



Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique

MÉMOIRE DE MASTER

Sciences et Technologies
Electronique
Electronique des Systèmes embarqués

Réf. : Entrez la référence du document

Présenté et soutenu par :
Bahamma karima

Le : mardi 2 juillet 2019

Réseaux industriel PROFINET basé sur les automates SIMENS S7-1200

Jury :

M.	Guesbaya tahar	MCA	Université de Biskra	Président
M.	Rahmani nacereddine	MAA	Université de Biskra	Examineur
M.	Bakhouche khaled	MCA	Université de Biskra	Rapporteur

REMERCIEMENTS

REMERCIEMENTS

Je remercie tous d'abord dieu le tout puissant et miséricordieux de m'avoir donné la Force, le courage, et la patience d'accomplir ce modeste travail.

Je tiens à remercier très particulièrement mon directeur de thème, le docteur Bakhouche Khaled qui m'a proposé ce sujet de recherche, et de m'avoir encadré et dirigé toute au long de ce parcours, mais surtout pour ses conseils et son expérience qui ont été décisifs dans l'accomplissement de ce travail.

Je remercie vivement monsieur Lakhdari issam de ma voir accueilli durant ma Période de stage au sein biskria ciment, je le remercie également pour ses conseils, son soutien et ces encouragements.

Mes plus profonds remerciements sont adressés à ma famille et mes amies en particulier mes parents et mon cher oncle pour leur patience et pour soutien qu'ils m'ont apporté tout au long de mon cursus universitaire. J'espère de tout mon cœur que la réalisation de ce travail soit un aboutissement à leurs efforts et espoir. Qu'ils trouvent ici l'expression de ma gratitude la plus affectueuse.

Dédicaces

Je dédie ce modeste travail à:
A mes parents .Aucun hommage ne pourrait être à la hauteur de
l'amour Dont ils ne cessent de me combler. Que dieu leur procure
bonne santé et longue vie.

RESUME

RESUME

L'objectif de travail présenté dans ce thème est, d'une part, de contribuer à la conception complète d'un réseau Profinet dans un système automatisé et d'autre part, de l'implémenter en utilisant des techniques de programmation concurrente. Le cas étudié, dans ce thème, un réseau industriel simulé et réalisé à l'aide de deux automates programmables S7-1200, un switch classique et une station PC dont TIA Portal est installé comme outil de configuration, de programmation et de commande. En se basant sur les bus PROFINET, le logiciel TIA Portal est utilisé pour le contrôle temps réel ainsi que pour la simulation. La supervision de notre système est assurée par WinCC RT.

ملخص

الهدف من العمل المعروف في هذا الموضوع هو، من ناحية ، للمساهمة في تصميم كامل لشبكة بروفينت (Profinet) في نظام آلي ، و من ناحية أخرى تحقيقها باستعمال تقنيات برمجة منافسة. تتعلق الحالة التي تم محاكاتها و انجازها في تكوين شبكة مكونة من جهازي PLC من نوع S7-1200 و مبدل Ethernet و جهاز حاسوب مثبت عليه برنامج TIA Portal مع WinCC RT.

Sommaire

Sommaire

Introduction générale.....	1
-----------------------------------	----------

Chapitre I: Automates Programabel industriel

I.1 Introduction.....	2
I.2 Définition d'un API.....	2
I.3 Domaines d'emploi des automates.....	2
I.4 Communication d'API.....	3
I.5 Choix des API.....	3
I.6 Architecture des automates programmables industriels	3
I.6.1 Structure générale des API.....	4
I.6.2 Architecture interne d'un API.....	4
I.6.3 Description des éléments d'un API.....	5
I.6.3.1 La mémoire.....	5
I.6.3.2 Le processeur.....	6
I.6.3.3 Les interfaces et les cartes d'Entrées / Sorties.....	7
I.6.3.4 Une alimentation électrique.....	7
I.6.3.5 Liaisons de communication.....	8
I.7 Nature des informations traitées par l'automate.....	8
I.8 Langages de programmation pour API.....	9
I.8.1 Le langage LD (Ladder Diagram).....	9
I.8.2 Le langage IL (Instruction List)	9

Sommaire

I.8.3 Le langage FBD (Functions Block Diagram).....	9
I.8.4 Le langage ST (Structured Text)	10
I.8.5 Le langage SFC (Sequential Function Chart), ou GRAFCET.....	10
I.9 Présentation de l'automate S7-1200.....	10
I.9.1 Choix de la CPU.....	11
I.9.2 Le choix des modules d'Entrées/Sorties.....	11
I.9.2.1 Les signaux TOR.....	12
I.9.2.2 Les signaux analogiques.....	12
I.10 Conclusion.....	13

Chapitre II :Profinet

II.1 Introduction.....	14
II.2. Objectifs et avantages de PROFINET.....	14
II.3. Simplicité de câblage.....	15
II.3.1 Sécurité intégrée.....	15
II.3.2 Standards TIC & accès sécurisé.....	15
II.3.3 Processus.....	15
II.3.4 Appareils de terrain décentralisés.....	16
II.3.5 Motion Control.....	16
II.3.6 Intelligence répartie.....	16
II.3.7 Communication Profinet	16
II.4 Installation du réseau.....	18
II.4.1 Commutateurs (switches).....	18

