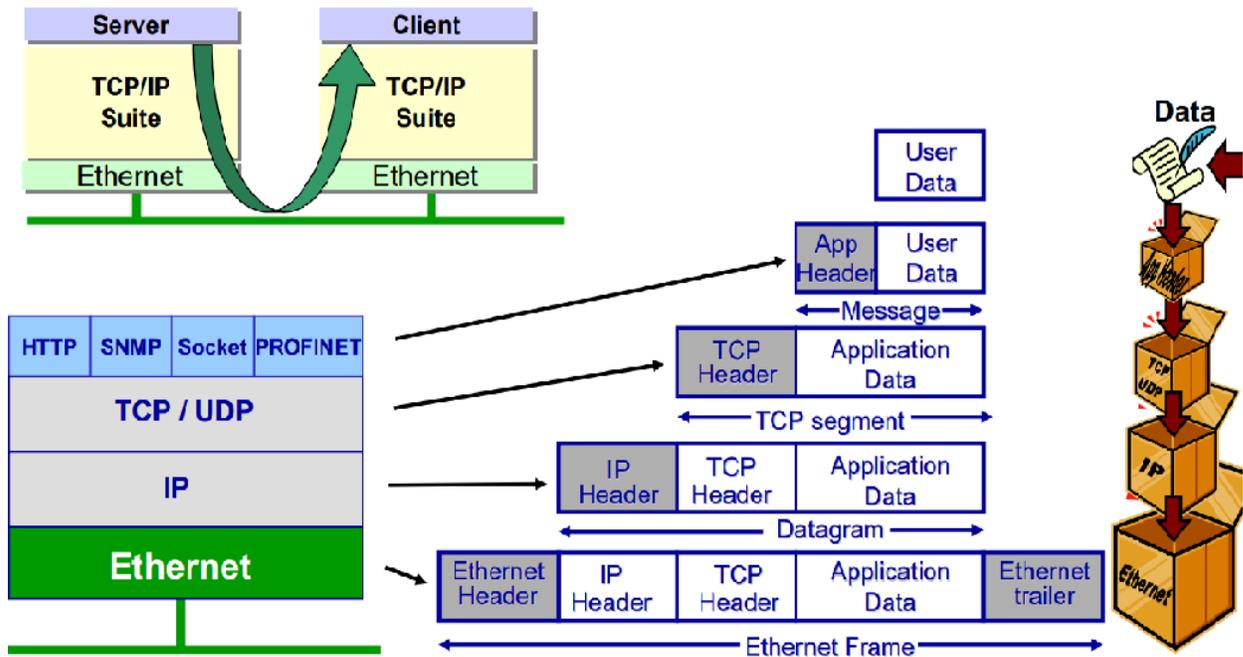


Figure 17 Infrastructure Ethernet prenant en charge plusieurs protocoles d'application et périphériques

❖ Interface actionneurs – capteurs AS-i :

En technique d'automatisation, l'interface actionneurs – capteurs (ASI ou Actuator-Sensor Interface) est le bus de terrain classique et représente le premier niveau d'un système d'automatisation. Le système AS-I relie les actionneurs et les capteurs à un système de bus, le travail de câblage est fortement réduit. De plus, cette interface permet de supprimer les cartes d'entrées et de sorties avec leurs réglettes à bornes. Le bus AS-I étant un bus série, il permet de diminuer le nombre de câbles nécessaire à l'utilisation d'un grand nombre de capteurs ou de pré-actionneurs : ces derniers sont reliés directement ou par l'intermédiaire d'une embase à l'automate central via un seul câble. Chaque embase permet de connecter jusqu'à 4 capteurs ou pré-actionneurs au bus. [9]

II. Chapitre : Les Réseaux Industriels de communication.

❖ Réseau Fipway :

Le réseau Fipway est un réseau local industriel assurant la communication entre les différents automates programmables Micro, Premium et TSX Série 7. Il sert de bus de synchronisation entre automates, il est dérivé de la norme FIP et inclut de nombreux services complémentaires. [7]

❖ Modbus (MODicon BUS) :

est un protocole de communication série initialement publié par Modicon (aujourd'hui Schneider Electric) en 1979 pour une utilisation avec ses automates programmables industriels. Modbus est devenu un protocole de communication standard et désormais un moyen couramment disponible pour connecter des appareils électroniques industriels. Modbus est maintenant un protocole ouvert et largement accepté du domaine public (Open, Public-Domain Protocol) qui nécessite une licence, mais pas de paiement de redevance à son propriétaire.

- MODBUS: est un protocole de messagerie de couche application pour la communication client / serveur entre périphériques connectés sur différents types de bus ou de réseaux. [9]

❖ Profinet :

Profinet représente une extension d'Ethernet TCP / IP et permet de communiquer jusqu'au niveau du terrain. Complété d'un protocole industriel, Ethernet classique a fait l'objet de nouveaux développements qui ont abouti au protocole PROFINET (PROcess Field EtherNET). Les données des applications en temps réel standard sont transférés avec le protocole standard TCP / IP. Grâce au protocole supplémentaire, les

II. Chapitre : Les Réseaux Industriels de communication.

données d'entrée et de sortie peuvent être traitées beaucoup plus rapidement, le Profinet est de ce fait capable de traiter les données en temps réel. [9]

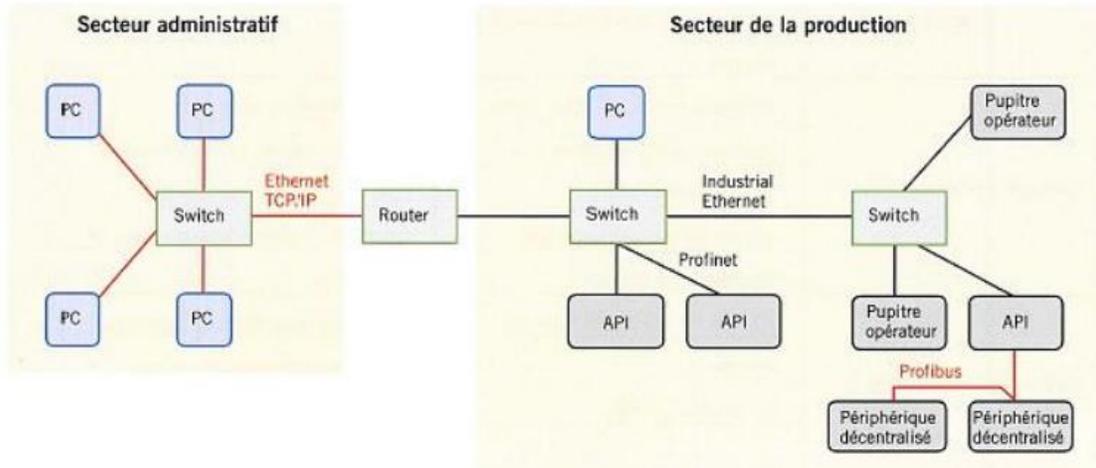
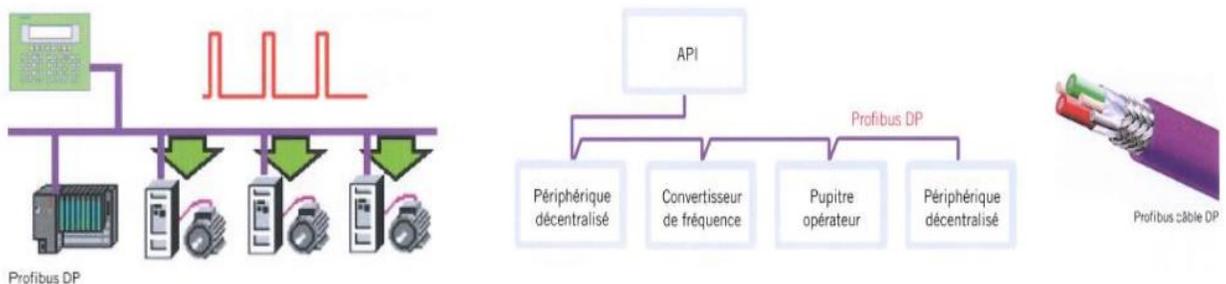


Figure 18 Exemple d'un câblage TCP / IP, Industriel Ethernet, Profinet et Profibus.

❖ Profibus DP (Périphérique Décentralisé) :

Établit une liaison directe entre l'API maître et des périphériques décentralisés (esclaves DP). Il peut s'agir d'écrans tactiles, de modules d'entrées et de sorties ou encore de convertisseurs de fréquence. Profibus DP permet un échange rapide des données entre les différentes stations. En raison des signaux de transmissions à haute fréquence, chaque extrémité de ligne doit être munie d'une résistance de terminaison, ce dernier est directement commutable sur le connecteur de bus. [9]



II. Chapitre : Les Réseaux Industriels de communication.

Figure 19 utilisation de Profibus.

II.6 Hiérarchie d'automatisation :

Afin d'assurer une mise en réseau judicieuse et efficace des différents éléments de commande, une hiérarchie et un ordre stricts doivent être respectés

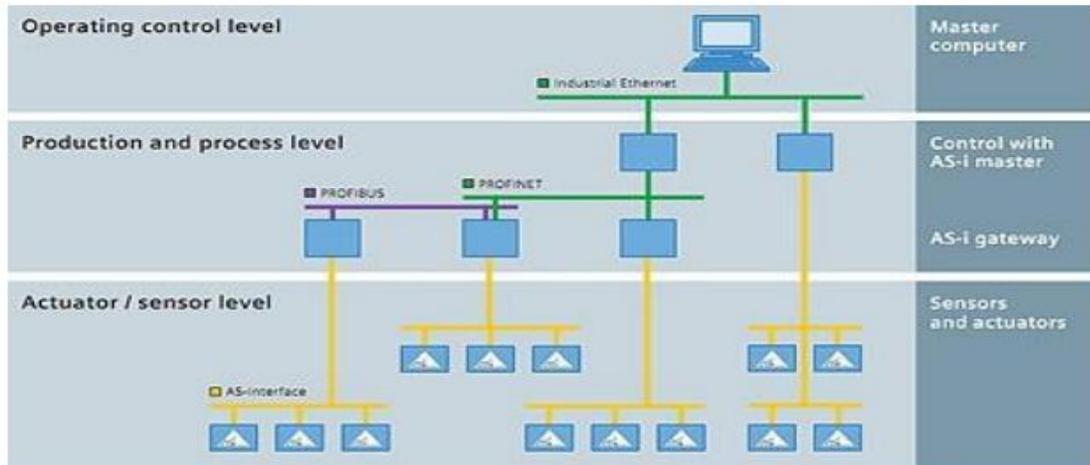


Figure 20 exemple de la hiérarchie à respecter.

II. Chapitre : Les Réseaux Industriels de communication.

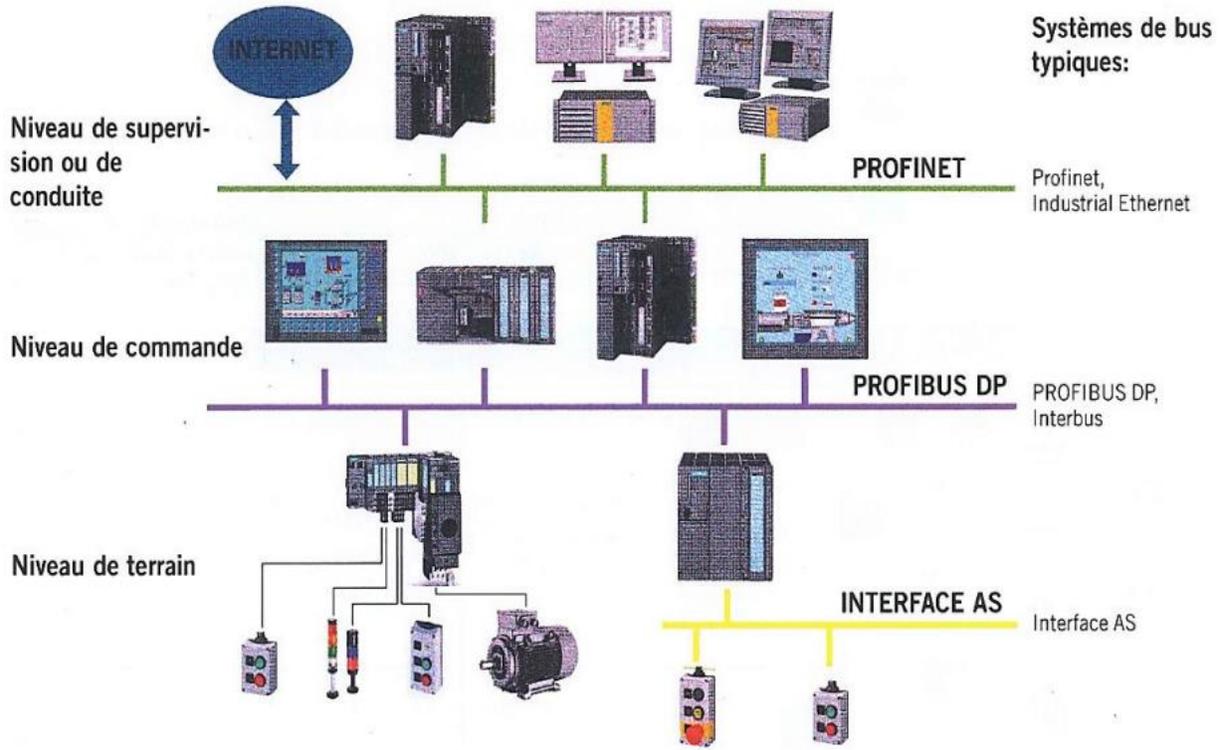
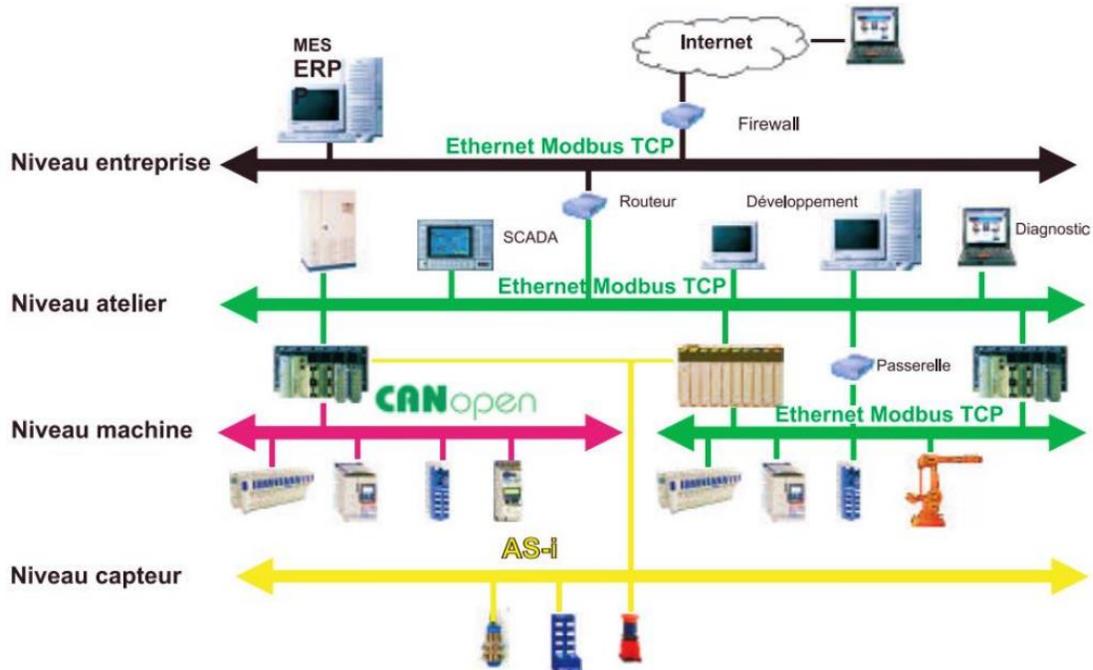


Figure 21 Les niveaux de communication préconisés par Siemens.



II. Chapitre : Les Réseaux Industriels de communication.

Figure 22 Les niveaux de communication retenus par Schneider Electric.

❖ **Le niveau de supervision ou de conduite:**

se trouve tout en haut de la pyramide d'automatisation et contrôle l'ensemble du système automatisé. Il fournit au niveau des différentes cellules de production (niveau des cellules) les informations de commande et de programmation, et décide avec quelle précision la production doit être effectuée. A ce niveau, on trouve l'ordinateur de supervision ainsi que les calculateurs pour l'ingénierie, l'interprétation, la commande et la collecte d'informations.

❖ **Au niveau commande:**

Les différentes unités de production (cellules), pilotées par un API, sont reliées entre elles. Sa mission principale est la communication ciblée entre systèmes intelligents.

❖ **Au niveau terrain:**

On trouve les appareils programmés pour la commande, le réglage et la surveillance des systèmes API et/ou ordinateurs industriels connectés. Ces derniers interprètent les signaux du niveau des capteurs / actionneurs et transfèrent les paquets de données avec des temps de réaction critiques aux systèmes du niveau supérieur. [9]

II.7 Critères de comparaison entre RLI :

Le CIAME (Comité Interprofessionnel pour l'Automatisation et la Mesure) propose un ensemble de critères (46) pour la comparaison des RLI. Ils sont répartis en deux sous-groupes : [7]

II.7.1 Les critères techniques :

❖ Topologiques

- Longueur maximale : Longueur maximum du réseau en fonction du nombre de répéteurs et du type de médium utilisé.
- Topologie : Architecture physique et implantation des nœuds connectés au réseau, structure de câblage de toutes les stations.

❖ Temporels

- Vitesse de transmission : Vitesse de transmission physique maximale possible pour le réseau. Différente du débit réel dépendent de l'efficacité du protocole.
- Temps de réaction maximal : Délai maximal possible qui peut survenir lors de l'envoi d'informations. Ce temps dépend du temps de cycle, du nombre d'abonnés, de la longueur du réseau, du médium et de la vitesse physique de transmission.

❖ Autres

- Nombre maximum d'équipements : Nombre d'équipements qui peuvent être connectés au réseau. Il dépend en partie du nombre de répéteurs utilisés.
- Efficacité du protocole : $LU \backslash LT \times 100\%$ avec LU=Longueur des données Utiles, LT=Longueur Totale du message (données utiles + bits de trame)
- Détection d'erreurs : Mécanisme de détection d'erreurs (parité, CRC...)

II.7.2 Les critères stratégiques :

❖ Standards

II. Chapitre : Les Réseaux Industriels de communication.

- Couches OSI : Définition des couches du modèle de référence OSI utilisé.
 - Certification : Entité responsable de la réalisation de tests et de la certification (tests des produits développés pour assurer la compatibilité.
- ❖ **Disponibilité de composants, de logiciels et de prestation de services**
- Composants : Disponibilité des composants existants, spécifications des fabricants.
- ❖ **Autres**
- Diffusion, nombre d'installations : Nombre d'installations en fonctionnement.
 - Perspectives pour l'avenir : Présomption empirique du rôle joué par le réseau dans l'avenir. [7]

II.8 Conclusion :

Pour résumer l'ensemble de l'étude de ce que nous avons sous la main, ce saut technique nous permet réellement de faire évoluer nos réseaux industriels dans tous leurs sens selon simplicité, sécurité, efficacité et stabilité constante du système.

III. Chapitre : Profinet.

III.1 Introduction :

Dans la technologie d'automatisation d'aujourd'hui, Ethernet et les technologies de l'information (TI) appellent de plus en plus les clichés avec des standards établis comme TCP/ IP et XML. Intégration des technologies de l'information dans l'automatisation s'ouvre nettement mieux options de communication entre l'automatisation systèmes, configuration et diagnostic étendus possibilités et fonctionnalité de service à l'échelle du réseau. Ces fonctions font partie intégrante de PROFINET dès le départ. PROFINET est le standard ouvert innovant pour l'Ethernet industriel. PROFINET satisfait tous exigences de la technologie d'automatisation.

Si l'application implique la production automatisée, automatisation de processus ou entraînements (avec ou sans sécurité fonctionnelle), PROFINET est le premier choix à tous les niveaux. En tant que technologie qui est la norme dans l'industrie automobile, Pour les fabricants d'installations et de machines, l'utilisation de PROFINET minimise les coûts d'installation, l'ingénierie et la mise en service. Pour les propriétaires de plantes, PROFINET offre une extension facile de l'installation et une disponibilité de l'installation grâce à un fonctionnement autonome unités d'usine et de faibles besoins d'entretien. La certification obligatoire pour les appareils PROFINET assure également un standard de qualité élevé. [11]

III.2 PROFINET en un coup d'œil

PROFINET est le standard de communication pour automatisation de PROFIBUS & PROFINET International (PI). La gamme de fonctions modulaire fait de PROFINET une solution flexible pour toutes les applications et tous les marchés. Avec PROFINET, les applications peuvent être réalisées pour l'automatisation de la production et des processus, de sécurité applications, et toute la gamme d'entraînement technologie jusque et y compris isochrone applications de contrôle de mouvement. Profils d'application permettent une utilisation optimale de PROFINET dans tous les domaines génie de l'automatisation. [11]

PROFINET est un mécanisme d'échange de données entre contrôleurs et appareils. Les contrôleurs peuvent être des API, des DCS ou des PAC (contrôleurs logiques programmables, systèmes de contrôle distribués ou contrôleurs d'automatisation programmables). Les appareils peuvent être des blocs d'E/S, des systèmes de vision, des lecteurs RFID, des lecteurs, des instruments de processus, des proxys ou même d'autres contrôleurs. Rapide et déterministe PROFINET échange des données rapidement et de manière déterministe. [12]

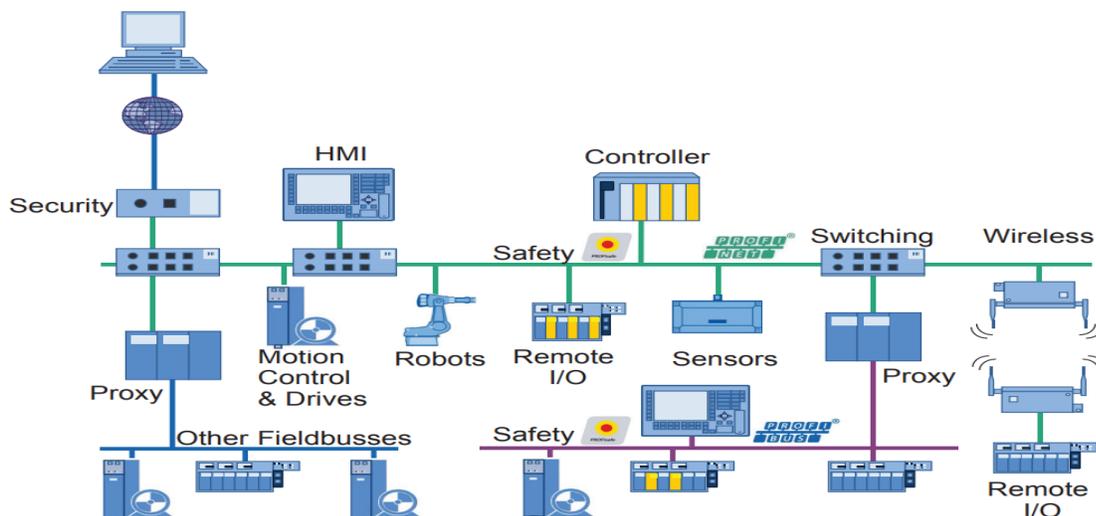


Figure 23 PROFINET satisfait à toutes les exigences de la technique d'automatisation.

III.3 Raisons d'utiliser PROFINET :

- ❖ Facilité d'utilisation
- ❖ Topologie de réseau flexible
- ❖ Diagnostic intégré
- ❖ Sécurité intégrée
- ❖ Haute disponibilité
- ❖ Temps réel évolutif
- ❖ Structures de système étendues
- ❖ Tout sur un seul câble
- ❖ Appui à l'optimisation énergétique
- ❖ Assistance mondiale [11]

III.4 Canaux de communication PROFINET :

Pour comprendre comment PROFINET déplace les informations, il est utile de garder à l'esprit le type d'informations qu'il déplace. Les informations doivent-elles être fournies immédiatement ? Est-ce critique pour la sécurité ? S'agit-il d'une grande quantité d'informations qui ne sont envoyées qu'une seule fois ? Ces différents types d'informations nécessitent des mécanismes de transmission différents, et ces mécanismes constituent des "canaux de communication". Il existe trois canaux de communication dans PROFINET : temps réel (RT), temps non réel (NRT) et temps réel isochrone (IRT).

Ces canaux de communication reposent sur de nombreux protocoles réseau pour échanger des données sur un réseau Ethernet. Au lieu de regrouper tous ces protocoles

dans un même panier, nous essaierons de les organiser dans un cadre pour en garder une trace. Ce cadre s'appelle le modèle OSI (Open System Interconnection). [12]

PROFINET utilise un panier plein de protocoles pour remplir le modèle OSI. Et chacun de ces protocoles contient des données qui doivent être transmises entre les partenaires de communication.

❖ Temps réel (RT) :

PROFINET utilise uniquement les couches OSI un et deux pour transmettre des données en temps réel. Sur le fil, cela signifie que seules les adresses MAC de destination et de source sont regroupées dans la trame, avec l'Ethertype.

- transfert haute performance.
- données cycliques.
- signaux évènements.
- utilise des contrôleurs Ethernet standard.

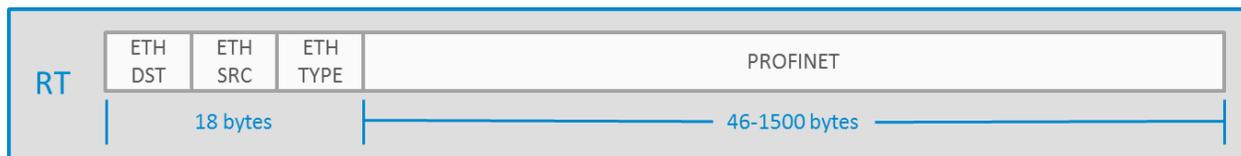


Figure 24 A layer 2 Ethernet frame.

❖ Temps non réel (NRT) :

PROFINET utilise le canal NRT pour des communications moins urgentes, comme l'établissement d'une connexion entre un appareil et un contrôleur ou l'accès aux données de diagnostic à partir d'un réseau externe. Mais cette flexibilité

supplémentaire s'accompagne d'une surcharge de données accrue - 108 octets supplémentaires sur chaque paquet.

- Paramétrage
- lecture des données de diagnostics
- chargement des interconnexions
- négociation des canaux de communication des données utilisateur.

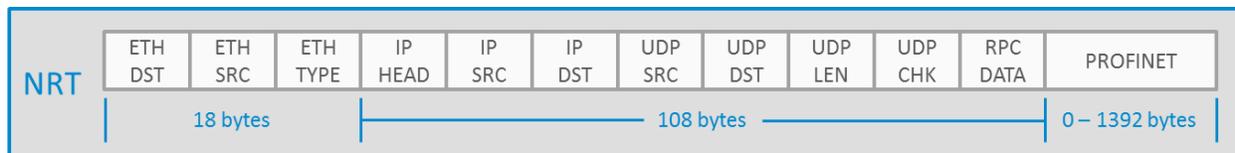


Figure 25 paquet non en temps réel avec des données de protocole supplémentaires.

❖ Temps réel isochrone (IRT) :

IRT élimine les retards de trafic en ajoutant aux règles utilisées pour commuter le trafic Ethernet et en créant des règles spéciales pour le trafic PROFINET. Il ajoute quelques extensions à l'Ethernet IEEE 802.3 standard pour implémenter quelque chose comme une «HOV Lane » pour le trafic IRT.

- transfert haute performance avec synchronisation d'horloge.
- jitter < 1u sec.
- Hardware spécifique nécessaire. [13]

III.5 Composants Profinet (stations):

Un réseau PROFINET peut être un ensemble complexe de stations, des périphériques d'E/S numériques aux actionneurs pneumatiques en passant par les scanners laser... La

liste semble interminable et s'allonge chaque jour. Mais dans ce réseau complexe, tous les composants PROFINET fonctionnent dans trois rôles différents. Il peut s'agir d'appareils, de contrôleurs ou de moniteurs. Ces rôles ne sont pas basés sur ce que font les composants dans le monde réel. Au lieu de cela, ils sont basés sur la façon dont chaque composant interagit avec tous les autres composants du réseau. Ce sont les fondements de la fondation PROFINET.

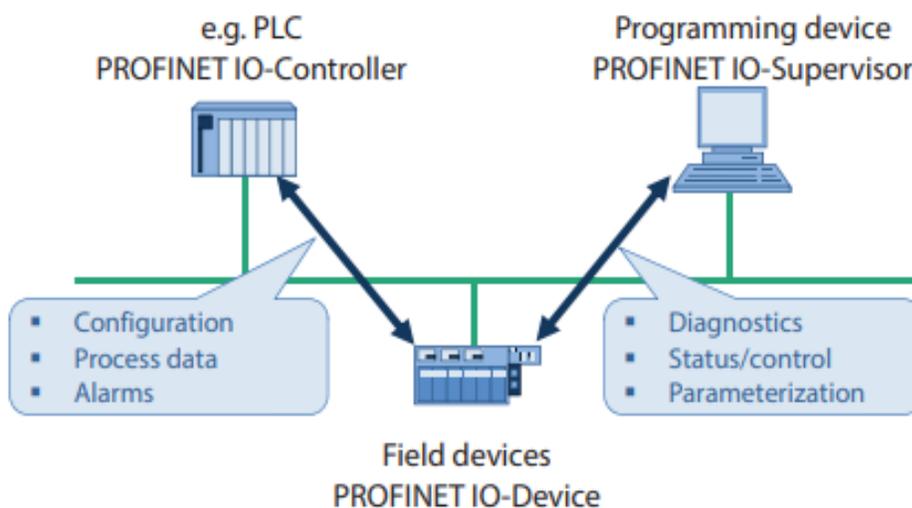


Figure 26 Voies de communication pour PROFINET IO.

III.5.1 I/O device :

Sont des unités autonomes conçues pour communiquer des informations en temps réel à un contrôleur. Ils n'essaient pas de communiquer directement avec d'autres appareils. Au lieu de cela, ils signalent leurs données en temps réel (cycliques) directement à un contrôleur et peuvent envoyer des données d'alarme ou de diagnostic (acycliques) à un superviseur. [12]

III.5.2 I/O contrôleurs :

Sont des agrégateurs de données en temps réel (cycliques) envoyées par un ou plusieurs appareils. Non seulement ils suivent les données en temps réel d'un appareil, mais ils collectent des informations sur l'état de maintenance de chaque appareil, collectent des messages d'alarme et mettent toutes ces informations à la disposition d'un utilisateur final. Les contrôleurs sont généralement des API, des applications logicielles pour PC ou des IHM haut de gamme. [12]

III.5.3 I/O Superviseur :

Cela peut être un dispositif de programmation (PD), ordinateur personnel (PC) ou machine humaine périphérique d'interface (IHM) pour la mise en service ou fins de diagnostic. [11]

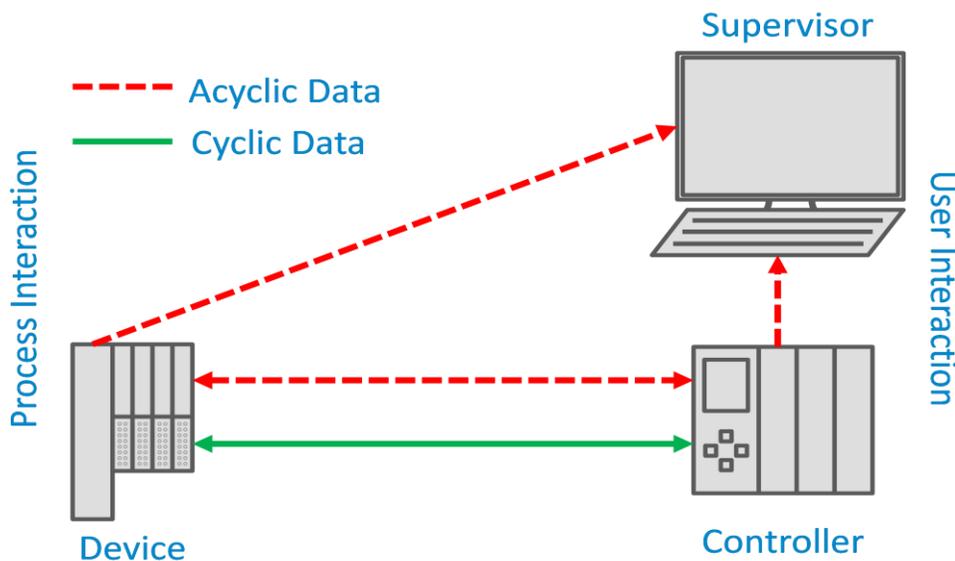


Figure 27 Les appareils et contrôleurs PROFINET échangent régulièrement des données cycliques, mais les superviseurs ne se connectent que lorsqu'ils ont besoin d'informations spécifiques.

III.6 Descriptions des appareils :

Pour permettre l'ingénierie du système, les fichiers GSD (Description générale de la station) des appareils de terrain à configurer sont nécessaires. Cette base XML GSDML décrit les propriétés et les fonctions des appareils de terrain PROFINET IO. Il contient toutes données pertinentes pour l'ingénierie ainsi que pour les données échange avec l'appareil de terrain. L'appareil de terrain le fabricant doit fournir le fichier GSD basé sur XML dans conformément à la spécification GSDML. [11]

III.7 Adressage :

Les périphériques Ethernet passent toujours leur Adresse MAC unique (voir encadré). Dans un système PROFINET IO, chaque appareil de terrain reçoit

Le nom symbolique qui identifie de manière unique le champ Appareils dans ce système IO. Utilisera ce nom Utilisé pour attribuer des adresses IP aux adresses MAC équipement de terrain. DCP (Découverte et Base protocole de configuration) est utilisé pour cela.

Ce nom est attribué aux appareils individuels et donc à l'adresse MAC du périphérique IO par un outil d'ingénierie utilisant le protocole DCP lors mise en service (initialisation de l'appareil). En option, le nom peut également être attribué automatiquement par le contrôleur IO au périphérique IO au moyen d'une topologie basée sur la détection de voisinage. L'adresse IP est attribuée en fonction du nom de l'appareil utilisant le protocole DCP. Parce que DHCP (Dynamique Host Configuration Protocol) est largement utilisé internationalement, PROFINET a prévu en option réglage de l'adresse via DHCP ou via des mécanismes spécifiques au fabricant. Les options d'adressage prises en charge par un appareil de terrain sont définies dans le fichier GSDML pour le champ respectif dispositif. [11]

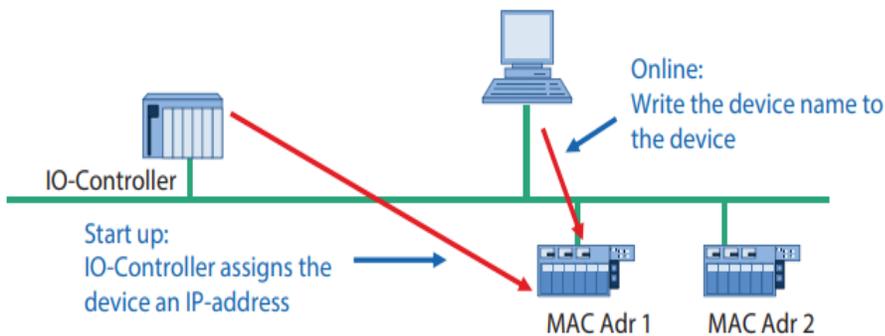


Figure 28 Attribution de nom.

III.8 Profinet DCP et DHCP :

On nous demande souvent : pourquoi PROFINET n'utilise-t-il pas le protocole DHCP (Dynamic Host Configuration Protocol) pour attribuer des adresses IP puisque DHCP est largement utilisé dans les bureaux ? Réponse : DHCP est facultatif, mais vous devez être conscient de certaines différences entre DHCP et PROFINET DCP. [12]

❖ Profinet DCP :

Profinet DCP fait partie de la suite de protocoles et signifie «Discovery and Basic Configuration Protocol». Il est utilisé par l'outil d'ingénierie et le contrôleur pour découvrir les périphériques, identifier les informations de l'appareil et configurer les paramètres de l'appareil tels que le nom de l'appareil Profit et l'adresse IP sur un réseau Profit. Profinet DCP est un protocole de couche de liaison Ethernet et offre plusieurs services. Il est normalement utilisé dans les systèmes Profinet pour gérer la gestion de l'adresse et du nom du réseau car chaque périphérique Profinet se voit attribuer un nom de périphérique unique basé sur les conventions de dénomination du système de noms de domaine (DNS) et une adresse IP (Ex: 192.168.1.2). [12]

Profinet DCP propose des services tels que :

DCP Identify All,

DCP Identify,

DCP Set,

DCP Set / Reset,

DCP Set / Signal (Flash),

DCP Get, DCP Hello. [12]

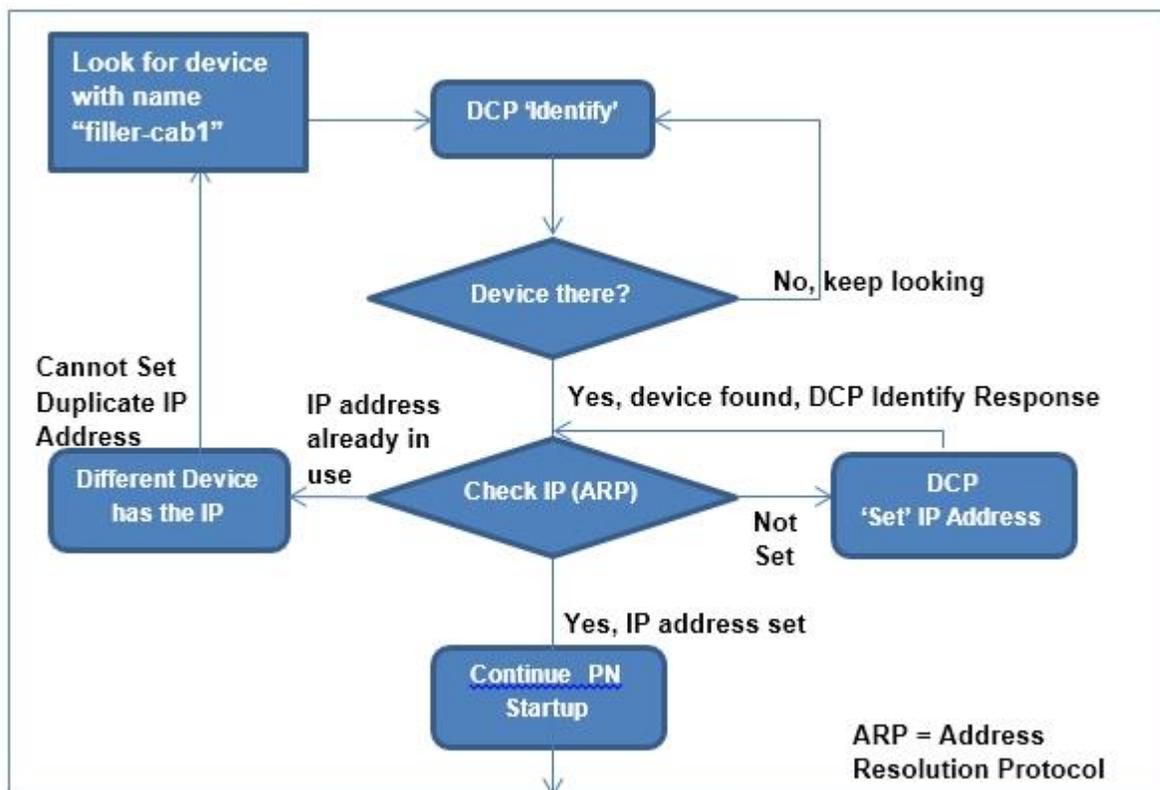


Figure 29 Extrait de la séquence de démarrage de Profinet à l'aide de DCP à partir du contrôleur I/O.

❖ **Profinet DHCP :**

DHCP fournit une méthode similaire pour gérer l'affectation d'adresse et peut être utilisée dans certaines installations spéciales. Il nécessite l'utilisation d'un serveur DHCP. Un hôte (client) fait une demande sur le démarrage et le serveur distribue une adresse IP «louée» en fonction des paramètres du serveur. C'est un problème, car DHCP s'appuie sur un serveur disponible et pris en charge par quelqu'un. [12]

III.9 LLDP et Profinet :

Vous pouvez configurer votre système PROFINET, y compris le contrôleur et les appareils. Lors du démarrage, les appareils commencent à échanger des informations LLDP et les informations de topologie peuvent être lues à partir des appareils par l'outil d'ingénierie PROFINET. Cette topologie peut maintenant être enregistrée en tant que "Baseline" et chargée dans le contrôleur PROFINET IO. Si des modifications de la topologie sont détectées, le contrôleur recevra une alarme de la station concernée (par exemple, les ports ont été permutés) ou le voisin de la station (l'appareil en aval est déconnecté du réseau, rupture de câble). En utilisant les mécanismes de PROFINET. [12]

III.10 Fonctions :

III.10.1 Données cycliques échangé :

Les données d'E/S cycliques sont transmises via "IO Data CR" non reconnu comme données en temps réel entre fournisseur et consommateur dans une base de temps assignable. Le temps de cycle peut être défini individuellement pour connexions aux

différents appareils et sont ainsi adapté aux exigences de l'application.

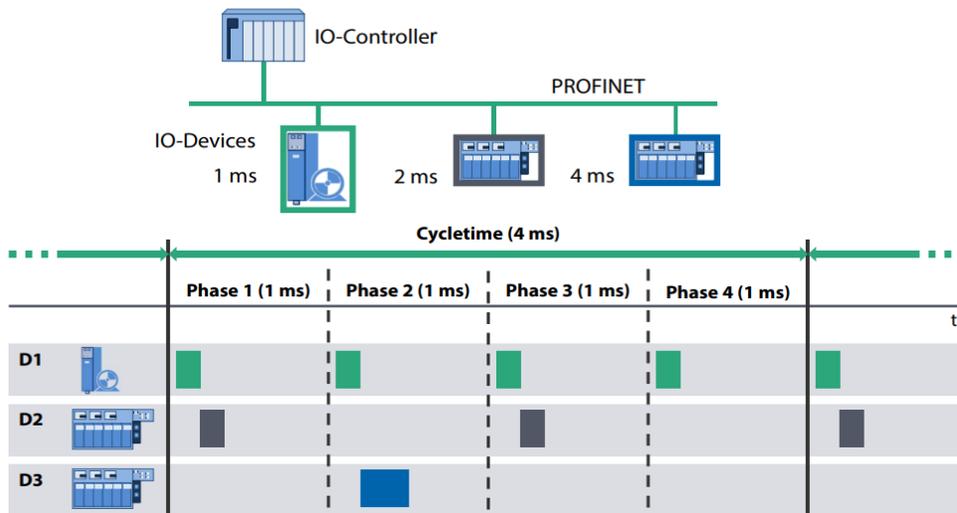


Figure 10: Real-time communication with cycle time monitoring

Figure 30 communication en temps réel avec surveillance du temps de cycle.

III.10.2 Échange de données acyclique :

Echange de données acyclique à l'aide de "Record Data CR" peut être utilisé pour le paramétrage ou configuration des périphériques IO ou lecture de l'état informations. Ceci est accompli avec le lire / écrire des trames à l'aide de services informatiques standard via TCP/IP dans lequel les différents enregistrements de données sont distingués par leur indice. En plus des enregistrements de données qui sont librement définissables par les fabricants d'appareils.

III.10.3 Diagnostic appareil/réseau :

Une approche de maintenance basée sur le statut est actuellement gagner en pertinence pour l'exploitation et la maintenance. Il est basé sur la capacité des appareils et composants pour déterminer leurs états et les communiquer via des mécanismes convenus. UN système de signalisation fiable des alarmes et de l'état les

messages des périphériques IO au contrôleur IO étaient définis pour PROFINET IO à cet effet. [11]

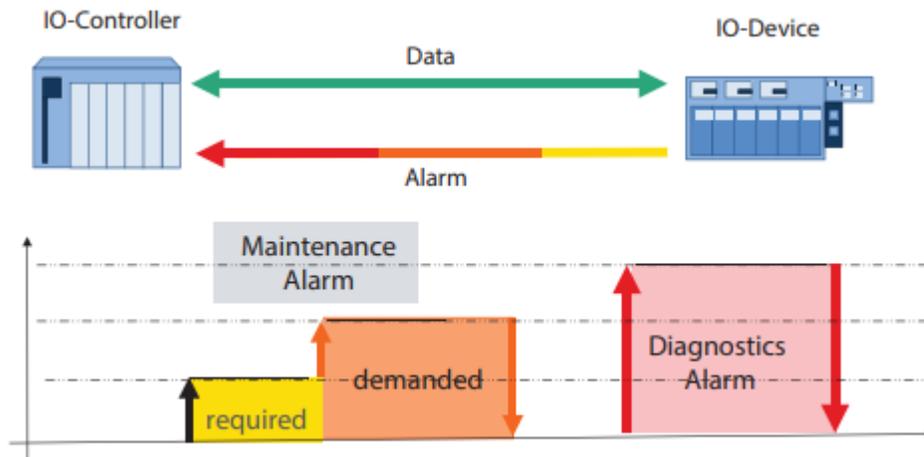


Figure 31 Modèle de diagnostic pour signaler les défauts avec différentes priorités.

III.11 Fonctions optionnelles :

- ❖ **Accès multiple aux entrées par divers contrôleurs :** Entrée partagée.

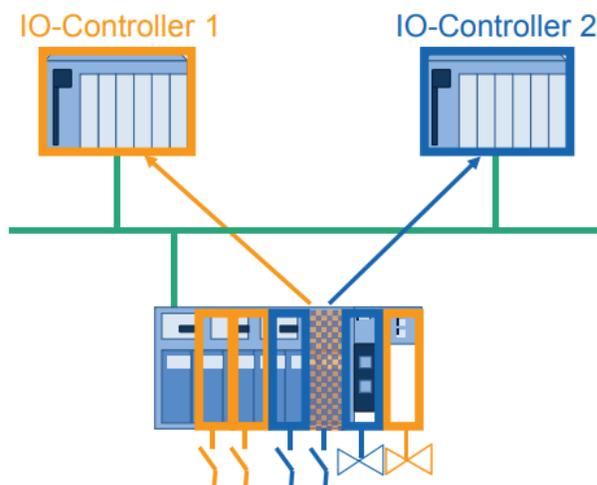


Figure 32 Plusieurs contrôleurs lus les mêmes entrées sur un appareil.