Sommaire

II.4.2 Topologies des réseaux.....	19
II.4.2.1 Étoile.....	19
II.4.2.2 Arbre.....	20
II.4.2.3 Ligne.....	20
II.4.2.4 Anneau (redondant).....	20
II.4.3 Profinet et modèle OSI.....	21
II.5 Type profinet.....	22
II.5.1 PROFINET CBA	22
II.5.2 PROFINET IO.....	22
II.5.2.1 Le fonctionnement de Profinet IO.....	23
II.5.2.1.1. Contrôleur d'E/S.....	23
II.5.2.1.2. Périphérique d'E/S.....	23
II.5.2.1.3 Superviseur d'E/S.....	23
II.5.3 Attribution d'adresse.....	24
II.6 Conclusion.....	24

Chapitre III: Logiciel et programmation

III.1 Introduction.....	26
-------------------------	----

Sommaire

III.2 TIA Portal V13.....	26
III.2.1 Les avantages du logiciel TIA portal.....	27
III.3 Création d'un nouveau projet SUR TIA PORTAL.....	27
III.3.2Créer un projet	28
III.3.3Configurer un appareil	29
III.3.4 Vue du projet.....	30
III.3.5Configuration matérielle.....	31
III.3.5.1Adressage des E/S.....	31
III.3.5.2 Les variables API.....	32
III.3.5.2.1 Adresses symbolique et absolue.....	32
III. 3.5.3 Table des variables API.....	32
III.3.6 Adresse Ethernet de la CPU.....	32
III.4 Configure-t-on, dans WinCC (TIA Portal), une liaison entre un pupiter IHM ou un Wincc RunTime Advanced.....	33
III.4.1Variables HMI.....	35
III.5 Programmation.....	36
III.5.1Les blocs.....	36
III.5.1.1Les blocs d'organisation.....	36
III.5.1.2Blocs fonctionnelles (FB).....	37
III.5.1.3Bloc fonction (FC).....	37
III.5.1.4Blocs de données	37
III.5.2Blocs système de communication PUT ET GET.....	38
III.5.3Programmation de la communication Client/serveur	39
III.5.4Liaison avec l'automate.....	42
III.5.4.1Paramétrage de l'interface PG/PC.....	42
III.5.5WinCC sur TIA portal.....	42
III.6Compilation et simulation.....	45

Sommaire

III.6.1 PLCSIM.....	45
III.6.2RUNTIME.....	45
III.7Réalisation pratique.....	46
III.8Conclusion.....	47
Conclusion générale.....	48
Bibliographies.....	49

Sommaire

Sommaire

Liste des figures

Liste des figures

Figure I.1: Un automate programmable industriel (API).....	2
Figure I.2. Automate programmable siemens avec modules associés.....	4
Figure I.3. Structure interne d'un Automate Programmable Industriel.....	5
Figure I.4. La mémoire d'un API.....	6
Figure I.5. Interfaces des E/S d'un API.....	7
Figure I.6. Modèle de base des communications.....	8
Figure I.7. L'automate programmable S7-1200.....	11
Figure I.8. Module 16 DI / 16 DO.....	12
Figure I.9. Module AI4.....	13
Figure II.1 : Topologie des réseaux en étoile.....	20
Figure II.2 : Topologie des réseaux en bus.....	20
Figure II.12 : Topologie des réseaux en anneau.....	21
Figure II.4 : profinet dans le modèle OSI.....	21

Liste des figures

Figure III.1 : Vue du TIA Portal.....	28
Figure III.2 : Créer un projet.....	28
Figure III.3 : Configurer un appareil.....	29
Figure III.4 :Vue du projet.....	30
Figure III.5 : Adressage des E/S.....	31
Figure III.6 : Adresse Ethernet de la CPU.....	33
Figure III.7 : ajouter un appareil.....	33
Figure III.8 :l'adresse IP et le masque de sous-réseau de la station runtime.....	34
Figure III.9 : la communication entre PLC1 et PLC2 et Pc-System.....	35
Figure III.10 : Bloc d'organisation OB1 et OB100 et BD et FB.....	37
Figure III.11 : Blocs de données.....	38
Figure III.12 : bloc PUT.....	39
Figure III.13 : bloc GET	40
Figure III.14 : Paramétrage de l'interface PG/PC.....	42
Figure III.15 : WinCC sur TIA portal.....	43
Figure III.16 : Animation d'une X_Adv_Ret.....	43
Figure III.17 : Le simulateur de S7-1214C.....	45
Figure III.18 : Réalisation pratique	47

Liste des figures

Symboles et abréviations

Symboles et abréviations

ISO	Organisation internationale de normalisation
OSI	Interconnexion des systèmes ouverts
MAC	Contrôle d'accès au support
LLC	Contrôle de lien logique
API	L'automate programmable industriel
E/S	Entrées / Sorties
T.O.R	Tout ou rien
IL	Instruction List
LD	Ladder Diagram
FBD	Functions Block Diagram
ST	Structural Text
SFC	Sequential Function Chart
RT	temps réel
IRT	temps réel isochrone (IRT)
CBA	Component Based Automation
TIA	Totally Integrated Automation

Introduction générale

Les réseaux industriels sont désormais incontournables dans le monde de l'automatisme pour bien exploiter une installation. Les réseaux industriels apportent une grande souplesse aux systèmes de contrôle / commande, ils diminuent les coûts de câblage, ils offrent des possibilités nouvelles pour le contrôle et la supervision des installations, tant pour les équipes d'exploitation que de maintenance, de production ou de gestion.

Les applications automatisées recouvrent des domaines multiples et variés dans des contextes tout aussi divers, avec des contraintes et des exigences qui leur sont propres. Par conséquent, tout à la fois pour des raisons historiques, économiques et technologiques, l'offre de solutions en réseaux industriels de communication est pléthorique.

Un certain nombre d'autres solutions de réseaux industriels de communication qui n'ont pas su ou pas pu évoluer ni se déployer sont devenues anecdotiques ou ont tout bonnement et tout simplement disparu.