- ❖ **Répartition des fonctions de l'appareil à divers contrôleurs :** Appareil partagé.

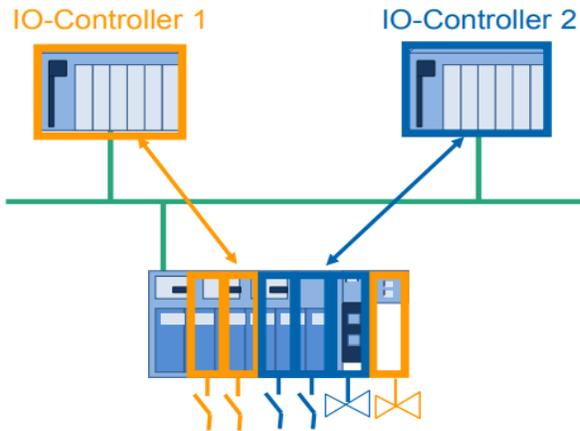


Figure 33 Accès par plusieurs contrôleurs à différents modules dans un dispositif.

Changements de configuration pendant opération : Configuration in Run (CiR).

Redémarrage rapide après rétablissement de la tension : Fast Start up (FSU).

Paramètre automatique affectation d'appareils à l'aide des ensembles de paramètres : Serveur de paramètres individuels. [11]

III.12 Facilité du câblage :

❖ **Sécurité intégrée :**

PROFINET du point de vue global de la sécurité pour les personnes, les machines et l'environnement Possède également toutes les qualités requises. C'est possible avec PROFIsafe Réseau pour une communication standard et sécurisée.

❖ **Standards TIC & accès sécurisé.**

❖ **Processus.**

❖ **Appareils de terrain décentralisés :**

PROFINET permet la connexion directe d'appareils de terrain à Industriel Ethernet Décentralisé, établissant ainsi un échange de données rapide entre les périphériques et

les contrôleurs, Cela s'accompagne d'une augmentation significative de la probabilité d'un diagnostic.

❖ **Motion Control :**

Basé sur Profinet, un réglage isochrone très rapide des applications de contrôle de mouvement peut également être facilement mis en œuvre via ISO-chrono en temps réel (IRT).

❖ **haute performance.**

❖ **Intelligence répartie.** [13]

III.13 Conclusion :

Dans ce chapitre, nous avons vu le protocole profinet et ses caractéristiques, avantages et sa flexibilité avec toutes les exigences industrielles, fournissant des solutions puissantes et intelligentes avec un travail considéré comme peu coûteux et efficace.

IV. Logiciel et le travail pratique :

IV.1 Introduction :

Les langages des A.P.I. sont des langages intermédiaires entre le langage évolué et le langage machine. Ils ont l'avantage d'avoir un jeu d'instructions incluant uniquement les fonctions logiques, cela a comme conséquences, une meilleure compréhension par les automaticiens et une simplification du compilateur de la console de programmation et du logiciel constructeur. [14]

Dans ce chapitre, nous utiliserons le tia portail v13 en essayant de programmer et de communiquer les deux automates avec profinet.

IV.2 Définition de tia portail :

TIA Portal est le logiciel d'ingénierie de Siemens qui permet de programmer des automates de la gamme Siemens. La nouvelle version de Step7 est fournie dans le logiciel d'ingénierie de Siemens TIA Portal (totally integrated Automation). TIA Portal Est un logiciel tout en un qui permet de programmer des automates, panels et contrôleurs d'axes Siemens. [15]

IV.3 Les langages des API :

Malheureusement il n'y a pas eu d'unicité chez les constructeurs quant au langage de programmation. Néanmoins 4 langages sont parmi les plus:

- Le langage LADDER (LD : Ladder diagram)
- Le langage booléen (FBD : Function Bloc Diagram)
- Le langage GRAFCET (SFC : Sequential Function Chart)
- Le langage mnémonique (IL : Instruction list) [14]

IV.4 La vue du logiciel tia portal :

La vue du portail offre un aperçu de toutes les étapes de configuration du projet et un accès orienté tâche de votre tâche d'automatisation.

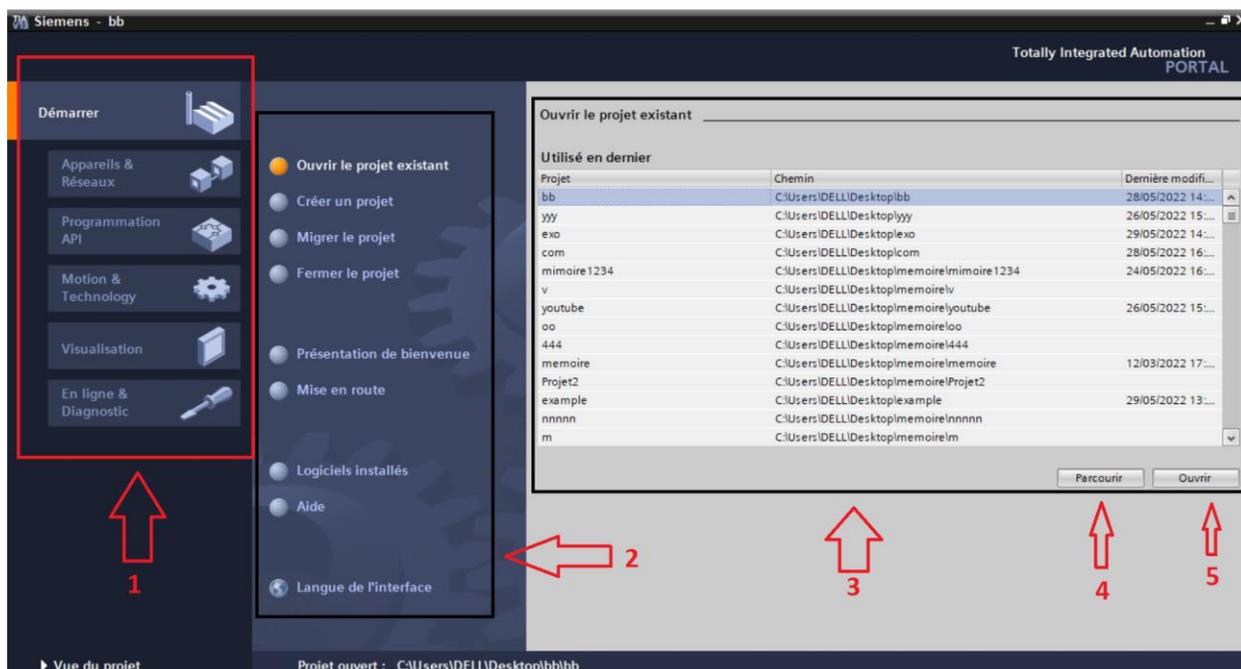


Figure 34 vue de logiciel.

1. portails des différentes taches.
2. Taches pour le portail sélectionné.
3. Fenêtre de sélection correspondant à tache sélectionnée.
4. Recherche du projet à ouvrir.
5. Action à ouvrir le projet sélectionné.

IV.5 La vue du projet :

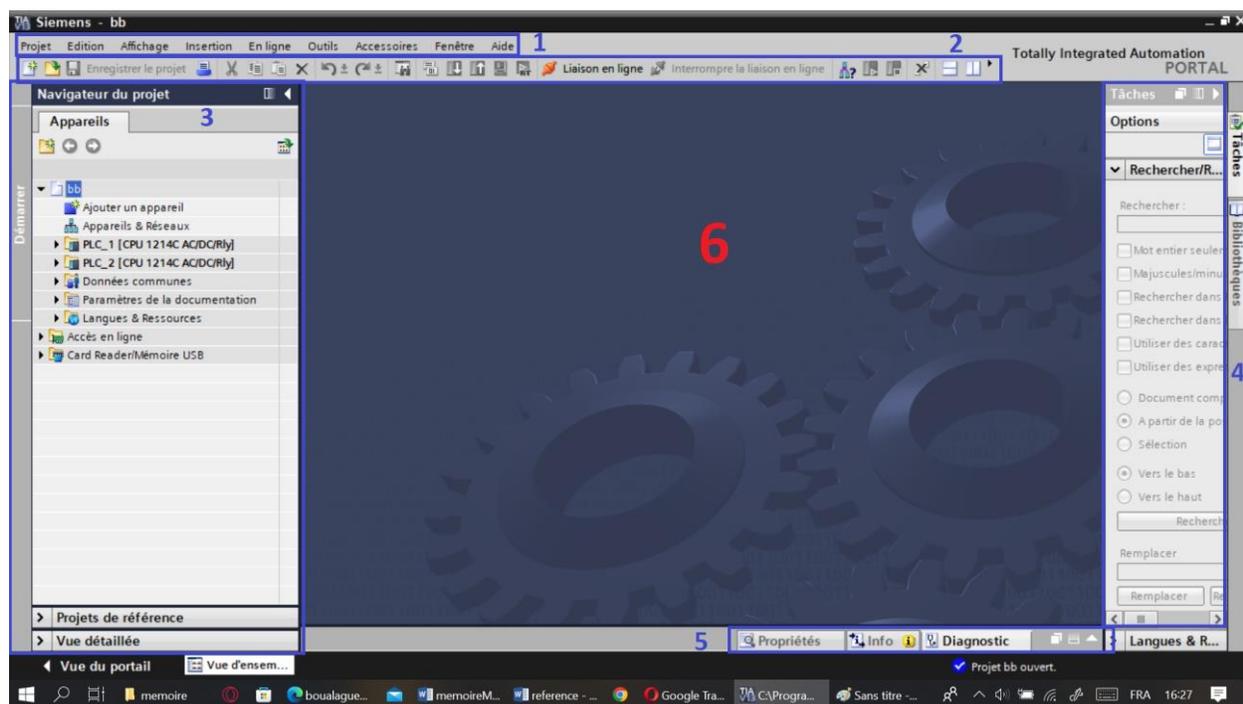


Figure 35 vue de projet.

1. Barre de menu.
2. Barre d'outils.
3. Navigation de projet.
4. Onglets des tâches.
5. Fenêtre d'inspection.
6. Zone de travail.

IV.6 Création et programmation d'un projet :

IV.6.1 Création :

Lorsqu' on appuyé sur le TIA portal la vue du portail s'affiche, l'une des méthodes de création du projet est la suivante: appuyer sur « créer un nouveau projet » puis appuyer sur « créer » puis cliquer simplement sur « vue du projet » en bas à gauche de la vue de portail, un nouveau projet est maintenant créé.

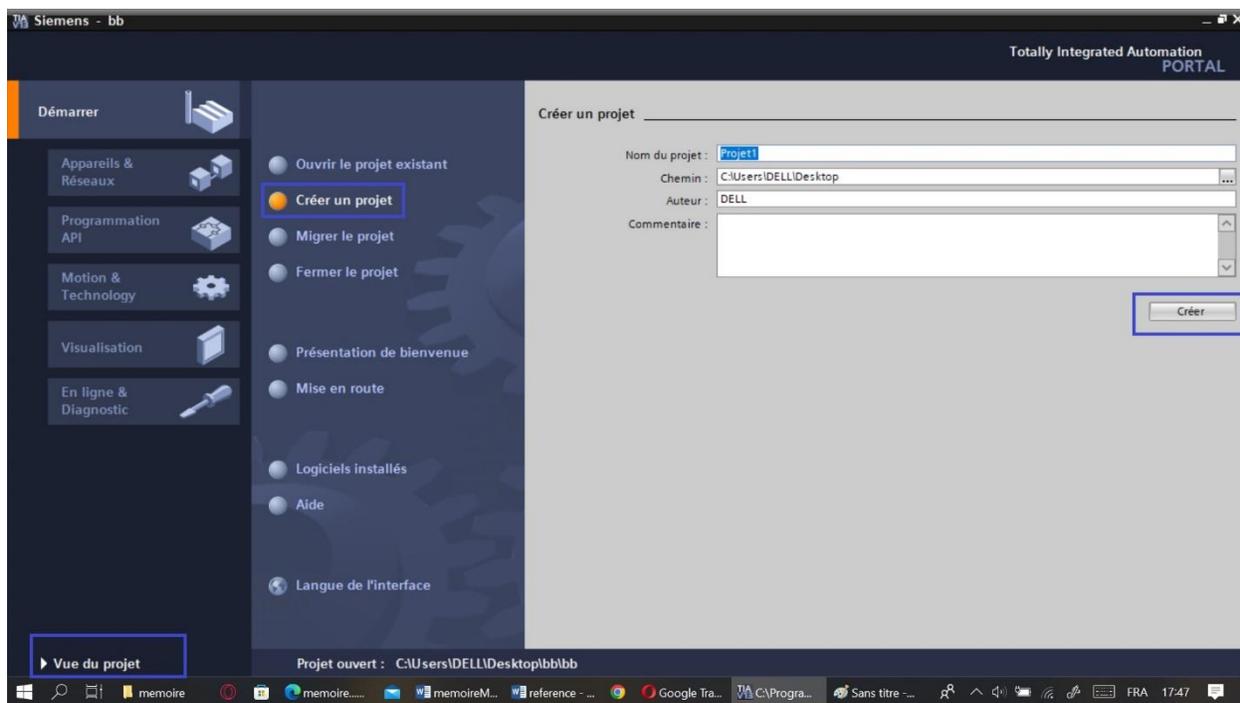


Figure 36 création de projet étapes.

IV.6.2 Configuration de matériaux :

En va voir la vue de projet, donc en allant dans le navigateur du projet et cliquer sur « ajouter un nouvel appareil » une nouvelle vue apparaîtra comme dans la, on choisit le type d'appareil Avec la caractéristique dont on a besoin pour travailler.

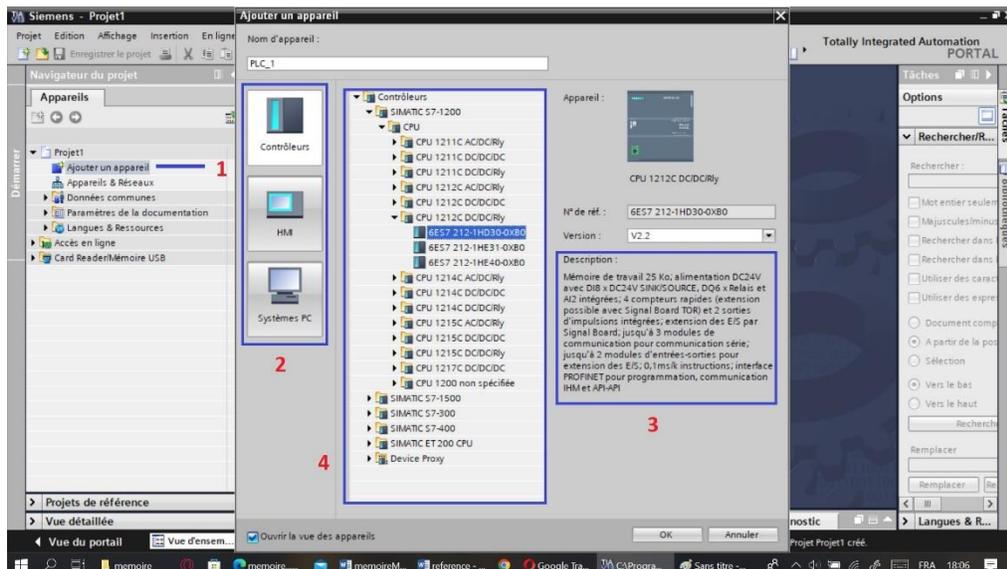


Figure 37 configuration de matériel.

1. Click pour ajouter un appareil.
2. Type d'appareil à configurer.
3. Description d'appareil.
4. Les modèles ou version des appareils.

IV.6.3 Adressage des E/S :

Après avoir créé le nouveau projet et choisi l'appareil, on doit premièrement vérifier les adresses des appareils qu'on trouve en cliquant sur l'appareil puis « Propriétés » puis appuyer sur les adresses numériques ou analogiques et voir s'il y a assez d'entrées et de sorties pour qu'on puisse travailler avec.

Si j'ai besoin de plus d'entrées ou sorties, nous pouvons simplement ajouter un nouveau module autant que nous en avons besoin

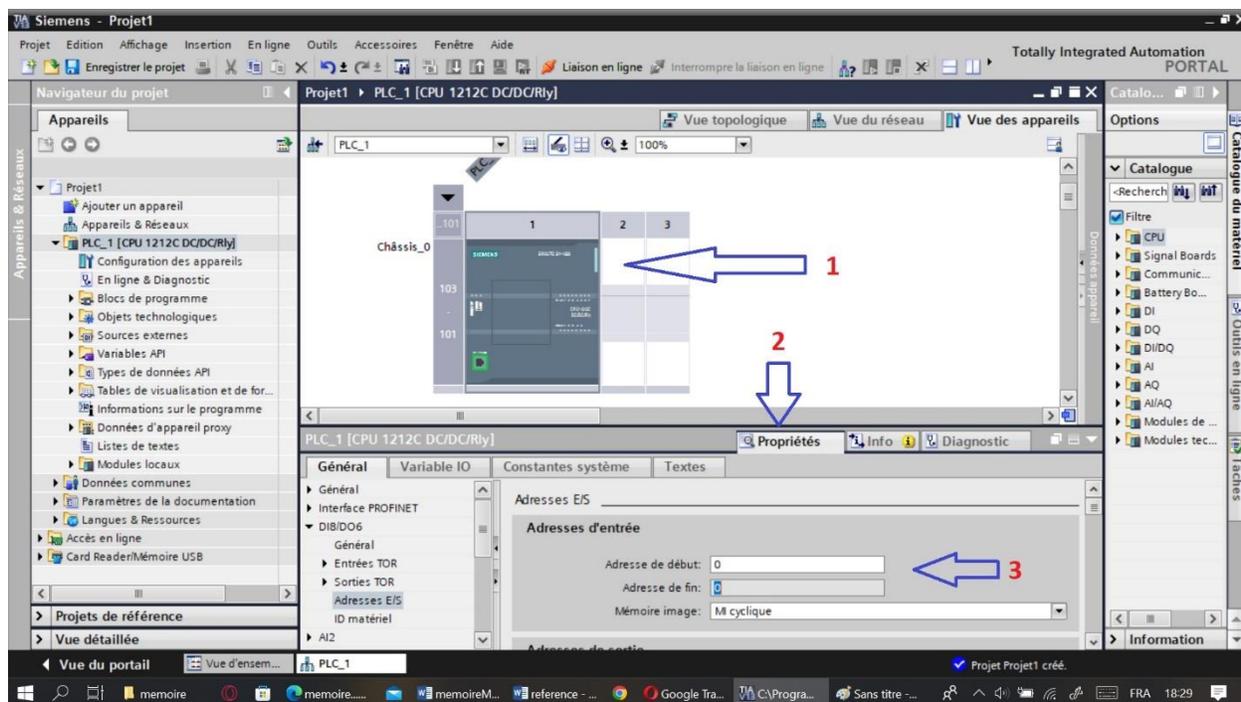


Figure 38 : adressage E/S.

1. Appareil.
2. Propriétés d'appareil.
3. Zone de modification.

IV.6.4 Programmation de projet :

Une fois que nous avons terminé notre configuration des matériaux, nous allons de l'avant pour commencer la programmation.

Nous ajoutons un nouveau bloc d'organisation en vous dirigeant vers blocs de programme, puis cliquez sur ajouter un nouveau bloc, puis OK.

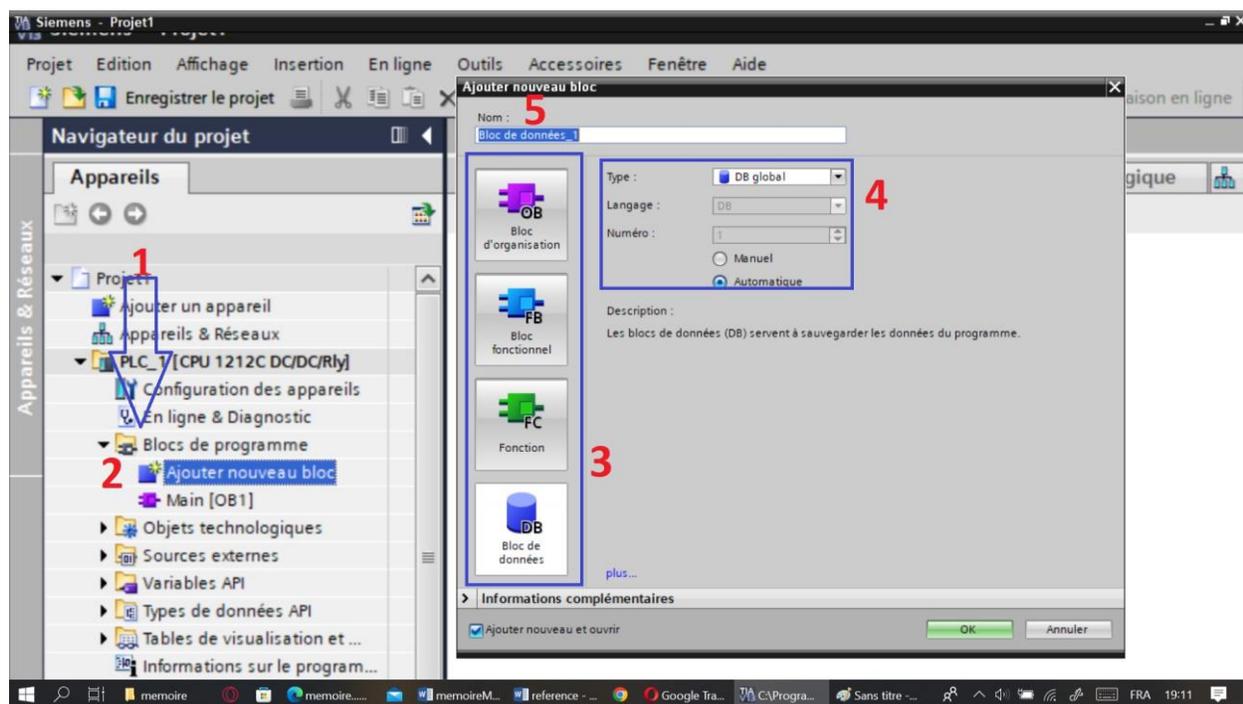


Figure 39 création d'un nouveau bloc de programmation.

1. Onglet pour les blocs.
2. Où est-on click pour ajouter un nouveau bloc.
3. Type de bloc qui on va ajout.
4. Définir le type et le langage et le numéro de bloc.
5. Définir un nom a le bloc.

❖ Les blocs de programmation :

- Les blocs de code (OB, FB, FC) qui contiennent les qu'on doit charger dans la CPU.
- Les blocs de données (DB) qui contiennent les données du programme.

❖ **Bloc OB** : Les OB sont appelés par le système d'exploitation, et ils y a des plusieurs types :

- Type qui fait le traitement de programmes pour les programmes cyclique.
- Type qui déclenchés par un événement.
- Type qui gère le comportement à la mise en route de l'automate programmable et en fin.

❖ **Bloc FB** : Le FB est sous-programme écrit par l'utilisateur et exécuté par des blocs de code on lui associe un bloc de données d'instance relatif à sa mémoire et contenant ses paramètres. [15]

❖ **Bloc FC** : Un FC est utilisé pour définir une fonction qui peut être exécutée sur commande d'un bloc d'organisation, les FC peuvent être considérés comme des sous-programmes que tu peux appeler à partir d'autres blocs.