Le but de ce travail est de concevoir un réseau industriel, le simuler et le réaliser à l'aide de deux automates programmables, un Switch et une station PC dont TIA Portal est installé comme outil de configuration, de programmation et de commande.

Ce mémoire comporte trois chapitres qui sont répartis comme suit :

Après une présentation, au chapitre I, de l'automate programmable industriel (API). Nous avons expliqué, au chapitre II, le bus de communication Profinet. Au chapitre III, nous avons illustré, le logiciel de programmation des automates SIEMENS et le logiciel de la supervision WinCC Advanced et simulé et réalisé un réseau industriel.

Chapitre I

Automate programmable industriel (API)

I.1 Introduction

Les premiers API ont été introduites en 1969 aux Etats-Unis pour satisfaire les besoins de l'industrie automobile. Le but recherché était de remplacer les armoires à relais utilisées pour l'automatisation des chaînes de fabrication par des équipements moins coûteux et surtout Plus faciles à modifier.

I.2 Définition d'un API

L'automate programmable industriel (API), ou en anglais 'Programmable Logic Controller' (PLC), est une machine électronique programmable destinée à piloter dans une ambiance industrielle et en temps réel des procédés logiques séquentiels. Autrement dit, un Utilisateur (censé être un automaticien) l'utilise pour le contrôle et essentiellement la commande d'un procédé industriel en assurant l'adaptation nécessaire entre tout ce qui est de grande puissance par rapport à ce qui est de faible puissance côté commande. Son objectif principal est de rendre tout le mécanisme de type "laisser-faire-seul" : le système contrôle ses sorties, décide et agit sur ses entrées afin de maintenir le fonctionnement comme prévu par l'utilisateur. C'est le principe de l'automatisme [1].

Un API est destinée à automatiser les tâches les plus nombreuses de l'industrie, afin d'assurer la commande des pré-actionneurs et actionneurs à partir d'information logique, analogique ou numérique (Figure. I.1) [2].

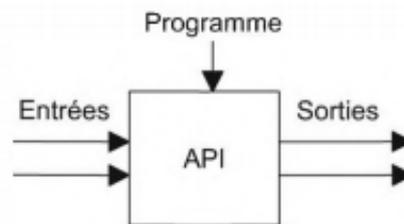


Figure I.1 : Un automate programmable industriel (API).

I.3 Domaines d'emploi des automates

Les API s'adressent à des applications que l'on trouve dans la plupart des secteurs Industriels.

Ces machines fonctionnent dans les principaux secteurs suivants [3] :

- Métallurgie et sidérurgie.
- Mécanique et automobile.
- Industries chimiques.

- Industries pétrolières.
- Industries agricoles et alimentaires

I.4 Communication avec un API

L'automate doit pouvoir se connecter et dialoguer avec d'autres matériels et les agents d'exploitation. L'API ne se limite pas à communiquer avec le processus qu'il pilote via ses modules d'E/S. Parmi les autres types de relations susceptibles d'être assurées, on cite :

- La communication avec un opérateur par un pupitre ou un terminal industriel.
- L'affichage local de valeurs numériques ou de message.
- Les échanges d'informations avec d'autre API ou système de commande.
- Les échanges d'informations avec des capteurs et actionneur intelligents.
- Les échanges d'informations avec un superviseur.
- Les échanges d'informations avec un processeur maître, ou avec des esclaves, dans le cadre d'un réseau [4].

I.5 Choix des API

Les performances prises en considération lors de choix d'un API sont :

- Le type des entrées/sorties nécessaire.
- Le nombre d'entrées/sorties nécessaire.
- La qualité du service après-vente.
- Les capacités de traitement du processeur (vitesse, données, opérations, temps réel...).
- Les compétences/expériences de l'équipe d'automaticiens en mise en œuvre et en programmation de la gamme d'automate.
- Détermination de l'emplacement de l'A.P.I [5].

I.6 Architecture des automates programmables industriels

Les automates sont organisés suivant l'architecture suivante : Un module d'unité centrale ou CPU, qui assure le traitement de l'information et la gestion de l'ensemble des unités. Ce module comporte un microprocesseur, des circuits périphériques de gestion des entrées/sorties, des mémoires RAM : mémoire vive lecture écriture et des EEPROM (mémoire effaçable électriquement nécessaire pour stocker les programmes, les données et les paramètres de configuration du système).

CHAPITRE I : Automate programmable industriel (API)

Un module d'alimentation qui à partir d'une tension 220V/50Hz ou dans certains cas de 24V fournit les tensions continues +/-5V, +/-12 V.

Un ou plusieurs modules de sorties 'Tout ou Rien'(TOR) (vrai ou faux/1 ou 0), ou analogiques (l'information traitée est continue pour l'acquisition des informations provenant de la partie opérative).

Un ou plusieurs modules de sorties 'Tout ou Rien'(TOR), ou analogiques pour transmettre à la partie opérative les signaux de commande. Il y a des modules qui intègrent en même temps des entrées et des sorties [6]. La figure I.3 présente une photo d'un automate SIEMENS avec ces modules associés (Figure. I.2).



Figure. I.2. Automate programmable siemens avec modules associés.

I.6.1 Structure générale des API

Les composants principaux d'un automate programmable industriel (API) sont :

- Coffret, rack, baie ou cartes
- Compact ou modulaire
- Tension d'alimentation
- Taille mémoire
- Sauvegarde (EPROM, EEPROM, pile, ...)
- Nombre d'entrées / sorties
- Modules complémentaires (analogique, communication...)
- Langage de programmation

I.6.2 Architecture interne d'un API

En général, un API est constituée de composants fonctionnels de base suivante [8] (Figure. I.3).

- Une unité de traitement,
- La mémoire,

CHAPITRE I : Automate programmable industriel (API)

- Une unité d'alimentation,
- Des interfaces d'entrées-sorties,
- Une interface de communication

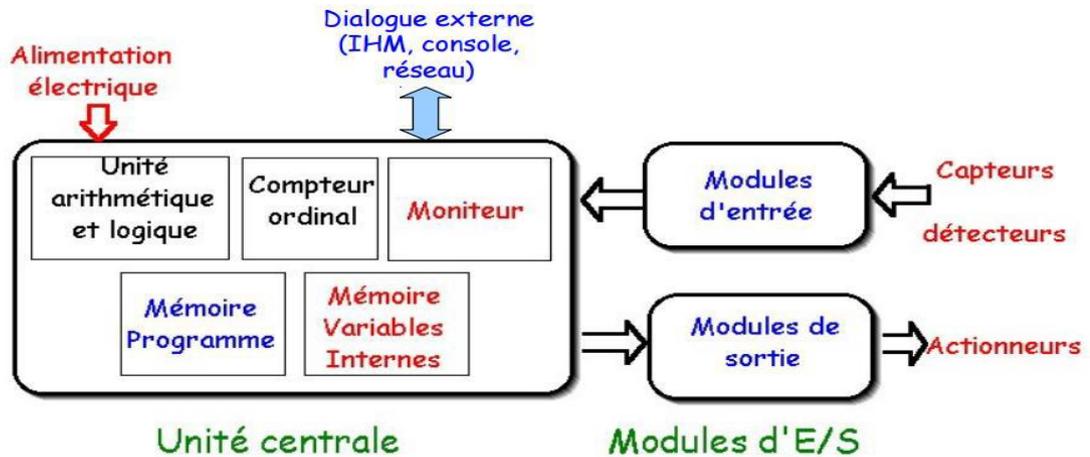


Figure I.3. Structure interne d'un Automate Programmable Industriel [9]

La structure interne d'un automate programmable industriel (API) est assez voisine de celle d'un système informatique simple, l'unité centrale est le regroupement du processeur et de la mémoire centrale. Elle commande l'interprétation et l'exécution des instructions programme.

I.6.3 Description des éléments d'un API

I.6.3.1 La mémoire

Elle est conçue pour recevoir, gérer, stocker des informations issues des différents secteurs du système, qui sont le terminal de programmation (PC ou console) et le processeur. Elle reçoit également des informations en provenance des capteurs (Figure I.4) [10].

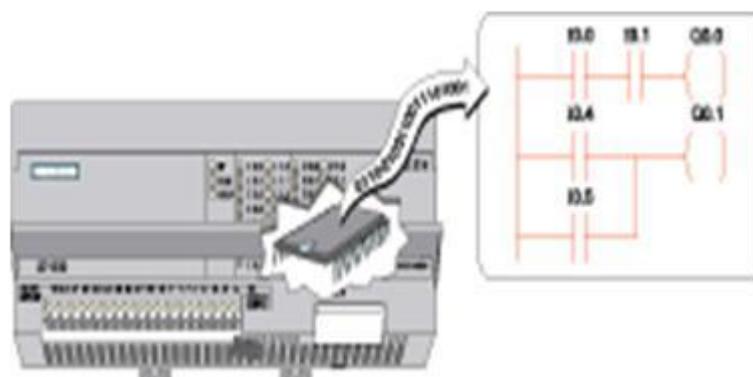


Figure I.4. La mémoire d'un API

CHAPITRE I : Automate programmable industriel (API)

Il existe dans les automates deux types de mémoires qui remplissent des fonctions différentes :

- La mémoire Langage où est stocké le langage de programmation. Elle est en général figée, c'est à dire en lecture seulement. (ROM : mémoire morte)

- La mémoire Travail utilisable en lecture-écriture pendant le fonctionnement c'est la RAM (mémoire vive). Elle s'efface automatiquement à l'arrêt de l'automate (nécessite une batterie de sauvegarde).

Réparation des zones mémoires :

- Table image des entrées.
- Table image des sorties.
- Mémoire des bits internes.
- Mémoire programme d'application.

I.6.3.2 Le processeur

Le processeur, appelé aussi unité de traitement ou unité arithmétique et logique, a double vocation d'assurer le contrôle de l'ensemble de la machine et effectuer les traitements demandés par l'instruction du programme. Il lit en permanence et à grande vitesse les états.

Logiques des signaux en provenance des capteurs périphériques en fonction du programme stocké dans la mémoire, et il transmet des ordres de sortie vers les actionneurs [11].

I.6.3.3 Les interfaces et les cartes d'Entrées / Sorties

L'interface d'entrée comporte des adresses d'entrée. L'interface de sortie comporte de la même façon des adresses de sortie. Le nombre de ces entrées et sorties varie suivant le type d'automate. Les cartes d'E/S ont une modularité de 8, 16 ou 32 voies. Les tensions disponibles sont normalisées (24, 48, 110 ou 230V continu ou alternatif ...) (Figure I.5). [12]