❖ **Bloc DB** : Une DB est une liste de données dans la mémoire de l'automate. Il s'agit généralement d'un enregistrement unique ou d'une combinaison d'enregistrements. Il peut être utilisé par votre programme pour sauvegarder des données dans l'unité centrale. Il y a deux types de blocs de données. Les DB globaux et les DB d'instance locale.

IV.6.5 Structure du programme :

L'écriture du programme utilisateur complet peut se faire à partir du bloc d'organisation OB1, ce qui n'est pas recommandé pour les programmes de grande taille voire de taille moyenne. Pour les automatisations complexes, il est recommandé

IV. Chapitre : Logiciel et le travail pratique.

de subdiviser en parties plus petites, qui correspondent aux fonctions techniques du procédé, appelées blocs (programmation structurée). Cette structure présente les avantages suivants :

- Standardiser certaines parties du programme.
- Simplifier l'organisation du programme.
- Modifier facilement le programme.
- Simplifier le test du programme en l'exécutant section par section.
- Faciliter la mise en service [16].

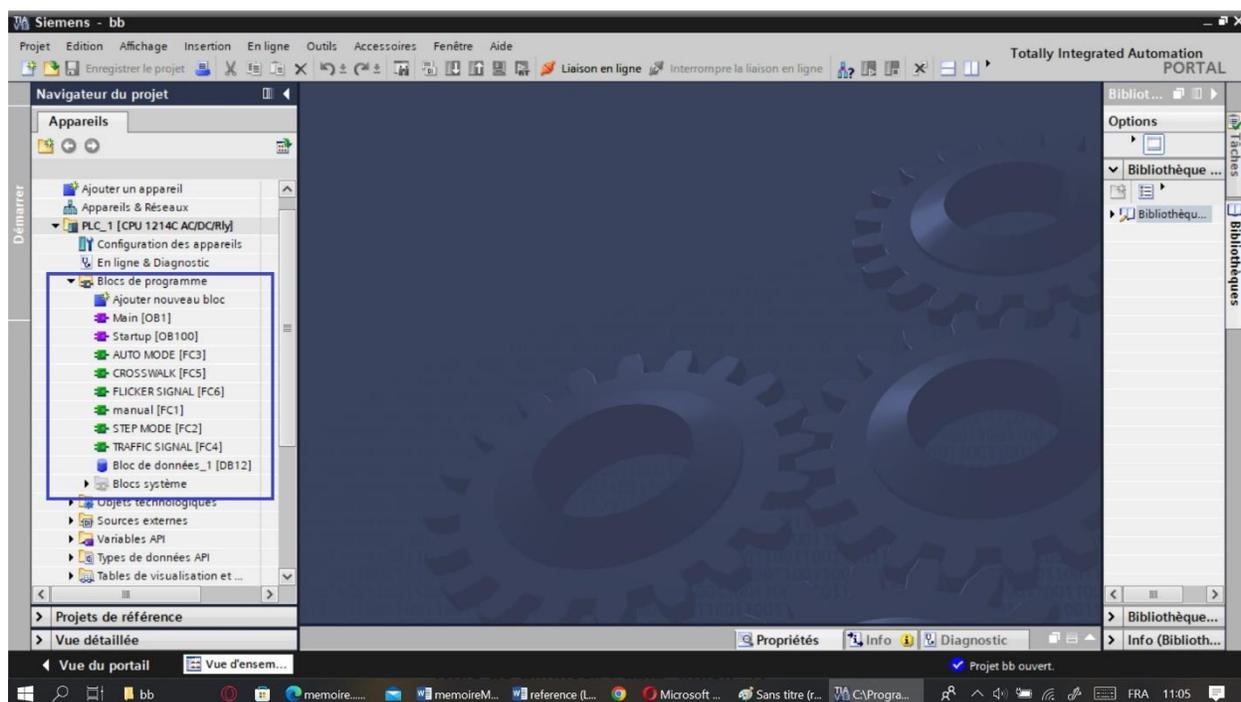


Figure 40 exemple de structure de programme.

IV.6.6 Exécution du programme :

Nous commençons la programmation et une fois cela fait, nous devons vérifier le programme en utilisant le bouton "Compiler" pour compiler, puis utiliser le bouton "Charger" pour le charger dans la CPU.

IV. Chapitre : Logiciel et le travail pratique.

En cliquant sur le bouton « en ligne » puis sur l'icône de « la visualisation » et on peut voir les changements dans le programme lorsqu'il fonctionnera.

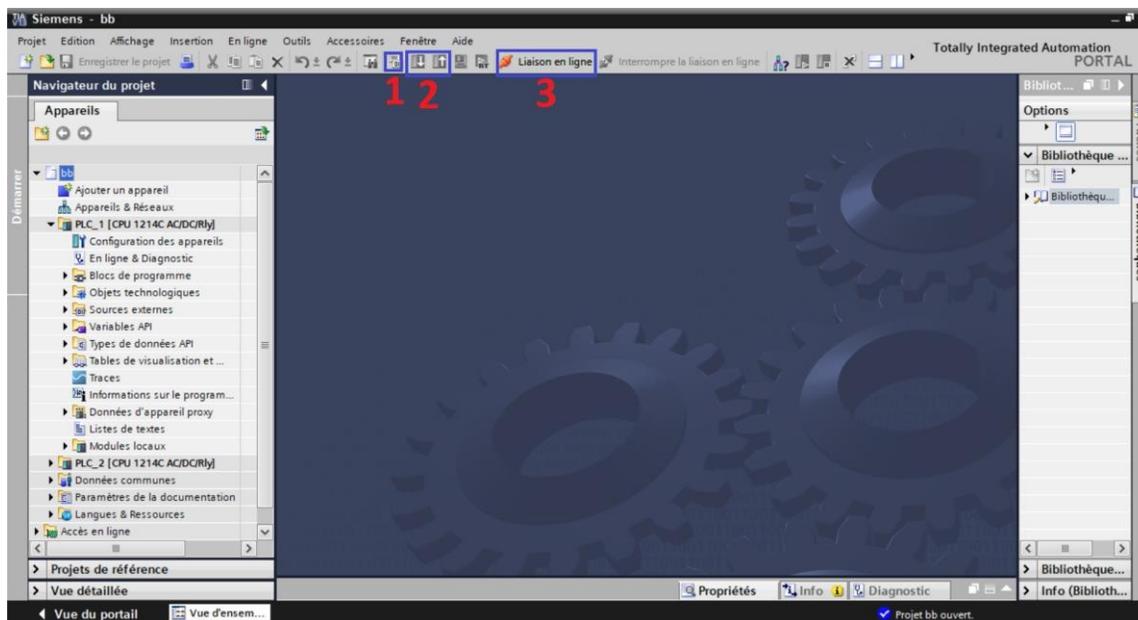


Figure 41 TIA portal en vue.

- 1) Compiler.
- 2) Chargement de programme.
- 3) Faire la liaison en ligne.

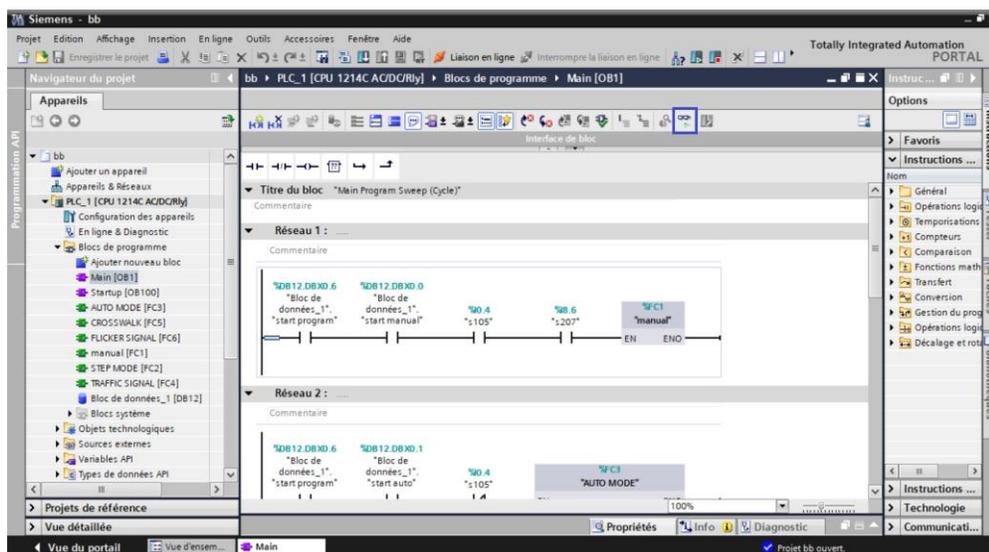


Figure 42 le bouton de visualisation.

IV.6.7 Simulation du programme :

On utilise un logiciel optionnel STEP7 le PLCSIM.

Appuyez sur le bouton "Simulation" pour démarrer la simulation, il ouvrira automatiquement PLCSIM, Après avoir "chargé" le programme dans CPU de l'automate, sélectionnez simplement un mode pour CPU.

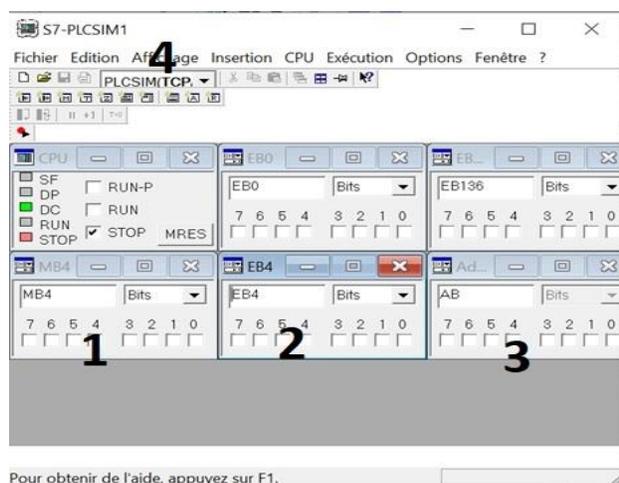


Figure 43 simulateur de step7.

1. Entrée mémoire.
2. Entrée physique.
3. Sortie.
4. Mode de communication.

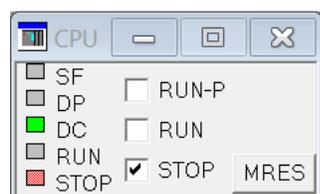


Figure 44 mode d'exécution.

IV.7 Programmation et communication avec deux méthodes :

IV.7.1 Feux de circulation :

Système Aperçu : Cet entraîneur de contrôle des feux de circulation est capable de pratiquer l'expérimentation pour contrôler le système de feux de circulation commun à l'intersection. L'entraîneur de contrôle des feux de circulation peut être connecté à un contrôleur devenu un interrupteur de diverses opérations et avoir des signaux et le piéton. Traverser un carrefour utilise une LED toutes les caractéristiques des modes de réalisation à est devenu le même que le réel, Il a été conçu pour être en mesure d'améliorer la compréhension de l'enseignement. En particulier, donc la programmation est prédite sur la base de tous les feux de circulation La séquence de règles pour la formation est grande. Il est possible d'effectuer l'exercice avec intérêt. En outre, on peut facilement découvrir qu'une partie des erreurs dysfonctionnelles du programme est erronée même se produisant lors de données également fixes, l'autre séquence est capable de modifications sans affecter.

1) La programmation :

a) Configuration matériels :

i) Automates :

- CPU : 1214C AC/DC/RLY.
- N de réf : 6ES7 214-1BG40-0XB0
- CPU : 1214C AC/DC/RLY.
- N de réf : 6ES7 214-1BG40-0XB0.

ii) Les modules E/S :

- Module Entrée : DI 16*24VDC.
- N de réf : 6ES7 221-1BH30-0XB0.
- Module sortie : DO 16*24VDC.

IV. Chapitre : Logiciel et le travail pratique.

- N de réf : 6ES7 222-1BH30-0XB0.

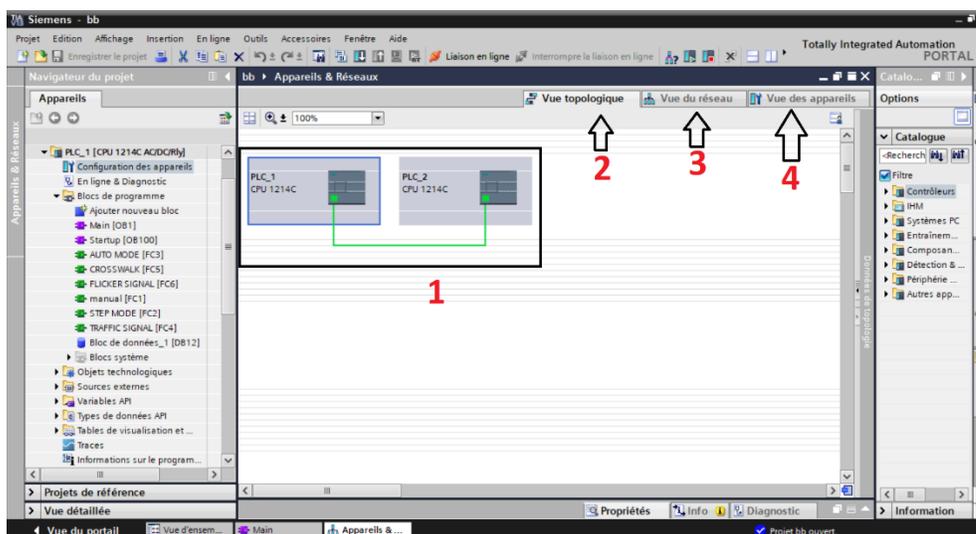


Figure 45 vue de zone de travail.

- 1 Deux automates connectent.
 - 2 Vue topologie de matériels.
 - 3 Vue de réseau (le protocole et ses adresses).
 - 4 Vue d'appareil spécialisée.
- b) **Adresse Ethernet de la CPU :** Dans les propriétés de la CPU, son adresse Ethernet peut également être définie. Une double-cliqué sur l'icône Ethernet d'un poste de travail fait apparaître une fenêtre d'inspection qui vous permet de définir ses propriétés.
- Pour établir une connexion entre la CPU et la console de programmation, les deux appareils doivent se voir attribuer des adresses appartenant au même sous-réseau.

IV. Chapitre : Logiciel et le travail pratique.

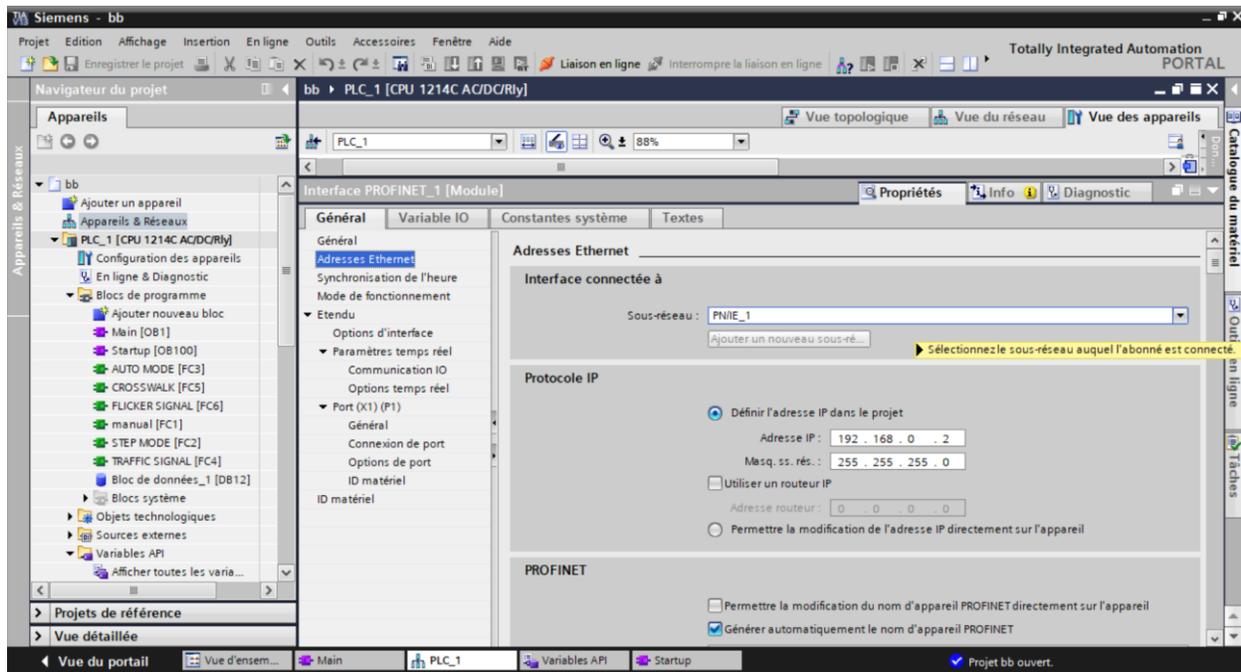


Figure 46 adressage d'Ethernet de CPU_1.

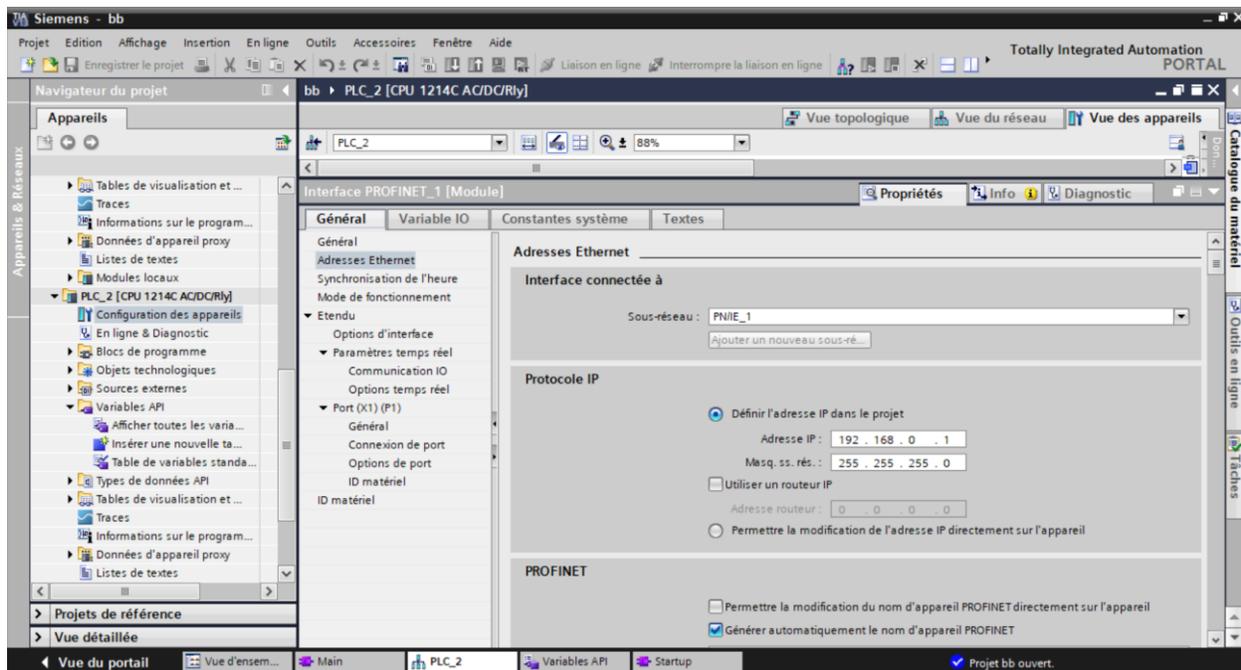


Figure 47 adressage d'Ethernet de CPU_2.

c) Bloc de programmation :

i) Tableau de variables :

On déclare les variables nécessaires sur la table des variables dans notre système.

| | Nom | Table des variables | Type de données | Adresse | Réma... | Visibl... | Acces... | Commentaire |
|----|------|------------------------|-----------------|---------|---------|-------------------------------------|-------------------------------------|-------------|
| 1 | s101 | Table de variabl... | Bool | %I0.0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 2 | s102 | Table de variables s.. | Bool | %I0.1 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 3 | s103 | Table de variables s.. | Bool | %I0.2 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 4 | s104 | Table de variables s.. | Bool | %I0.3 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 5 | s105 | Table de variables s.. | Bool | %I0.4 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 6 | s106 | Table de variables s.. | Bool | %I0.5 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 7 | s107 | Table de variables s.. | Bool | %I0.6 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 8 | s108 | Table de variables s.. | Bool | %I0.7 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 9 | s109 | Table de variables s.. | Bool | %I1.0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 10 | s110 | Table de variables s.. | Bool | %I1.1 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 11 | s111 | Table de variables s.. | Bool | %I1.2 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 12 | s112 | Table de variables s.. | Bool | %I1.3 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 13 | s113 | Table de variables s.. | Bool | %I1.4 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 14 | s114 | Table de variables s.. | Bool | %I1.5 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 15 | s201 | Table de variables s.. | Bool | %I8.0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 16 | s202 | Table de variables s.. | Bool | %I8.1 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 17 | s203 | Table de variables s.. | Bool | %I8.2 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 18 | s204 | Table de variables s.. | Bool | %I8.3 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 19 | s205 | Table de variables s.. | Bool | %I8.4 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 20 | s206 | Table de variables s.. | Bool | %I8.5 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 21 | s207 | Table de variables s.. | Bool | %I8.6 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 22 | L10 | Table de variables s.. | Bool | %Q0.0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |

IV. Chapitre : Logiciel et le travail pratique.

The figure consists of two screenshots of a software interface, likely Siemens SIMATIC Manager, showing variable declarations for a PLC. The interface title is 'bb ▶ PLC_1 [CPU 1214C AC/DC/Rly] ▶ Variables API'. The top screenshot shows a list of variables from L35 to Tag_21, and the bottom screenshot shows a list from L11 to L34. Both tables have columns for 'Nom', 'Table des variables', 'Type de données', 'Adresse', 'Réma...', 'Visibl...', 'Acces...', and 'Commentaire'.