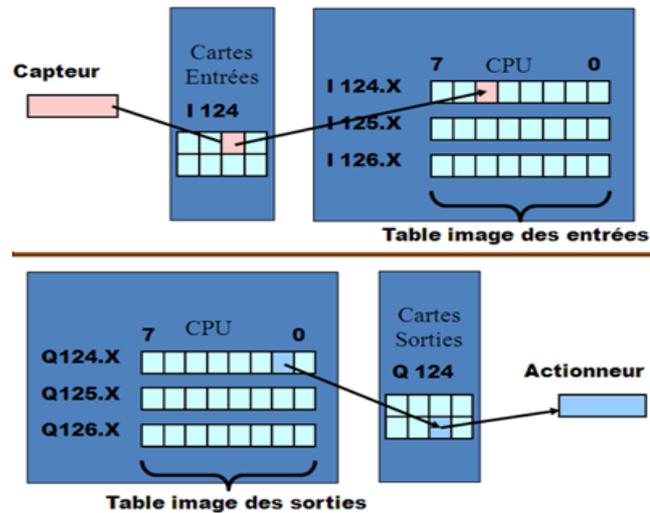


Figure I.5. Interfaces des E/S d'un API

a) Cartes d'entrées

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

b) Cartes de sorties

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

I.6.3.4 Une alimentation électrique

Elle a pour le rôle de transformer la tension du réseau en tension stable pour le bon fonctionnement de l'unité centrale, des modules d'entrées/sorties et des mémoires notamment face aux microcoupures de réseau électrique qui constitue de la source d'énergie, un onduleur est nécessaire pour éviter le risque de coupure non tolérées, la tension d'alimentation peuvent être de 5V, 12V ou 24V [13].

I.6.3.5 Liaisons de communication

L'interface de communication est utilisée pour recevoir et transmettre des données sur des réseaux de communication qui relient l'API à d'autres API distants (voir Figure I.4). Elle est impliquée dans des opérations telles que la vérification d'un périphérique, l'acquisition de

données, la synchronisation entre des applications et la gestion de la connexion (Figure I.6) [14].

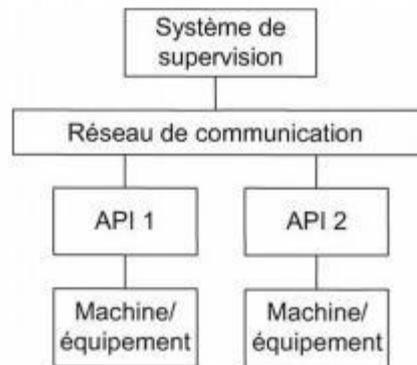


Figure I.6. Modèle de base des communications [14].

I.7 Nature des informations traitées par l'automate [15]

Les informations peuvent être de type :

- **Tout ou rien (T.O.R)** : l'information ne peut prendre que deux états (vrais/faux, 1 ou 0...). C'est le type d'information délivré 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 continue dans des mots codés sous forme binaire ou bien hexadécimale, C'est le type d'information délivrée par un ordinateur.

I.8 Langages de programmation pour API

Chaque automate possède son propre langage. Mais par contre, les constructeurs proposent tous une interface logicielle répondant à la norme CEI 1131-3. Cette norme définit cinq langages de programmation utilisables, qui sont : [16]

I.8.1 Le langage contact (Ladder Diagram)

C'est une représentation graphique d'équations booléennes combinant des contacts (en entrée) et des relais (en sortie). Il permet la manipulation de données booléennes, à l'aide de symboles graphiques organisés dans un diagramme comme les éléments d'un schéma électrique à contacts. Les diagrammes LD sont limités à gauche et à droite par des barres d'alimentation.

I.8.2 Le langage IL (Instruction List)

Un langage textuel de bas niveau. Il est particulièrement adapté aux applications de petite taille. Les instructions opèrent toujours sur un résultat courant (ou registre IL). L'opérateur indique le type d'opération à effectuer entre le résultat courant et l'opérande. Le résultat de l'opération est stocké à son tour dans le résultat courant.

Un programme IL est une liste d'instructions. Chaque instruction doit commencer par une nouvelle ligne, et doit contenir un opérateur, complété éventuellement par des modificateurs et, si c'est nécessaire pour l'opération, un ou plusieurs opérandes, séparés par des virgules (','). Une étiquette suivie de deux points (':') peut précéder l'instruction. Si un commentaire est attaché à l'instruction, il doit être le dernier élément de la ligne. Des lignes vides peuvent être insérées entre des instructions. Un commentaire peut être posé sur une ligne sans instruction [16].

I.8.3 Le langage FBD (Functions Block Diagram)

C'est un langage graphique. Il permet la construction d'équations complexes à partir des opérateurs standards, de fonctions ou de blocs fonctionnels.

Les principales fonctions sont :

- L'énoncé RETURN (peut apparaître comme une sortie du diagramme, si liaison connectée prend l'état booléen TRUE, la fin du diagramme n'est pas interprétée.
- Les étiquettes et les sauts conditionnels sont utilisés pour contrôler l'exécution du diagramme. Aucune connexion ne peut être réalisée à droite d'un symbole d'étiquette ou de saut [16].
- Saut à une étiquette (le nom de l'étiquette est « LAB »).

I.8.4 Le langage ST (Structured Text)

Un langage textuel de haut niveau dédié aux applications d'automatisation. Ce langage est principalement utilisé pour décrire les procédures complexes, difficilement modélisables avec les langages graphiques. C'est le langage par défaut pour la programmation des actions dans les étapes et des conditions associées aux transitions du langage SFC. Un programme ST est une suite d'énoncés. Chaque énoncé est terminé par un point-virgule (« ; »). Les noms utilisés dans le code source (identificateurs de variables, constantes, mots clés du langage...) sont délimités par des séparateurs passifs ou des séparateurs actifs, qui ont un rôle d'opérateur. Des commentaires peuvent être librement insérés dans la programmation [16].

I.8.5 Le langage SFC (Sequential Function Chart) ou GRAFCET

C'est un langage graphique utilisé pour décrire les opérations séquentielles.

Le procédé est représenté comme une suite connue d'étapes (états stables), reliées entre elles par des transitions, une condition booléenne est attachée à chaque transition. Les actions dans les étapes sont décrites avec les langages ST, IL, LD ou FBD.

Les principales règles graphiques sont :

- un programme SFC doit contenir au moins une étape initiale.
- une étape ne peut pas être suivie d'une autre étape.
- une transition ne peut pas être suivie d'une autre transition [16].

I.9 Présentation de l'automate S7-1200

L'automate SIMATIC S7-1200 fabriqué par SIEMENS est un automate de conception modulaire et compact, polyvalent, destiné à des tâches d'automatisation simple mais d'une précision extrême, il constitue donc, un investissement sûr et une solution parfaite à une grande variété d'applications.

Une conception modulaire et flexible, une interface de communication répondant aux exigences les plus sévères dans l'industrie et une large gamme de fonctions technologiques performantes et intégrées, font de cet automate, un composant à part entière d'une solution d'automatisation complète (Figure I.6) [17].



Figure I.7. L'automate programmable S7-1200[16]

I.9.1 Choix de la CPU

Les CPU du système SIMATIC S7-1200 se déclinent en trois classes de performances : CPU 1211 C, CPU1212 C et CPU1214 C, chacune d'elles pouvant être étendue en fonction des besoins de la station. Sur chaque CPU, il est possible de greffer une platine d'extension pour ajouter des E/S TOR ou analogiques supplémentaires sans modification de l'encombrement de l'automate. Des modules d'E/S supplémentaires peuvent être ajoutés du côté droit de la CPU pour étendre la capacité d'E/S TOR ou analogiques [17].

I.9.2 Le choix des modules d'Entrées/Sorties

Le choix des modules Entrées/ Sorties est basé sur les critères suivants :

- Le type et la valeur de la tension d'entrée ou de sortie.
- Le nombre de voies.
- Le type d'entrée ou de sortie (sonde, thermocouple, électrovanne...).

I.9.2.1 Les signaux TOR

Dans notre cas, on trouve que l'on a 18 entrées, donc le type d'entrée doit être TOR, avec une tension 24VDC. Pour un automate S7-1200, les modules entrées TOR 24VDC disponibles, on a le choix entre le modèle de 8 voies ou de 16 voies.

La CPU choisie contient 14 entrées TOR de 24VDC, et pour assurer 18 entrées TOR il faut un module de 8 voies. Pour prendre en compte la marge de sécurité, on ajoute alors un module de 16 voies pour satisfaire nos besoins avec une large marge de sécurité.

Pour les sorties, on a 26 sorties, la CPU choisie contient 10 sorties TOR de 24VDC, donc on a besoin d'un module de 16 voies pour satisfaire nos besoins

Donc, on choisit un seul module numérique de 16 entrées et de 16 sorties **6ES7223-1BL32-0XB0** (Figure I.8). [17]



Figure I.8. Module 16 DI / 16 DO

I.9.2.2 Les signaux analogiques

Dans notre système, on a que quatre (4) entrées analogiques, alors on choisit un module de quatre (4) entrées analogiques pour assurer la marge de sécurité. 6ES7 231-5ND320-0XB0

Module d'entrées analogiques AI4 x 16 bits ; borniers enfichables ; entrées : 1,25V, 2,5V, 5V, 10V et 0...20mA ; réjection des fréquences perturbatrices paramétrable ; lissage paramétrable ; diagnostic paramétrable (Figure I.9) [17].



Figure I.9. Module AI4

I.10 Conclusion

Ce chapitre présente des généralités sur l'automate programmable industriel (API), définitions, architecture interne, architecture externe et langages de programmations. Aussi S7-1200 brièvement décrite dans chapitre.

Chapitre II

PROFINET

II.1 Introduction

PROFINET est la première norme Industriel Ethernet intégrée pour l'automatisation et utilise les avantages d'Ethernet et de TCP/IP pour une communication ouverte du niveau de la gestion d'entreprise au processus lui-même [19].

Profinet est un standard de communication ouvert pour l'automatisation industrielle .Il a été créé par pi (profibus & profinet international)-l'organisation des utilisateurs profibus qui compte plus de 1200 membres-et développé par simens, Phoenix contact, moxer et d'autres constructeur. De par son ouverture et l'utilisation d'un media de communication standard (Ethernet), profinet permet l'utilisation de toute marque de matériel [20].

PROFINET (« Process Field Network ») se répartit selon différentes classes de performance pour répondre à diverses contraintes temporelles : PROFINET RT pour le temps réel mou ou lorsque le temps réel n'est pas requis, et PROFINET IRT pour une performance temps réel dur. La technologie a été mise au point par Siemens et les sociétés membres de l'organisation d'utilisateurs PROFIBUS, PNO. Successeur de PROFIBUS DP basé sur Ethernet, PROFINET I/O spécifie tous les transferts de données entre les contrôleurs d'E/S ainsi que le paramétrage, le diagnostic et la topologie du réseau [21].

II.2. Objectifs et avantages de PROFINET

La norme PROFINET a les objectifs suivants [22]:

- Standard Ethernet ouvert, conçu pour l'automatisation sur la base de Industrial Ethernet :

Les composants Industrial Ethernet et Ethernet standard sont compatibles entre eux, toutefois les appareils Industrial Ethernet sont plus robustes et donc mieux adaptés à un environnement industriel (température, immunité aux perturbations, etc.).