| | Nom | Table des variables | Type de données | Adresse | Réma... | Visibl... | Acces... | Commentaire |
|----|--------|------------------------|-----------------|---------|--------------------------|-------------------------------------|-------------------------------------|-------------|
| 45 | L35 | Table de variables s.. | Bool | %Q9.7 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 46 | Tag_1 | Table de variables s.. | Bool | %M0.1 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 47 | Tag_2 | Table de variables s.. | DWord | %MD10 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 48 | Tag_3 | Table de variables s.. | Bool | %M0.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 49 | Tag_4 | Table de variables s.. | Bool | %M0.2 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 50 | Tag_5 | Table de variables s.. | Bool | %M0.3 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 51 | Tag_6 | Table de variables s.. | Bool | %M0.4 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 52 | Tag_7 | Table de variables s.. | Bool | %M1.3 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 53 | Tag_8 | Table de variables s.. | Bool | %M1.4 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 54 | Tag_9 | Table de variables s.. | Bool | %M1.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 55 | Tag_10 | Table de variables s.. | Bool | %M1.1 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 56 | Tag_11 | Table de variables s.. | Bool | %M1.2 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 57 | Tag_12 | Table de variables s.. | Bool | %M4.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 58 | Tag_13 | Table de variables s.. | Bool | %M1.5 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 59 | Tag_14 | Table de variables s.. | Bool | %M1.7 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 60 | Tag_15 | Table de variables s.. | Bool | %M1.6 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 61 | Tag_16 | Table de variables s.. | Bool | %M2.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 62 | Tag_17 | Table de variables s.. | Bool | %M2.1 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 63 | Tag_18 | Table de variables s.. | Bool | %M2.2 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 64 | Tag_19 | Table de variables s.. | Bool | %M3.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 65 | Tag_20 | Table de variables s.. | Bool | %M3.1 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 66 | Tag_21 | Table de variables s.. | Bool | %M3.2 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |

| | Nom | Table des variables | Type de données | Adresse | Réma... | Visibl... | Acces... | Commentaire |
|----|-----|------------------------|-----------------|---------|--------------------------|-------------------------------------|-------------------------------------|-------------|
| 23 | L11 | Table de variables s.. | Bool | %Q0.1 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 24 | L12 | Table de variables s.. | Bool | %Q0.2 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 25 | L13 | Table de variables s.. | Bool | %Q0.3 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 26 | L14 | Table de variables s.. | Bool | %Q0.4 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 27 | L15 | Table de variables s.. | Bool | %Q0.5 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 28 | L16 | Table de variables s.. | Bool | %Q0.6 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 29 | L17 | Table de variables s.. | Bool | %Q0.7 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 30 | L20 | Table de variables s.. | Bool | %Q8.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 31 | L21 | Table de variables s.. | Bool | %Q8.1 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 32 | L22 | Table de variables s.. | Bool | %Q8.2 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 33 | L23 | Table de variables s.. | Bool | %Q8.3 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 34 | L24 | Table de variables s.. | Bool | %Q8.4 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 35 | L25 | Table de variables s.. | Bool | %Q8.5 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 36 | L26 | Table de variables s.. | Bool | %Q8.6 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 37 | L27 | Table de variables s.. | Bool | %Q8.7 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 38 | L28 | Table de variables s.. | Bool | %Q9.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 39 | L29 | Table de variables s.. | Bool | %Q9.1 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 40 | L30 | Table de variables s.. | Bool | %Q9.2 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 41 | L31 | Table de variables s.. | Bool | %Q9.3 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 42 | L32 | Table de variables s.. | Bool | %Q9.4 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 43 | L33 | Table de variables s.. | Bool | %Q9.5 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 44 | L34 | Table de variables s.. | Bool | %Q9.6 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |

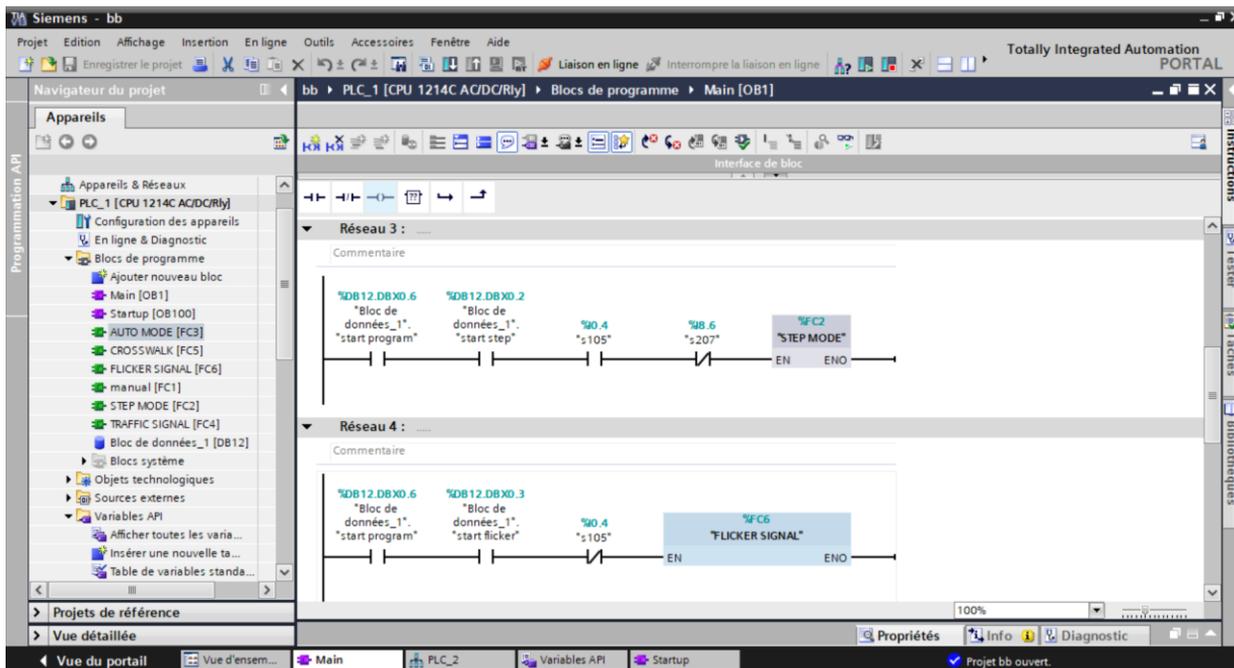
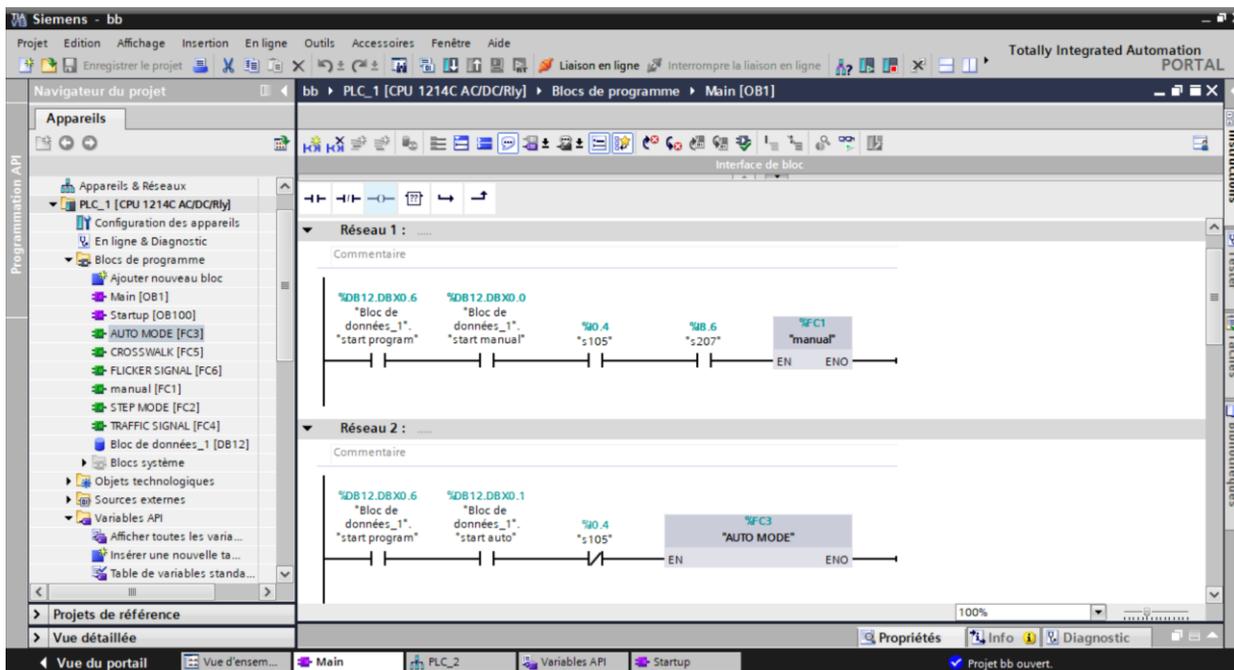
Figure 48 tableaux de variables.

ii) **Bloc d'organisation OB100** : Un OB100 est généralement utilisé pour initialiser des variables ou modules d'E/S au démarrage du programme.

IV. Chapitre : Logiciel et le travail pratique.

iii) **Le programme dans OB** : dans notre programme on utilisant le langage LADDER, Le TIA portal fournit des Cartes de tâches qui contiennent les instructions pour le programme.

- CPU_1 :



IV. Chapitre : Logiciel et le travail pratique.

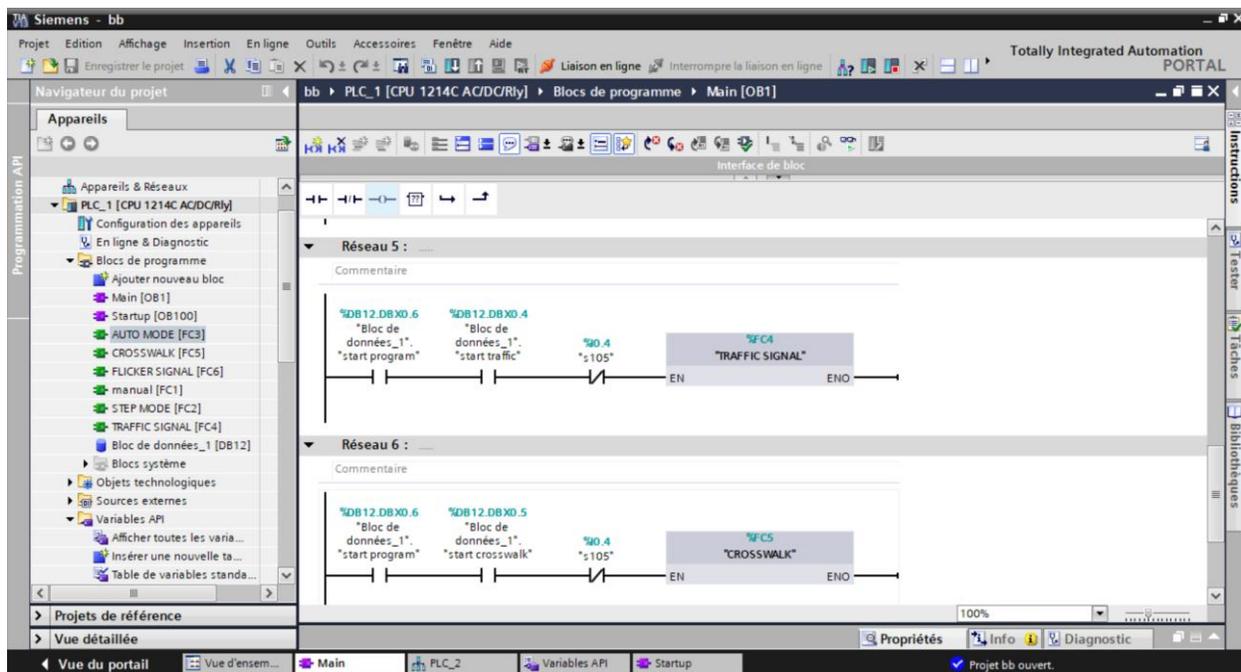
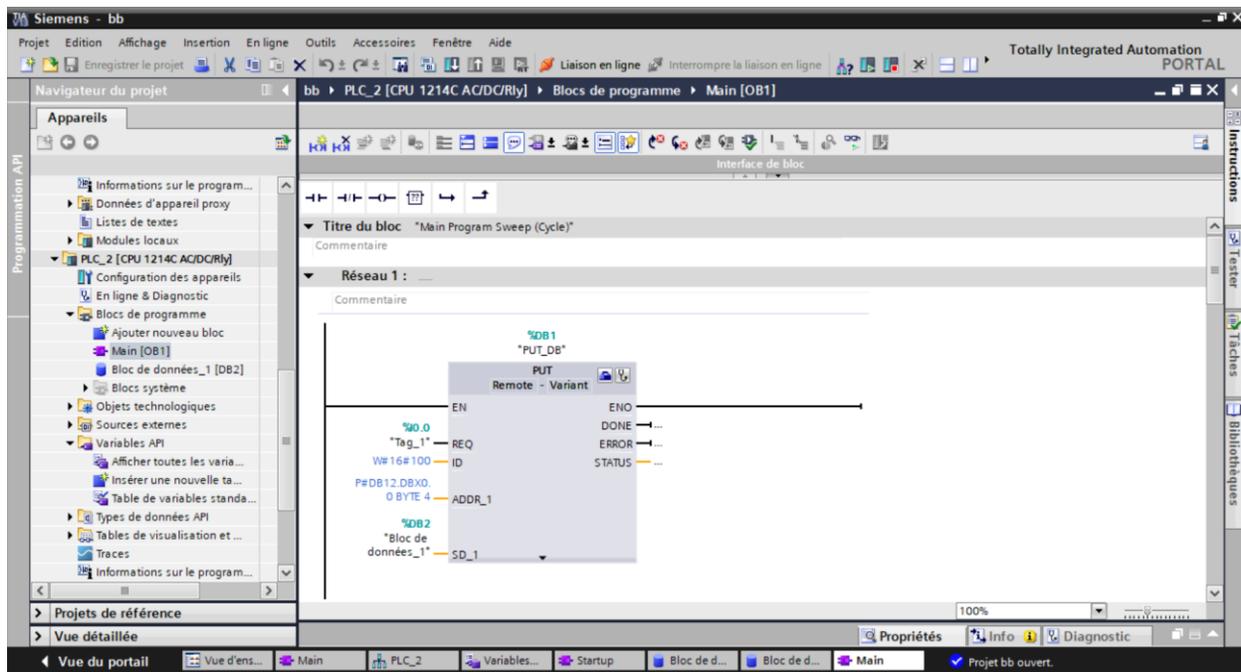
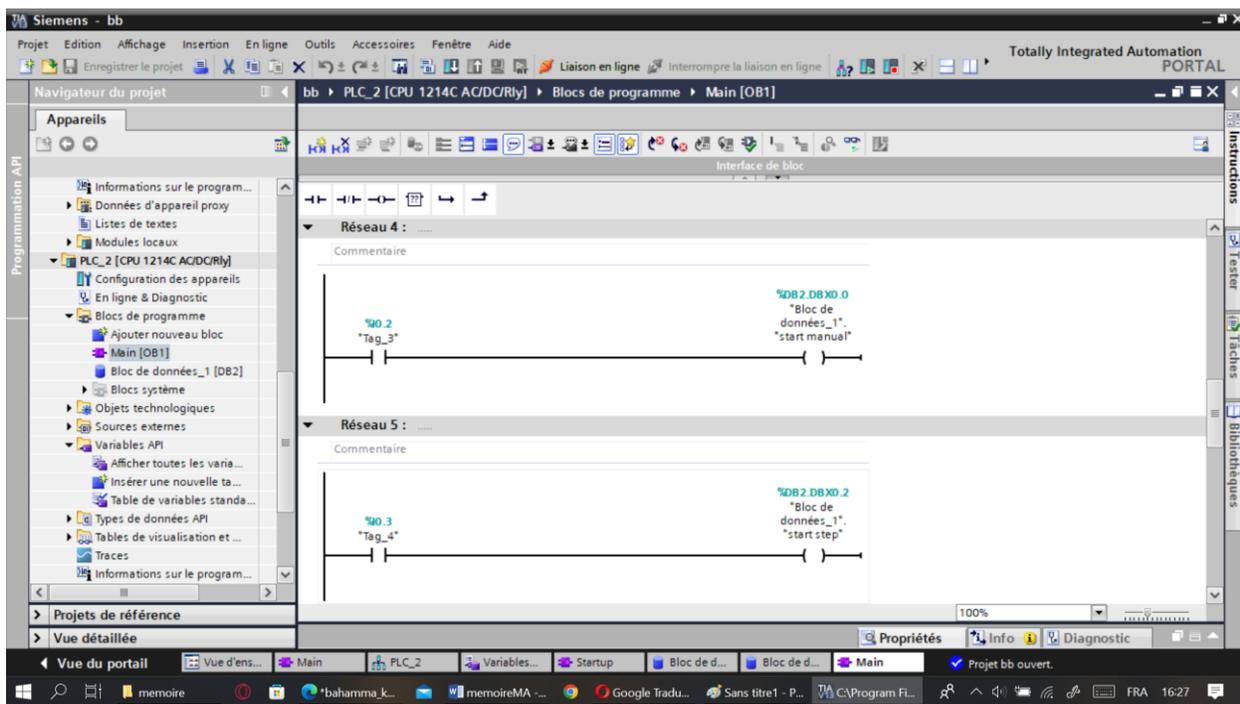
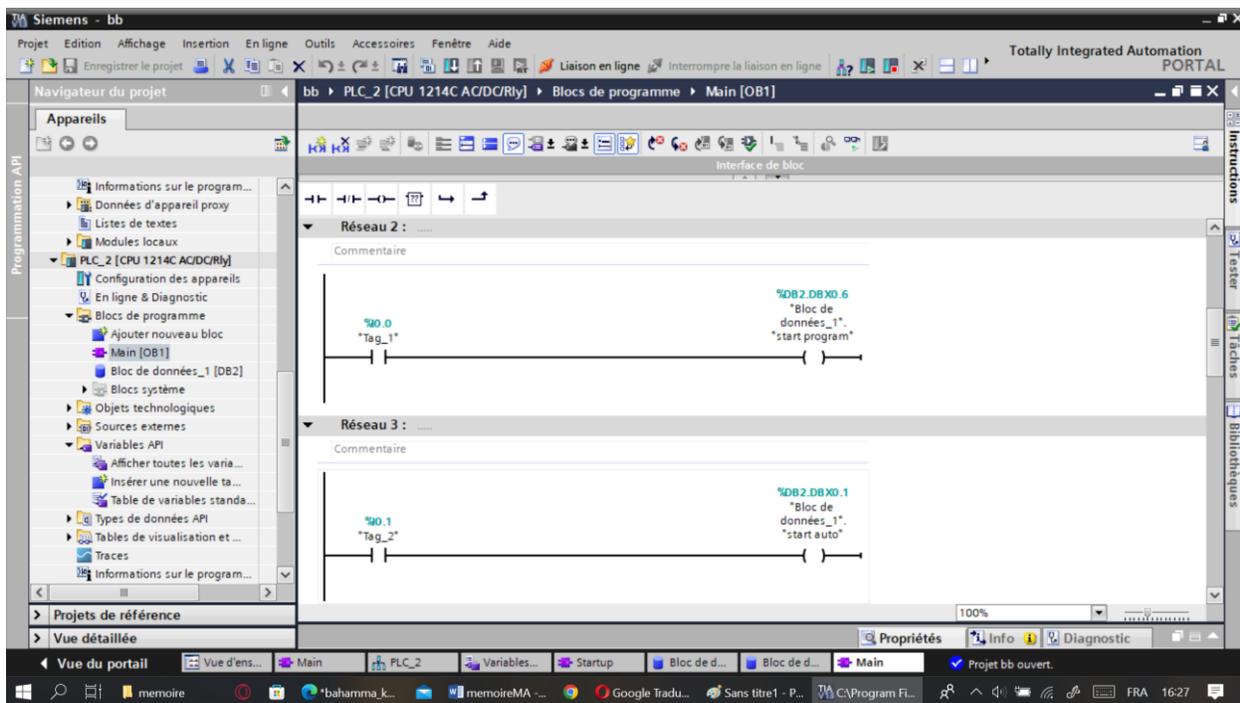


Figure 49 le programme dans l'OB CPU_1.

- CPU_2 :



IV. Chapitre : Logiciel et le travail pratique.



IV. Chapitre : Logiciel et le travail pratique.

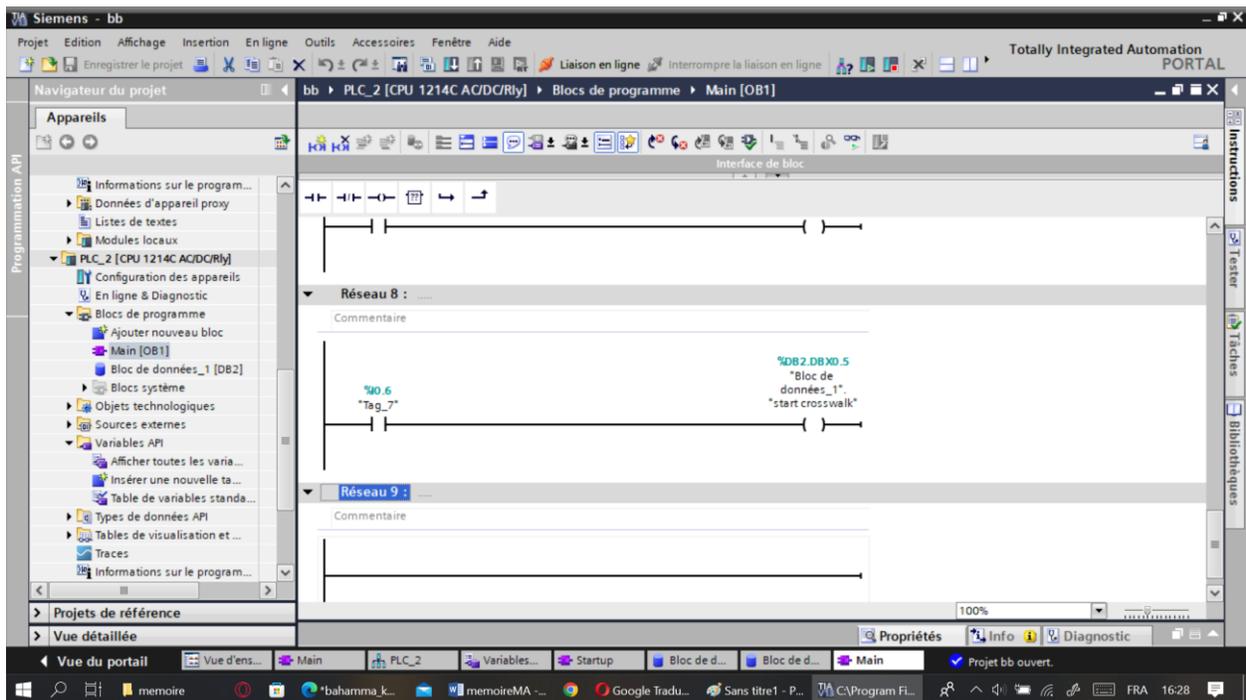
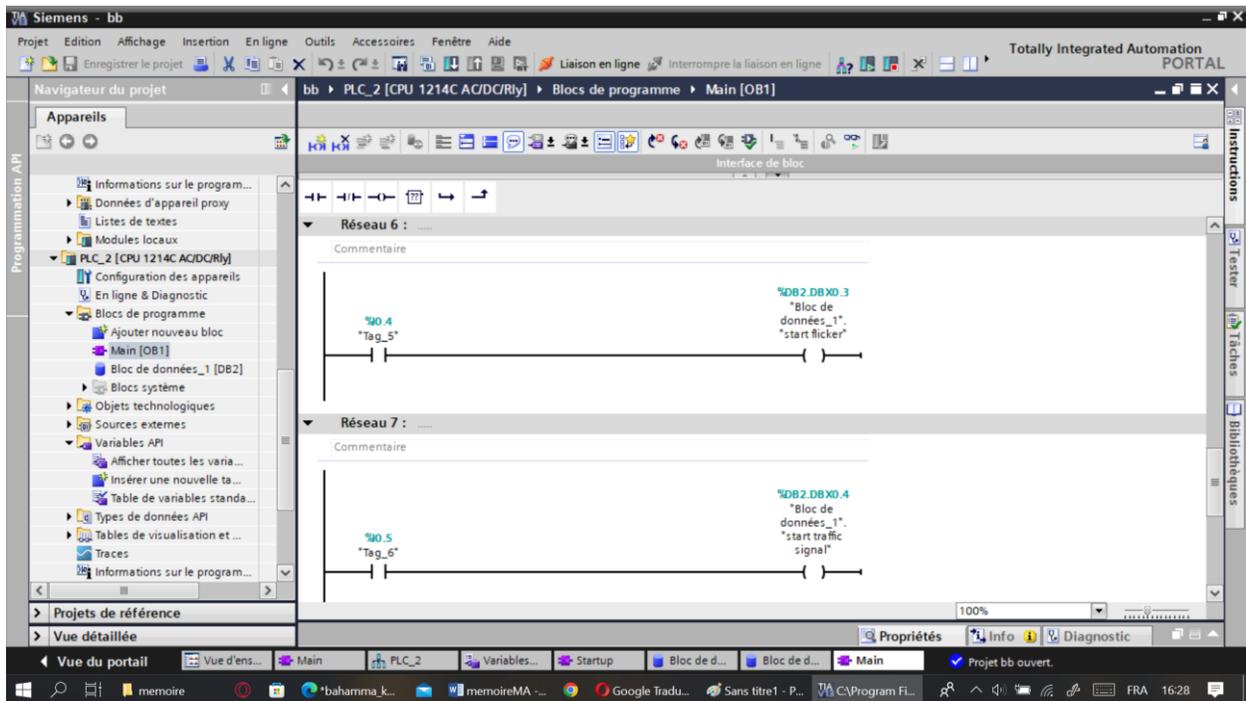