- Utilisation de TCP/IP et de standards des NTIC
- Automatisation avec Ethernet temps réel
- Intégration sans faille de systèmes de bus de terrain

PROFINET précise les fonctions servant à réaliser une solution d'automatisation globale, de l'installation du réseau au diagnostic sur le Web. Grâce à sa structure modulaire, il sera très facile de lui ajouter des fonctions à l'avenir.

Il en résulte les avantages suivants :

- Flexibilité grâce à l'exploitation d'Ethernet et des standards éprouvés des NTIC
- Economies au niveau de l'ingénierie et de la mise en service grâce à la modularisation
- Rentabilisation des investissements en matériel et applications PROFIBUS

- Plus rapide que les bus spécifiques actuels dans le domaine du Motion Control
- Vaste gamme de produits disponibles sur le marché

II.3. Simplicité de câblage

PROFINET mise sur la technologie commutée 100 Mbit/s et supporte non seulement la topologie en étoile d'Ethernet, mais aussi les topologies en bus et en anneau. Ceci réduit les coûts de câblage et garantit un maximum de flexibilité.

La communication sans fil avec IWLAN ouvre la porte à de nouvelles applications dans l'industrie – et permet même de réaliser des applications de conduite et de supervision [23].

II.3.1 Sécurité intégrée

Sous l'angle de la sécurité totale pour l'homme, la machine et l'environnement, PROFINET présente également toutes les qualités requises. L'utilisation de PROFIsafe permet de réaliser un réseau pour la communication standard et de sécurité – sur un seul et même câble ou sans fil avec Industriel Wireless LAN (IWLAN) [23].

II.3.2 Standards TIC & accès sécurisé

PROFINET offre toutes les fonctionnalités pour une configuration et un diagnostic optimal. L'accès à l'ensemble des données concernées est possible depuis n'importe où dans le monde entier via Internet.

PROFINET satisfait également aux exigences accrues en matière de sécurité des données et des réseaux [23].

II.3.3 Processus

PROFINET est conçu comme un standard complet pour toutes les tâches d'automatisation.

Les réponses à des exigences spéciales pour une utilisation dans des installations du génie des procédés et pour l'interconnexion d'appareils d'instrumentation de processus et d'analyse sont actuellement en cours d'élaboration dans le cadre de la standardisation. Et dans le secteur des processus secondaires, PROFINET compte déjà de nombreuses réalisations pratiques à son actif [23].

II.3.4 Appareils de terrain décentralisés

PROFINET permet de raccorder directement à Industrial Ethernet des appareils de terrain décentralisés instaurant ainsi un échange de données rapide entre la périphérie et le contrôleur, avec en corollaire une amélioration notable des possibilités de diagnostic.

II.3.5 Motion Control

Sur la base profinet, il est également possible de réaliser aisément ,avec ISO-chrono real-time(IRT) ,des régulations isochrones très rapide pour des application de contrôle de mouvement haute performance [23].

II.3.6 Intelligence répartie

Pour les structures d'installations modulaires, profibus international a défini le standard profinet CBA (component based automation).

Le secteur de la construction de machines et d'équipements dispose déjà d'une bonne expérience en matière de modularisation : les éléments fréquemment utilisés sont préfabriqués et, après passation de commande, rapidement assembles pour former unité personnalisée [23].

II.3.7 Communication Profinet :

La communication Profinet sur Ethernet s'échelonne sur trois niveaux de performance [24]

- le non-temps réel, par exemple communication TCP/IP et UDP/IP
- le temps réel (RT) et
- le temps réel isochrone (IRT)

Ces trois échelons couvrent toutes les applications d'automatismes. Parmi ses caractéristiques clés, citons:

- La coexistence de transmissions temps réel et TCP/IP sur une seule ligne ;
- Un protocole temps réel standardisé pour toutes les applications et la communication aussi bien entre composants intelligents décentralisés qu'entre contrôleur et périphérie décentralisée.
- Une communication temps réel évolutive, de performante à ultra-performante, avec synchronisation d'horloge.

Ces caractéristiques constituent l'un des points forts de PROFINet ; elles en garantissent la cohésion à tous les niveaux de l'entreprise, de l'atelier à la direction, et une grande réactivité au sein du processus.

La communication TCP/IP et UDP/IP :

La transmission de données à temps non critique avec TCP/IP et UDP/IP est la base technique de la communication – par exemple pour le paramétrage et la configuration. TCP/IP est le standard de fait dans le paysage informatique.

Communication temps réel :

Profinet est basé sur industriel Ethernet le standard TCP/IP (transport control Protocol / internet Protocol) pour le paramétrage, la configuration et le diagnostic. La communication en temps réel pour la transmission des données utiles de process s'effectuer sur le même câble. Les appareils profinet supportent [24]. Les caractéristique temps réel suivant :

➤ **Real -time (RT) :**

Le temps réel s'utilise pour les données de processus critiques – c'est-à-dire les données utiles cycliques ou les alarmes sur événements. Pour répondre aux exigences de temps réel dans le secteur de l'automatisation, PROFINET utilise un canal de communication temps réel optimisé.

Cette solution minimise les temps d'exécution et se traduit par une augmentation des performances lors de l'actualisation de données de process.

La performance dépasse celle des bus de terrain classiques et autorise des temps de réaction de l'ordre de la microseconde.

Dans le même temps, la puissance de calcul du processeur mobilisée par la communication est nettement allégée, et l'utilisation de constituants de réseau standard est parfaitement possible.