Figure 50 le programme dans l'OB CPU_2.

iv) Bloc de données :

- ❖ Les blocs de données servent à stocker les données du programme utilisateur. On fait la distinction entre les blocs de données globaux et les blocs de données d'instance :
 - Les blocs de données globaux contiennent des informations auxquelles on peut accéder à partir de tous les blocs logiques du programme utilisateur (FB, FC, OB). ils ne sont pas affectés à un bloc précis (comme dans STEP 5).
 - Les blocs de données d'instance ils sont toujours associés à un FB. Les données de ce DB ne devraient être traitées que par le FB correspondant.
- ❖ CPU_1 :

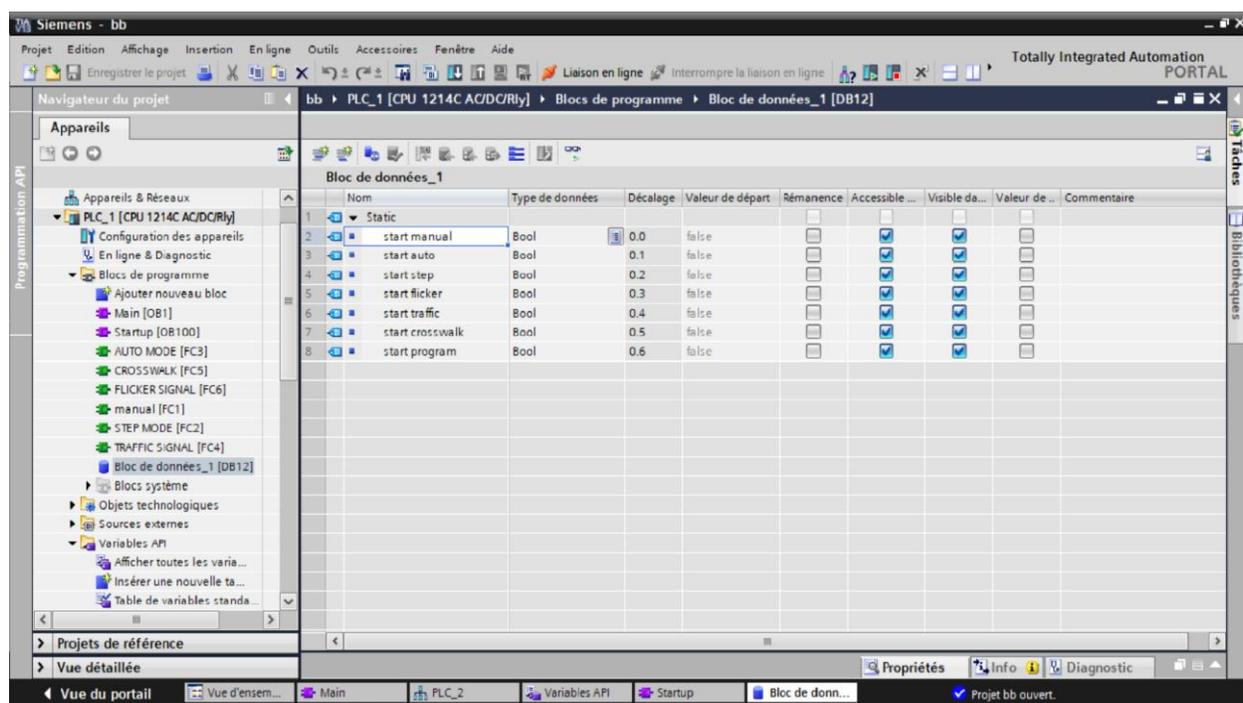


Figure 51 DB pour CPU_1.

- ❖ CPU_2 :

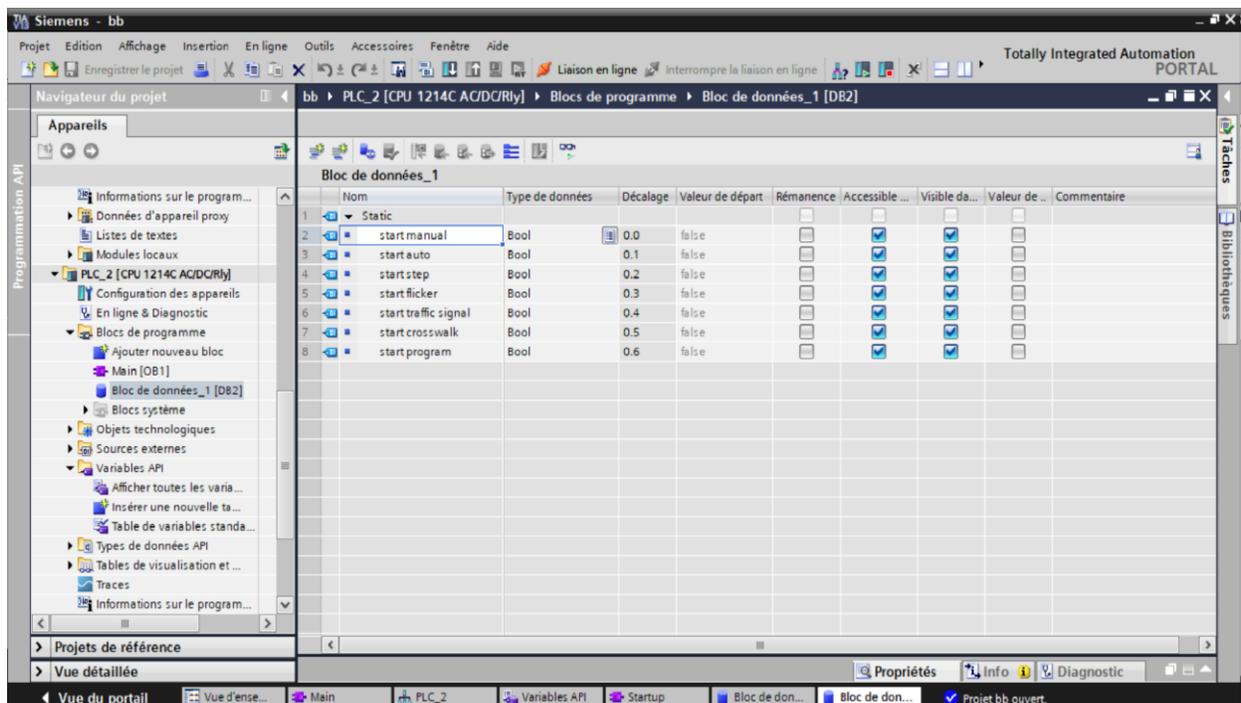


Figure 52 DB pour CPU_2.

v) Les instructions de communication PUT_GET:

- ❖ **PUT** : Ce bloc vous permet d'écrire des données dans une CPU distante lorsque la liaison ne s'effectue pas via un CP.

L'instruction est lancée en cas de front montant à l'entrée de commande (REQ). Les pointeurs désignant les zones où écrire les données (ADDR_i) et les données (SD_i) sont envoyés à la CPU partenaire.

IV. Chapitre : Logiciel et le travail pratique.

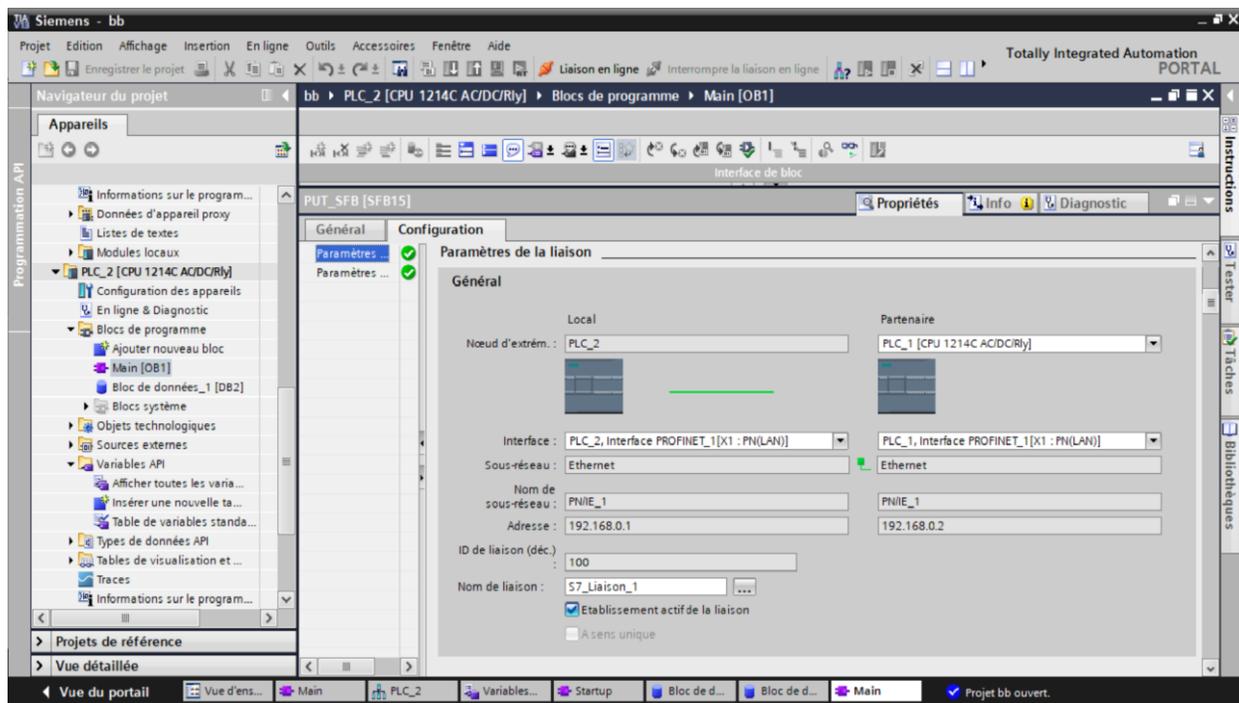


Figure 53 paramètre de la liaison de fonction PUT.

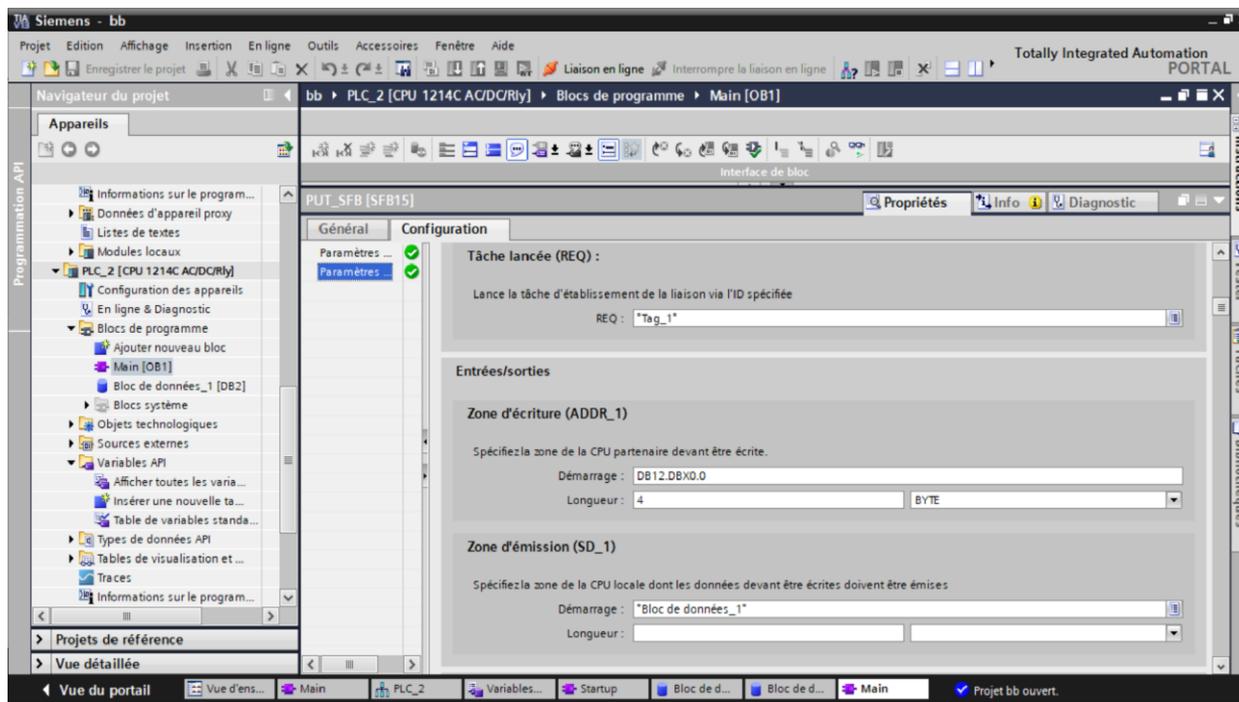


Figure 54 paramètre de bloc de fonction PUT.

❖ **GET** : ce bloc vous permet de lire des données dans une CPU distante. Bloc est lancé en cas de front montant à l'entrée de commande (REQ).

Les pointeurs requis désignant les zones où lire les données (ADDR_i) sont envoyés à la CPU partenaire. La CPU partenaire peut être à l'état de fonctionnement Marche ou Arrêt.

L'achèvement de cette opération est signalé par la valeur 1 dans le paramètre d'état NDR. [13]

La communication S7 avec les services PUT et GET est utilisé dans ce projet d'application pour l'échange de données. Dans notre travail, L'automate CPU_1 est considérée comme serveur, Alors que la deuxième automate CPU_2 est considérée comme client.

vi) Fonctionnement de PUT dans nos projet :

Dans ce projet le CPU_2 est comme un responsable à de démarrer le CPU_1 en général (bouton Start) et ces fonction (FC) une par une.

En utilisant la fonction PUT à programmer des entrées physiques (boutons physiques) pour commander les (FC) de CPU_1.

Tout d'abord, nous aurions dû donner un accès autorisé avec PUT/GET connections à les deux CPU.

Aller à la section protection dans les propriétés

IV. Chapitre : Logiciel et le travail pratique.

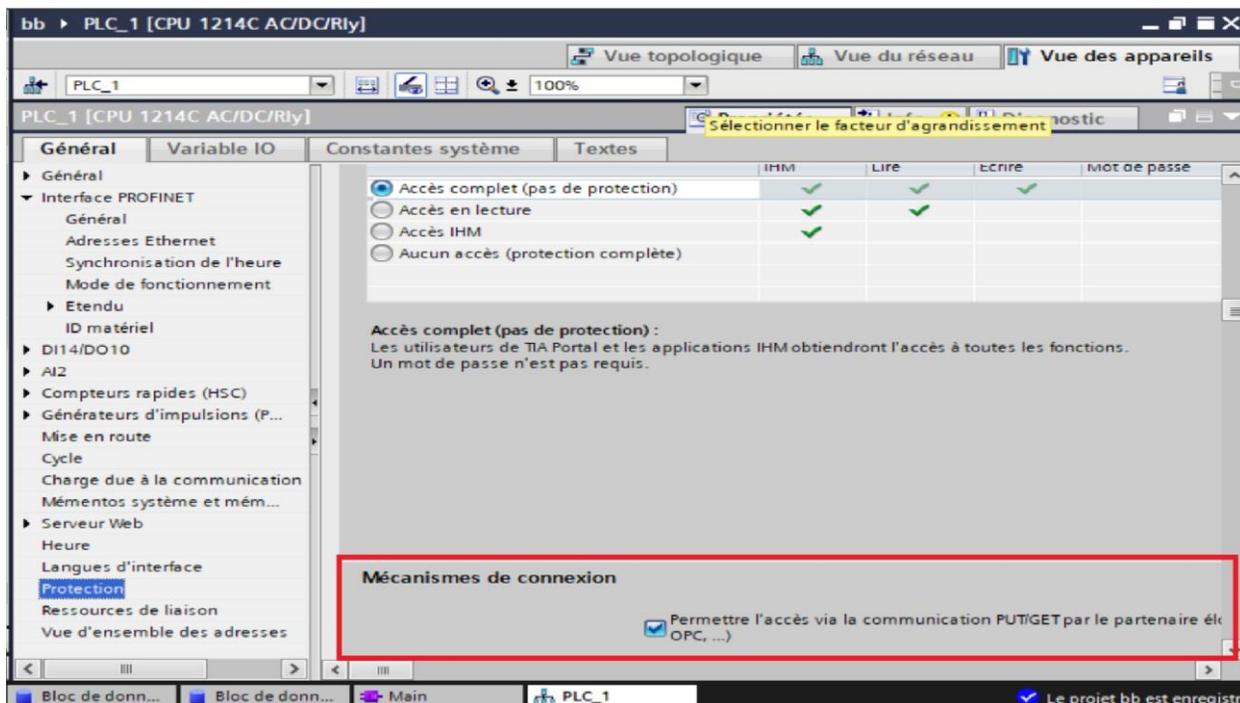


Figure 55 donne Access à PUT/GET.

En débutant et avant programmer la deuxième CPU, en crée le DB de nos data qui on utilise pour le.

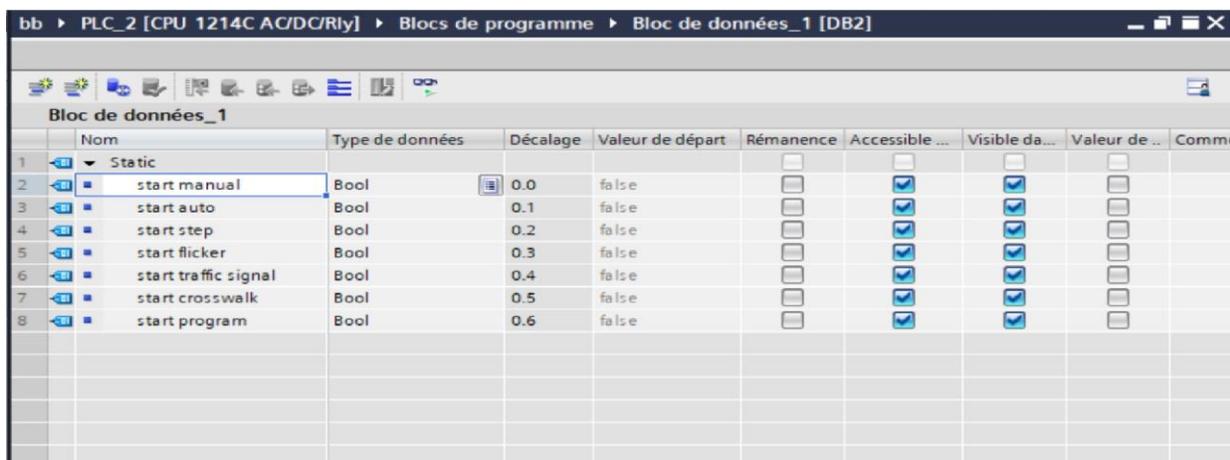


Figure 56 DB de CPU_2.

Et alors, on crée la même data dans le DB de CPU_1.

IV. Chapitre : Logiciel et le travail pratique.

The screenshot shows the 'Bloc de données_1' configuration window in SIMATIC Manager. The table lists the following data:

| Nom | Type de données | Décalage | Valeur de départ | Rémanence | Accessible ... | Visible da... | Valeur de ... | Comm |
|----------------------|-----------------|----------|------------------|--------------------------|-------------------------------------|-------------------------------------|--------------------------|------|
| Static | | | | | | | | |
| start manual | Bool | 0.0 | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| start auto | Bool | 0.1 | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| start step | Bool | 0.2 | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| start flicker | Bool | 0.3 | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| start traffic signal | Bool | 0.4 | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| start crosswalk | Bool | 0.5 | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| start program | Bool | 0.6 | false | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |

Figure 57 DB de CPU_1.

Puis, on utilise l'instruction PUT dans la deuxième CPU et configurer la.

En appuyant en,

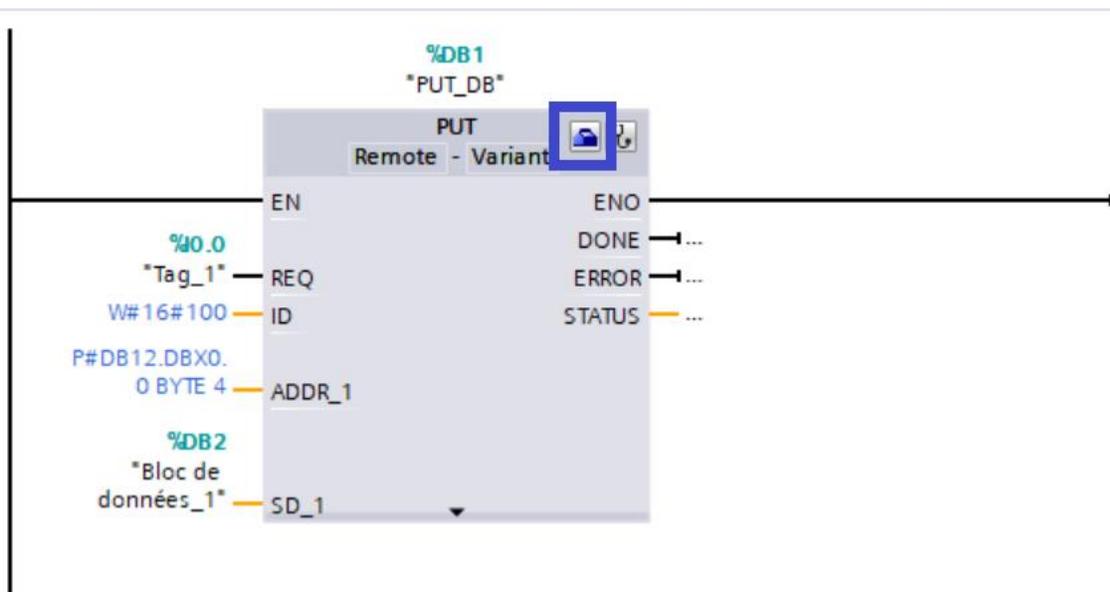


Figure 58 PUT instruction.

IV. Chapitre : Logiciel et le travail pratique.

Et ce fenêtre va ouvrir et on démarrera a configurer le bouton qui va démarrer le transfert des données (REQ) et d'où il ira chercher les données (SD_1) (bloc de donné_1 DB2) de CPU_2 jusqu'à où il les enverra (ADDR_1) le début des entrées (pointeur) nous réservant pour les boutons physique dans le CPU_1 (P#DB12.DBX0.0). Où P# définir pointeur DB12 est le DB et ce numéro, DBX0.0 adresse qu'il pointeur va point in.

1 Condition de démarrage de fonction.

2 A partir de quel octet nous commencerons à mettre nos données.

3 Le nombre de bits nous réservant (nécessaire).

4 Types de donnée.

5 D'où il ira chercher les données.

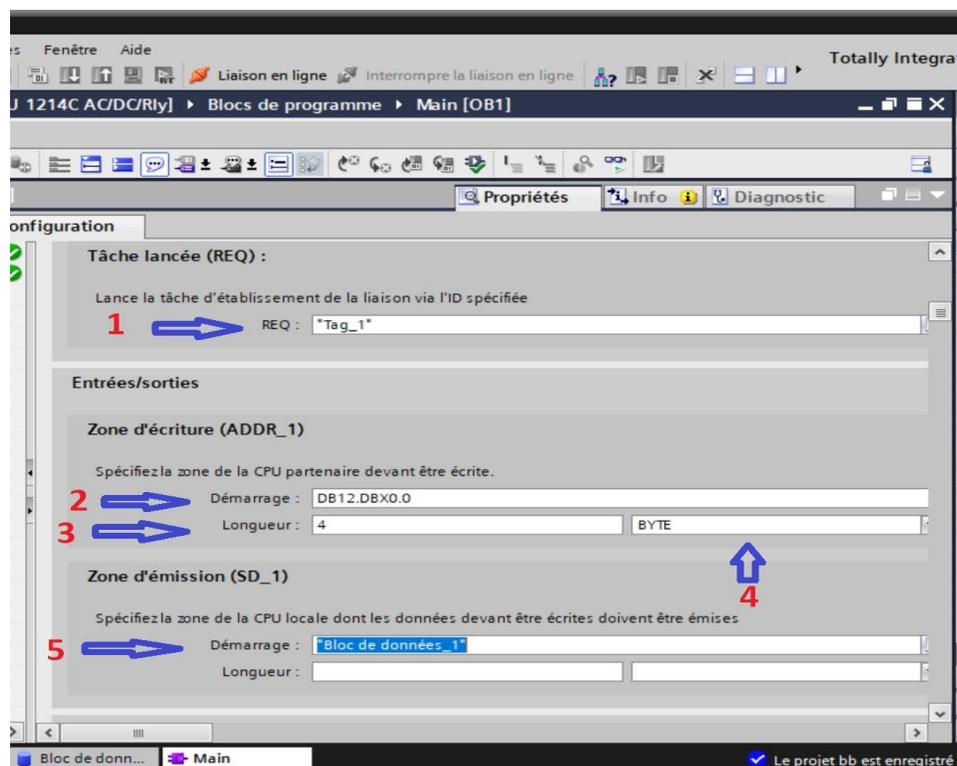


Figure 59 configuration instruction PUT.

IV. Chapitre : Logiciel et le travail pratique.

Et maintenant nous pouvons commencer à programmer en utilisant les adresses qu'il nous a données.

vii) Exécution de programme :

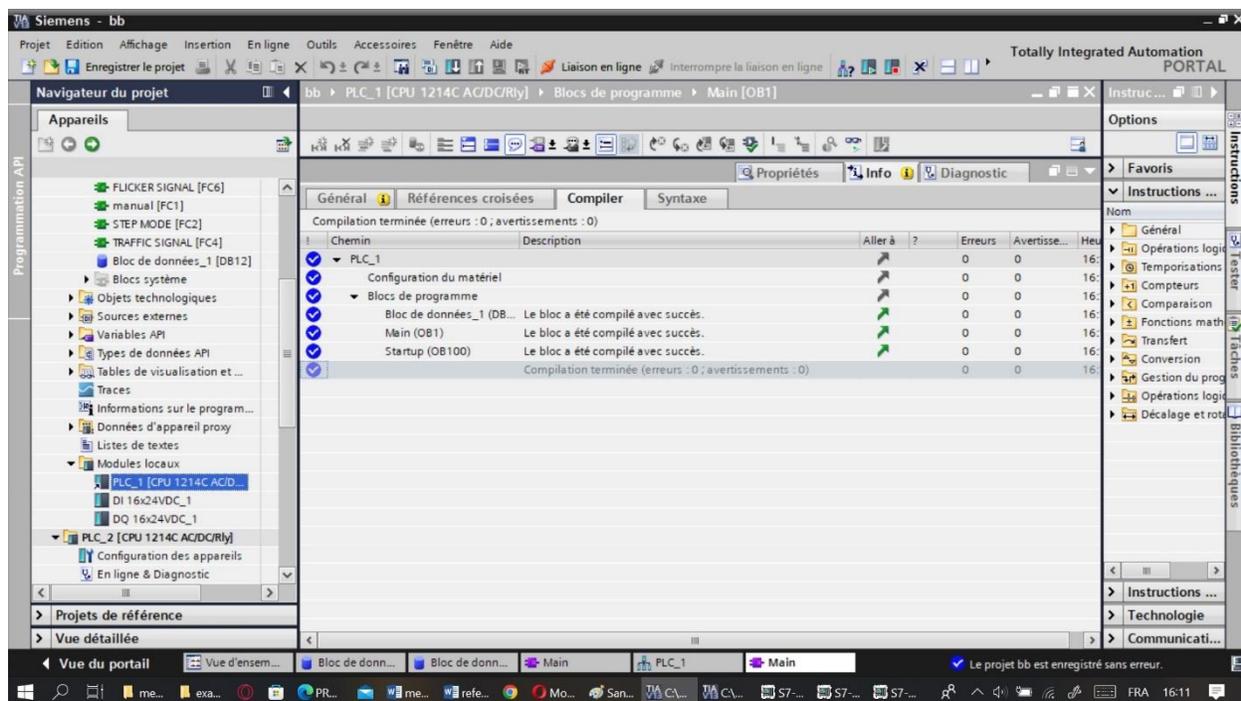


Figure 60 compilation avec succès CPU_1.

IV. Chapitre : Logiciel et le travail pratique.

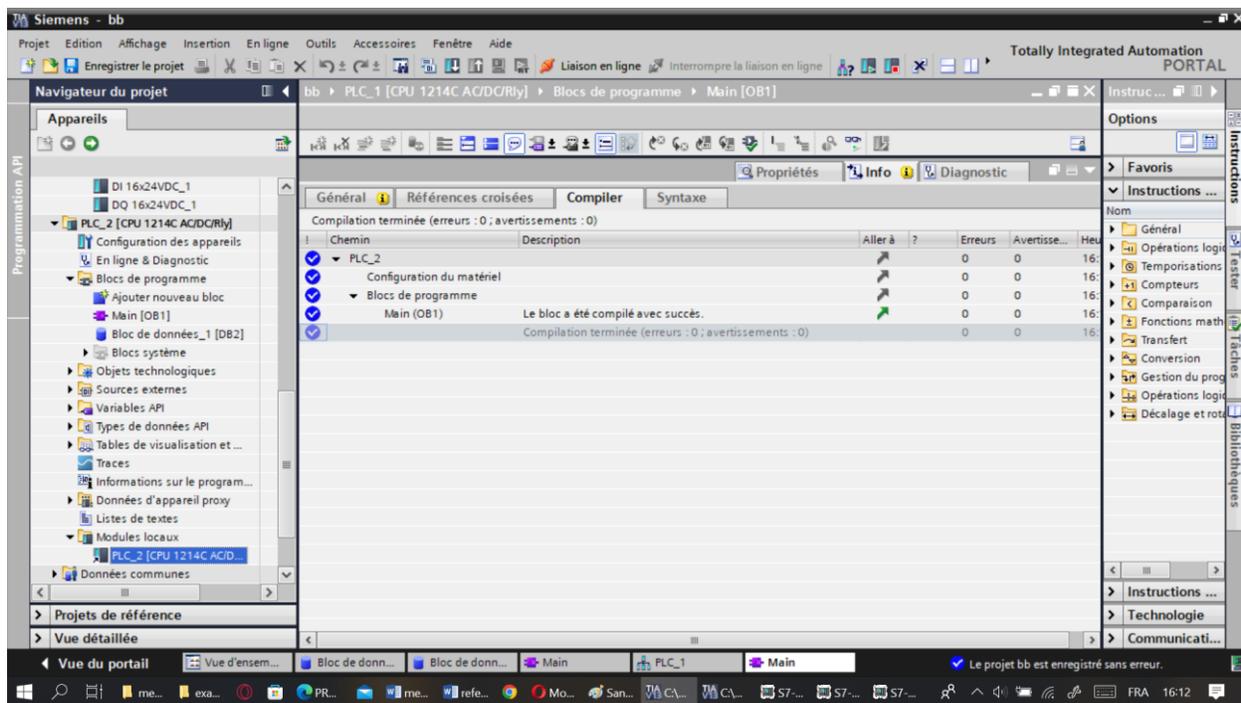


Figure 61 compilation avec succès CPU_2.

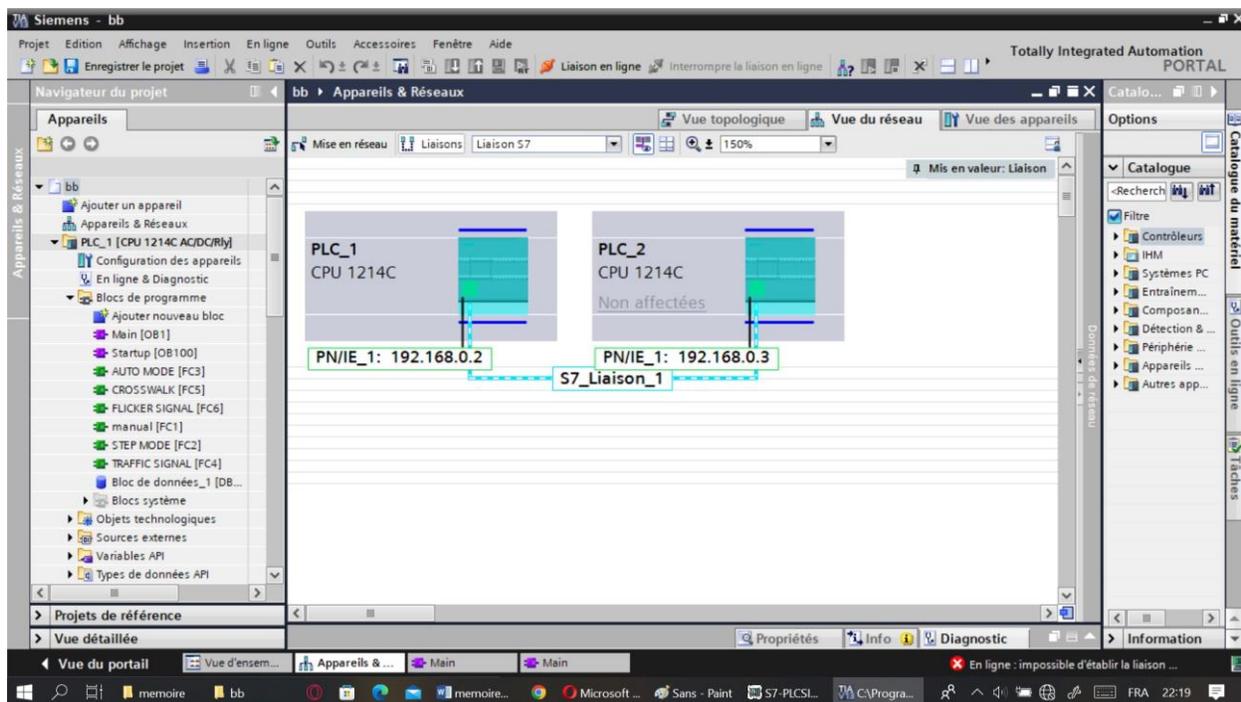


Figure 62 vue de réseau et d'adresses.

IV.7.2 Des fonctions mathématiques :

Ici, nous utiliserons une autre méthode de communication pour exécuter notre programme.

1) **Système aperçu** : Des fonctions simples entre les entrées d'un même automate et entre les deux automates.

2) **La programmation** :

a) **Configuration du matériel** :

i) **Automates** :

- CPU_1 : CPU 314C-2 PN/DP.
- N de réf : 6ES7 314-6EH04-0AB0.
- CPU_2 : CPU 315-2 PN/DP.
- N de réf : 6ES7 315-2EH14-0AB0.

ii) **Les modules E/S** :

- Module E : DI 16x24VDC_1.
- N de réf : 6ES7 321-1BH02-0AA0.
- Module S : DO 8x24VDC/2A.
- N de réf : 6ES7 322-1BF01-0AA0.

iii) **Périphérique IO** :

- Nom : IM 155-6 PN ST
- Description : Coupleur avec interface PROFINET V2.2 (RT/IRT) avec temps de cycle à partir de 250µs, connectique PN sélectionnable via adaptateur de bus, 32 modules de périphérie y compris modules F ; redondance de supports de transmission (MRP) ; contrôle de configuration via API ; remplacement de modules en fonctionnement ; démarrage avec emplacements vides ; prise en charge de PROFIenergy ; firmware V1.1

- N de réf : 6ES7 155-6AU00-0BN0.

b) Configuration Ethernet :

Dans les propriétés de la CPU, son adresse Ethernet peut également être définie. Une double-cliqué sur l'icône Ethernet d'un poste de travail fait apparaître une fenêtre d'inspection qui vous permet de définir ses propriétés.

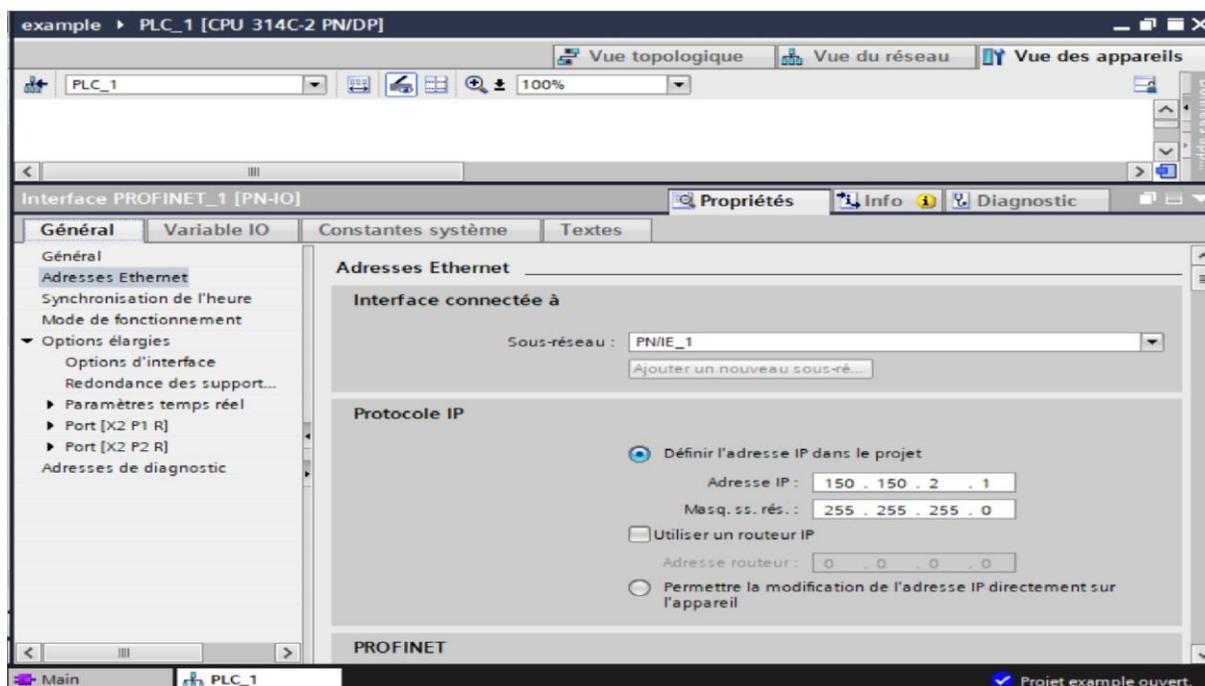


Figure 63 adressage de CPU_1.

IV. Chapitre : Logiciel et le travail pratique.

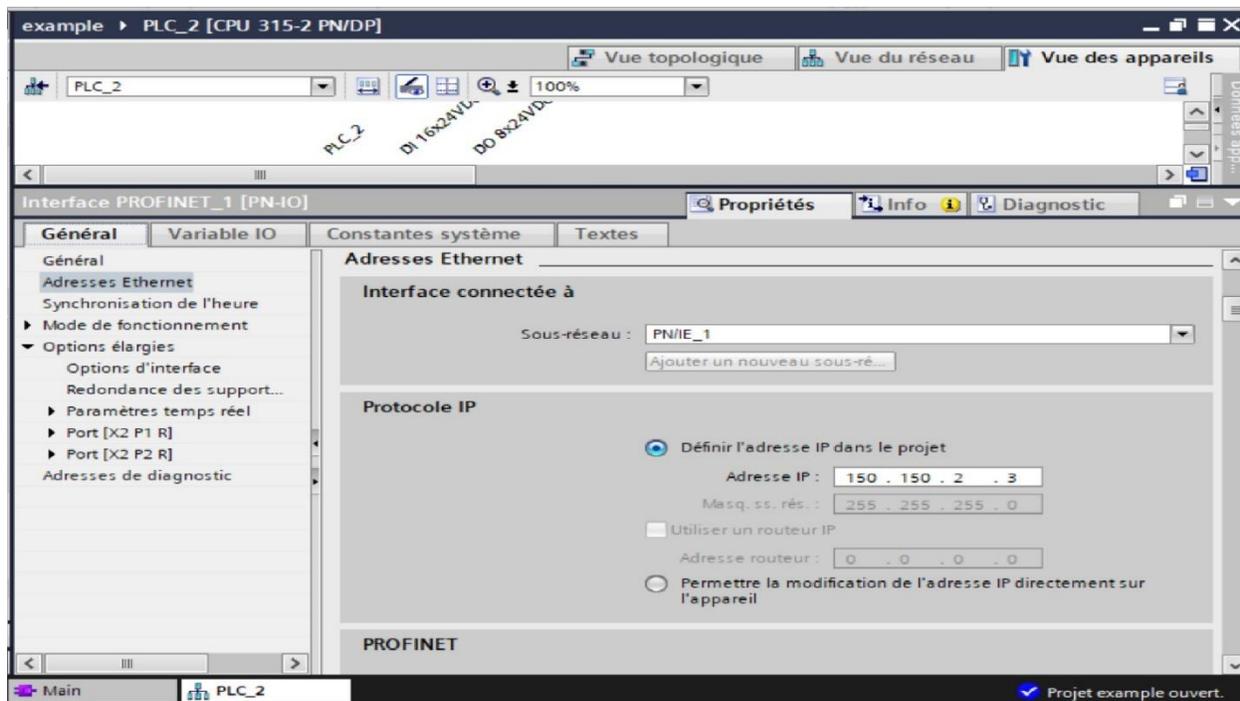


Figure 64 adressage de CPU_2.

c) Blocs de programmation :

i) Tableau des variables :

On déclare les variables nécessaires sur la table des variables dans notre système.

| | Nom | Table des variables | Type de données | Adresse | Réma... | Visibl... | Acces... | Commentaire |
|---|-----------|-------------------------|-----------------|---------|---------|-------------------------------------|-------------------------------------|-------------|
| 1 | Tag_1 | Table de variabl... | Bool | %I0.0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 2 | Tag_2 | Table de variables s... | Bool | %I0.2 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 3 | Tag_3 | Table de variables s... | Bool | %Q2.0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 4 | Tag_4 | Table de variables s... | Bool | %I4.0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 5 | Tag_5 | Table de variables s... | Bool | %I5.0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 6 | Tag_6 | Table de variables s... | Bool | %Q0.0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 7 | Tag_7 | Table de variables s... | Bool | %Q2.1 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 8 | <Ajouter> | | | | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |

Figure 65 tableau des variables CPU_1.

| | Nom | Table des variables | Type de données | Adresse | Réma... | Visibl... | Acces... | Commentaire |
|---|-----------|------------------------|-----------------|---------|---------|-------------------------------------|-------------------------------------|-------------|
| 1 | Tag_1 | Table de variabl... | Bool | %I2.0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 2 | Tag_2 | Table de variables s.. | Bool | %Q1.0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 3 | Tag_3 | Table de variables s.. | Bool | %I0.0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 4 | Tag_4 | Table de variables s.. | Bool | %Q2.0 | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 5 | <Ajouter> | | | | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |

Figure 66 tableau des variables CPU_2.

ii) Blocs d'organisation OB83 :

Le système d'exploitation de la CPU appelle l'OB83 dans les cas suivants :

- Après que vous avez débrosché ou enfiché un module configuré,
- Après que vous avez modifié des paramètres d'un module dans STEP 7 et chargé ces paramètres dans la CPU à l'état MARCHE, tout ceci dans le cadre d'une modification de l'installation en cours de fonctionnement (procédure CiR). [18]

iii) Le programme dans OB : dans notre programme on utilisant le langage LADDER, Le TIA portal fournit des Cartes de tâches qui contiennent les instructions pour le programme.

- CPU_1 :

IV. Chapitre : Logiciel et le travail pratique.

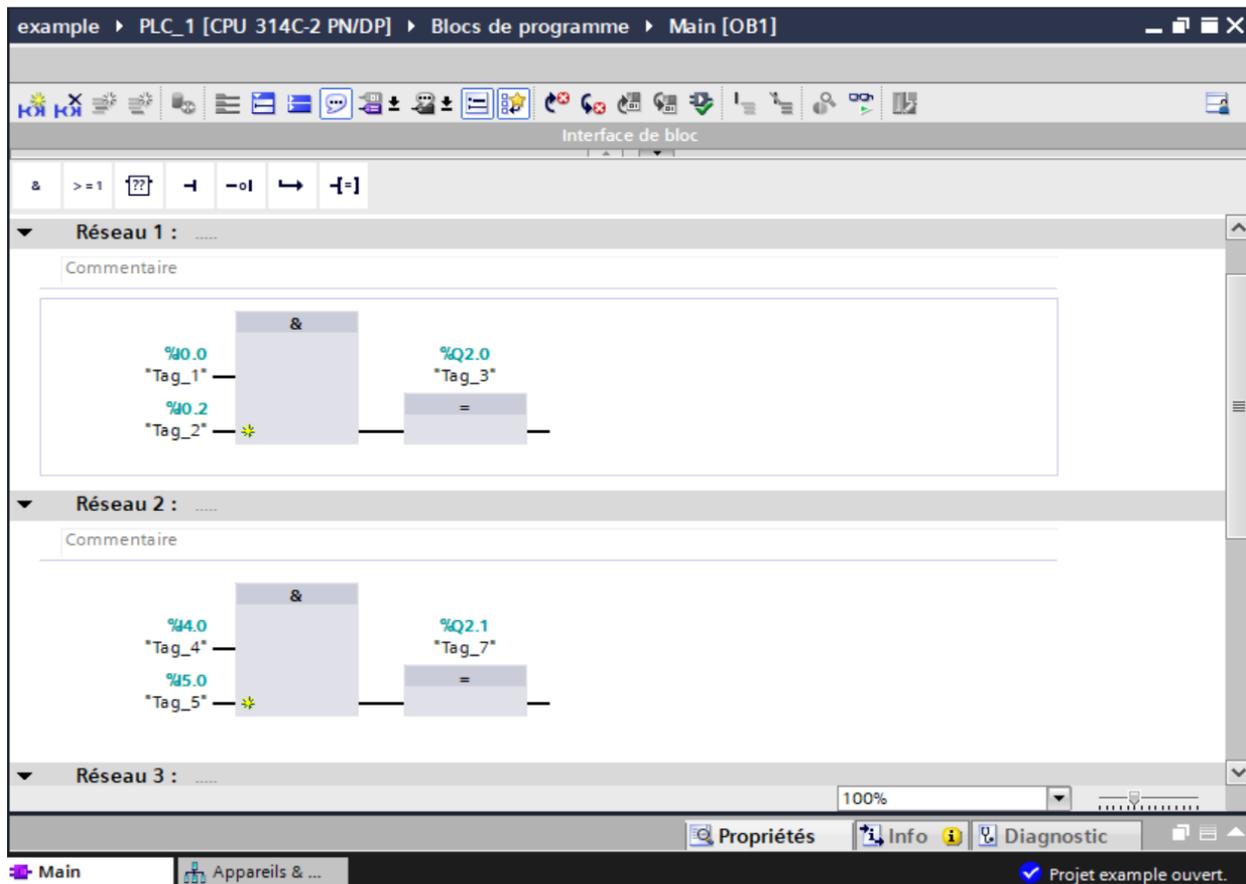


Figure 67 OB1 de CPU_1.

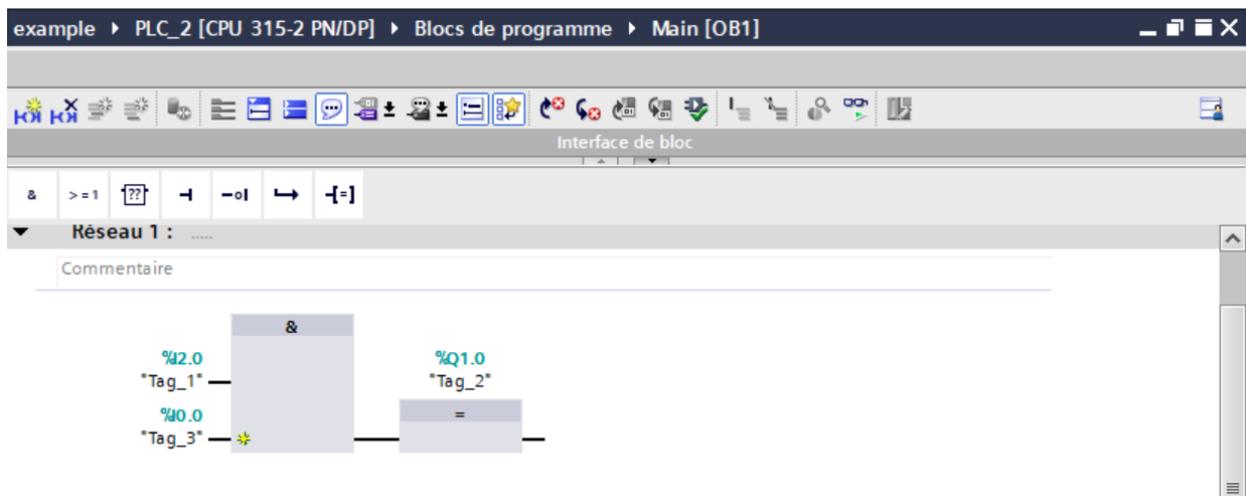


Figure 68 OB1 de CPU_2.

iv) La configuration de maître esclave :

Maître/esclave est un modèle de communication ou de contrôle asymétrique dans lequel un appareil ou un processus (le « maître ») contrôle un ou plusieurs autres appareils ou processus (les « esclaves ») et sert de centre de communication. Dans certains systèmes, un maître est sélectionné parmi un groupe d'appareils éligibles, les autres appareils jouant le rôle d'esclaves. [19]

Dans ce projet on va mettre le CPU_2 comme esclave à le CPU_2.

Et pour faire ça, nous devrions aller à la prospérité d'interface profinet de CPU_2 et à section (mode de fonctionnement) active et l'option «périphérique IO » et le TIA portal va sélectionne le comme esclave à CPU_1 automatiquement.

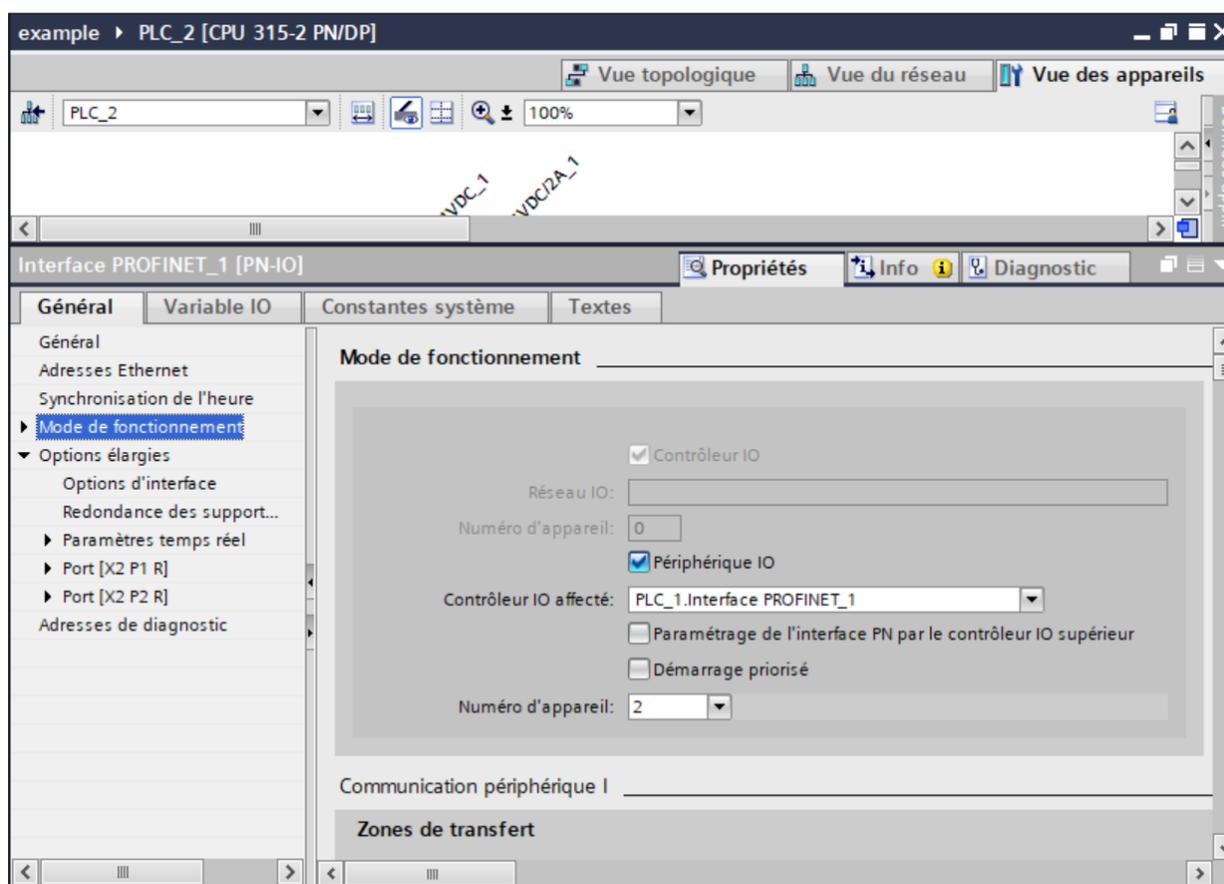


Figure 69 définir CPU_2 comme un esclave de CPU_1.

v) **Fonctionnement de notre projet :**

Après tout cette configuration nous pouvons maintenant commencer la programmation,

D'abord nous devrions créer les zones de transfert, en appuyant dans la section de (zones de transfert) autant que nous avons besoin.

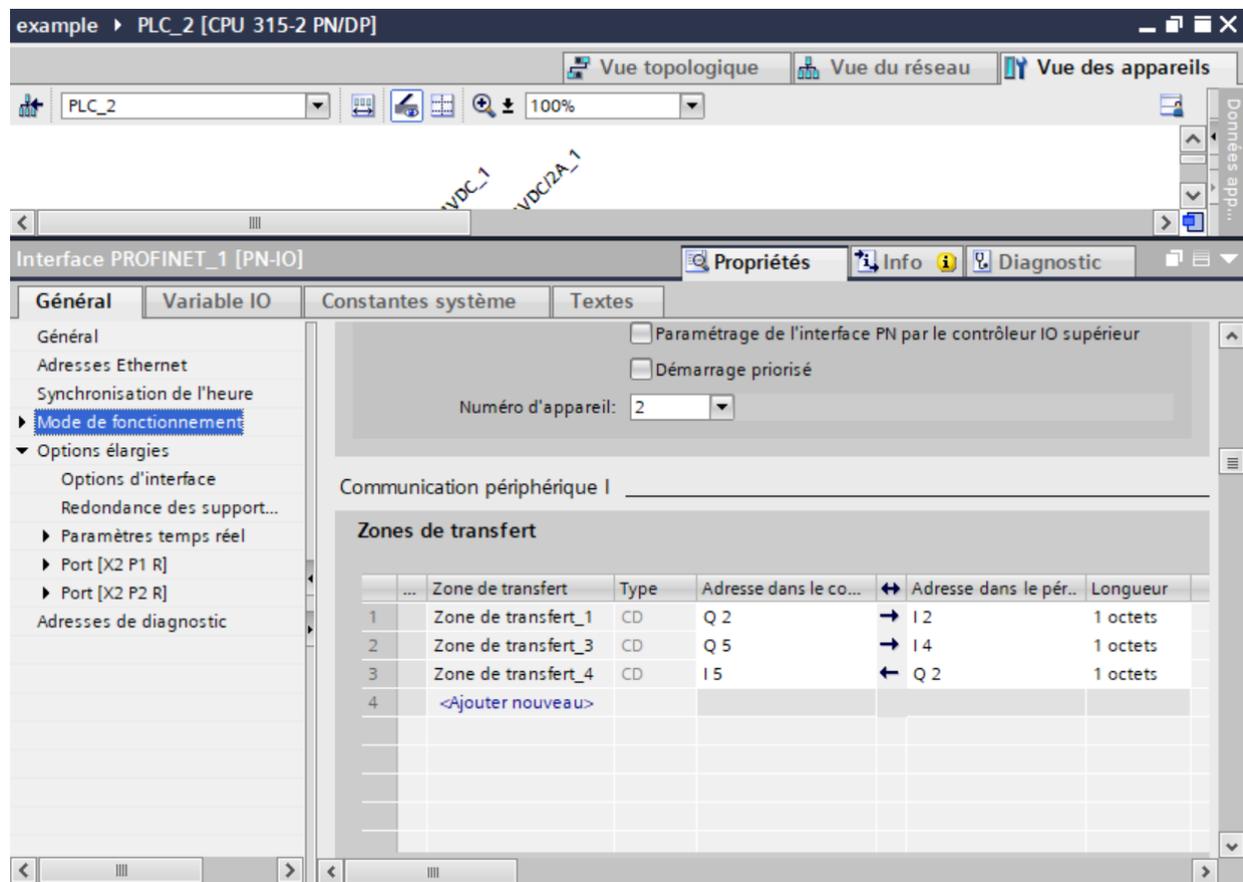


Figure 70 création des zones de transfert.

Et maintenant nous pouvons commencer à programmer en utilisant les adresses qu'il nous a données.

vi) **Exécution de programme :**

IV. Chapitre : Logiciel et le travail pratique.

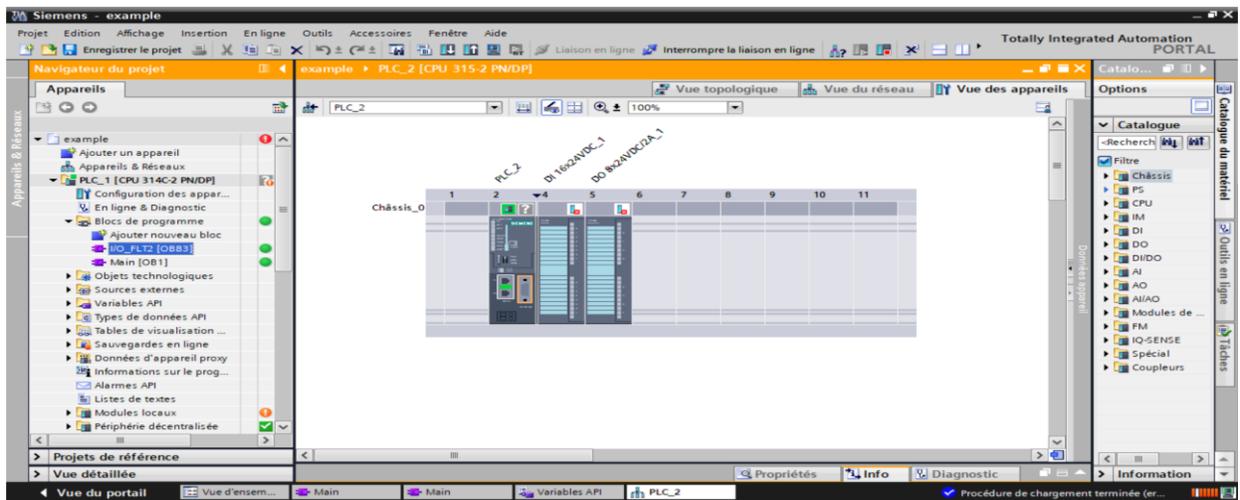


Figure 71 CPU_1 bien communiqué avec le simulateur.

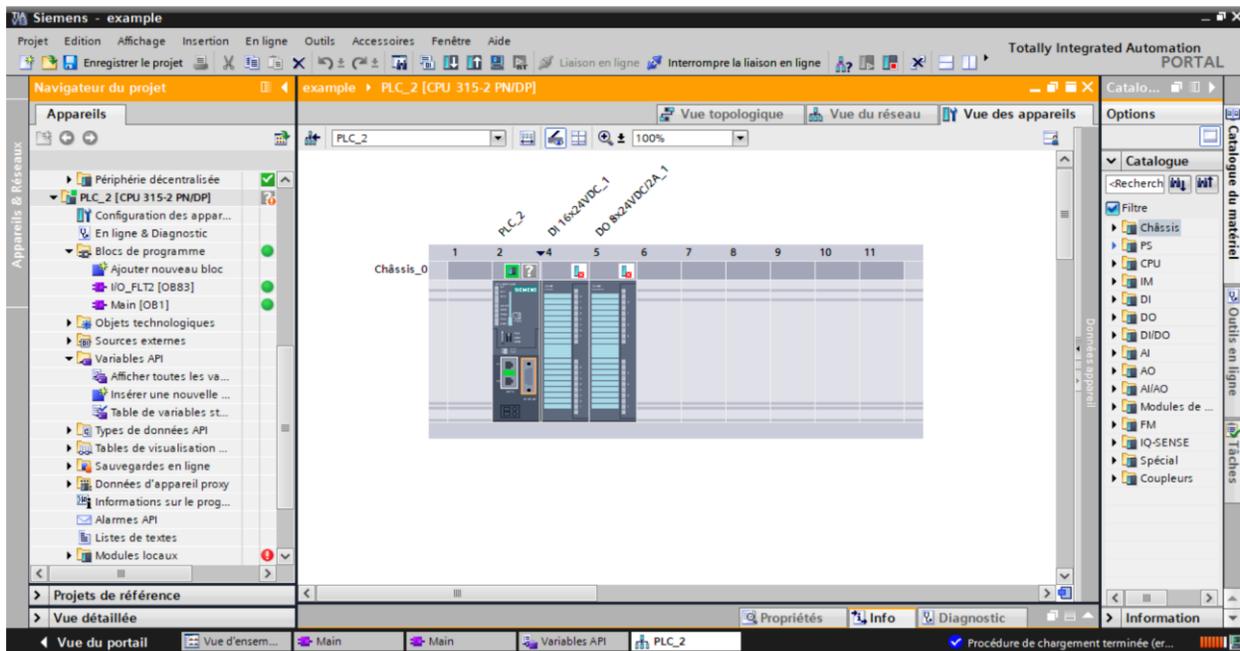


Figure 72 CPU_2 bien communiqué avec le simulateur.

IV. Chapitre : Logiciel et le travail pratique.

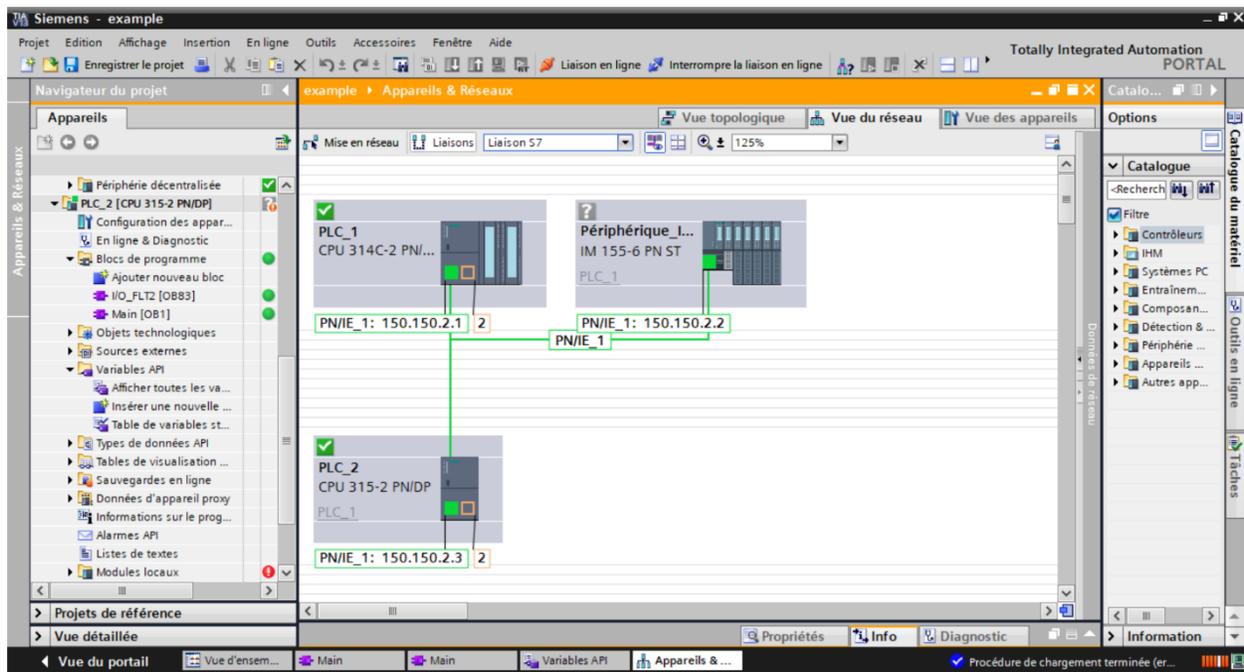


Figure 73 vue de réseau et d'adresses.

IV.8 Conclusion :

Dans ce chapitre, nous avons présenté le logiciel de programmation pour automates SIEMENS TIA portal et Périphérique IO de communication sous PROFINET IO,

On utilise deux automates programmables, et une station PC pour la simulation et la production dont portail TIA est installée en tant qu'outil de configuration, de programmation et de contrôle.

Conclusion général :

Les recherches menées dans ce projet nous permettent de comprendre un nouveau monde, à savoir Système de contrôle des instruments.

Dans la fabrication, l'automatisation industrielle est un processus intégré

Les machines et les équipements industriels automatisent des tâches telles que

Soudage, manutention, emballage, palettisation, distribution, Découpe etc...

Améliorations rendues possibles par l'utilisation de matériels et logiciels d'automatisation Productivité, sécurité et rentabilité. L'automatisation apporte de nombreux avantages correctement intégré.

Ce Projet est basé sur un réseau industriel est simulé à l'aide de deux automates et un pc avec TIA portal installé pour configurer, programmer et commander.

L'objectif de notre projet est d'étude le protocole Profinet et ses avantages et de communiquer deux API ensemble utilisant ce protocole Profinet par une ou plusieurs méthodes de communication.

Pour atteindre notre objectif nous avons bien présenté le système automatisé, les API utilisant dans celle-ci, après l'Ethernet dans l'industrie et les réseaux locaux et ces topologie et le mode OSI, avancer le travail pratique et utilise tout ça dans notre projet.

Nous espérons avoir posé une pierre dans le l'établissement d'une liaison directe entre l'université et que les protocoles de communication permettent une bonne gestion et organisation et un suivi des systèmes de production.

Bibliographie et webographie :

[1] Institut des sciences et techniques appliquées-UFM-Constantine-1 MR :
Bouchahed Adel

[2] Lapp France s.a.r.l. Technopôle Forbach-Sud BP 50084 57602 Forbach Cedex
Tél. : +33 3 87 84 19 29 www.lappfrance.fr

[3] <http://technologie-sciarretta.ovh/?p=739>

[4] FESTECH automation group

[5]

https://www.geea.org/IMG/pdf/LES_AUTOMATES_PROGRAMMABLES_INDUS_TRIELS_pour_GEEA.pdf

[6] P. Hoppenot (1999) Informatique industrielle Introduction aux Réseaux Locaux Industriels

[7] par Ouahid Belkacem, professeur d'électronique et d'informatique industrielle au lycée Jacquard à Paris 19e, avec la collaboration des experts réseaux et NTIC de Schneider Electric

[8] https://us.profinet.com/wp-content/uploads/2012/10/Short_Ethernet_Intro_slides.pdf

[9] <https://www.scribd.com/document/510208789/L3-GIM-Reseau-automates-cours2-Reseaux-Industriels-de-Communication>

[10] <https://www.controleng.com/articles/why-industrial-ethernet/>

[11] http://us.profinet.com/wp-content/uploads/2012/11/PROFINET_SystemDescription_ENG_2014_web.pdf

[12] <https://profinetuniversity.com>

[13] MÉMOIRE DE MASTER Réseaux industriel PROFINET basé sur les automates SIMENS S7-1200, Bahamma karima Le : mardi 2 juillet 2019

[14] <http://www.est-usmba.ac.ma/coursenligne/GE-S2-M8.1-Automatismes%20logiques%20Industriels-CRS-EI%20Hammoumi.pdf>

[15] MÉMOIRE DE MASTER Etude ET simulation d'un système automatisée sur le Réseaux informatique Boualag Ahmed Le: samedi 6 juillet 2019

[16] Pierre Duysinx, Geoffray Hutsemekers, Henri Lecocq " AUTOMATISATION ET ROBOTISATION DE LA PRODUCTION " UNIVERSITÉ DE LIÈGE 2009-2010.

[17] <https://sti-monge.fr/maintenancesystemes/wp-content/uploads/2013/02/Initiation-1-TIA-Portal-MS1.pdf>

[18] SIMATIC Logiciel système pour SIMATIC S7-300/400 - Fonctions standard et fonctions système - Volume 1/2 Manuel de référence.

[19] Alternatives de terminologie maître-esclave que vous pouvez utiliser dès maintenant.