➤ **NRT-Channel:**

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

➤ **Real-Time Channel RT:**

- transfert haute performance

- données cycliques
- signaux évènements
- utilise des contrôleurs Ethernet standard
- **Real-Time Channel IRT:**
 - transfert haute performance avec synchronisation d'horloge
 - jitter < 1usec
 - Hardware spécifique nécessaire

II.4 Installation du réseau [23]

La normalisation internationale ISO/CEI 11801 et son équivalente européenne EN 50173 définissent un réseau informatique standard indépendant de l'application pour le secteur de la bureautique.

Mais comme les exigences imposées à un réseau en environnement industriel sont plus élevées que dans le secteur de la bureautique, le guide PROFINET « Cabling and Interconnection Technology », décrit l'infrastructure passive dans et entre les cellules d'automatisme pour Fast Ethernet avec PROFINET :

- Connecteurs et câbles pour réseaux cuivre et optiques
- Connecteurs pour alimentation 24 V
- Connecteurs pour alimentation 400 V

Il décrit en outre les conditions de test et de contrôle correspondantes pour les différents composants.

La description de tous les composants passifs nécessaires pour un réseau PROFINET permet de garantir une interaction parfaite entre les différents composants passifs

II.4.1 Commutateurs (switches) [23]

PROFINet utilise toujours des commutateurs implantés tout au long de la transmission entre stations pour régénérer et aiguiller les signaux. Ces équipements, normalisés ISO/CEI 15802-3, servent à structurer le réseau.

Les commutateurs adaptés à Profinet sont ceux conçus pour Fast Ethernet (100 Mbit/s, IEEE 802.3u) et la transmission en full duplex ; dans ce mode, le commutateur reçoit et émet simultanément sur un même port, sans risque de collision et, donc, sans perte de bande passante due aux mécanismes de détection Ethernet. La configuration du réseau en est grandement

simplifiée puisqu'il n'y a pas de contrôle des longueurs de segment dans un domaine de collision.

II.4.2 Topologies des réseaux [23]

Les topologies visent à satisfaire les exigences des unités à fédérer sur le réseau. Les plus utilisées sont l'étoile, le bus, l'arborescence et l'anneau. En pratique, un réseau tend à panacher ces structures qui empruntent des supports physiques comme le câble en cuivre ou la fibre optique.

La fibre optique en verre est utilisée pour la transmission sur de longues distances. Pour les courtes distances, la FO plastique, par exemple POF (Polymer Optique Fiber) ou la FO verre enrobée de plastique, par exemple PCF (Polymer Cladded Fiber) constituent une alternative simple et aisément réalisable par l'utilisateur.

Pour PROFINET, on dispose de constituants de réseau actifs tels que les appareils de terrain décentralisés avec interfaces POF/ PCF intégrées. Afin de garantir une disponibilité élevée, les câbles sont surveillés lors de la mise en service et en cours d'exploitation afin de détecter d'éventuelles atténuations du signal consécutives au vieillissement du matériau.

II.4.2.1 Étoile

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 [25]. (voir Figure II.1).

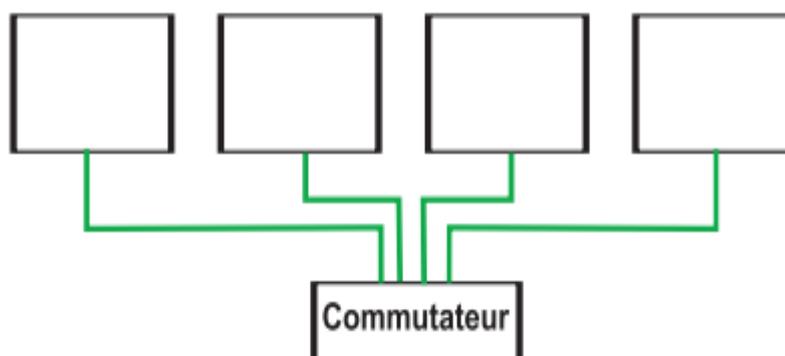


Figure II.1 : Topologie des réseaux en étoile

II.4.2.2 Arbre

Cette topologie consiste à réunir plusieurs étoiles pour former un réseau combinant au besoin fibre optique et paire torsadée. Elle permet de subdiviser des installations complexes en sous-réseaux [25].

II.4.2.3 Ligne

La structure linéaire fait appel à un commutateur situé à proximité du terminal de connexion ou intégré au terminal. Elle est surtout adaptée aux installations de grande envergure – par exemple la manutention et la liaison de cellules de fabrication [25]. (voir Figure II.2).

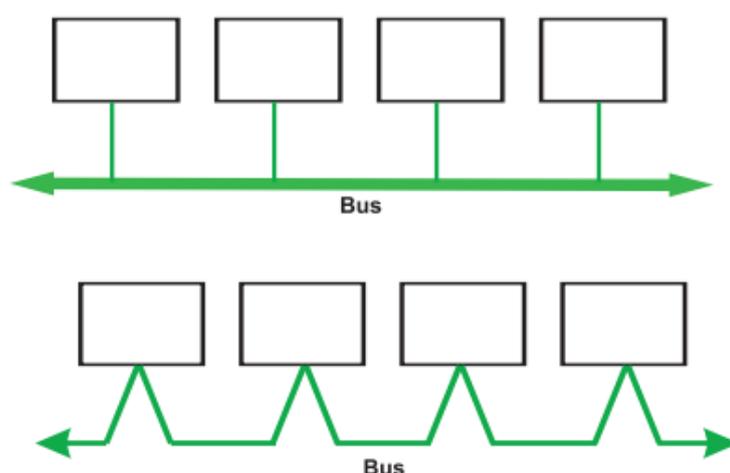


Figure II.2 : Topologie des réseaux en bus

II.4.2.4 Anneau (redondant)

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 (voir Figure II.3). [25].

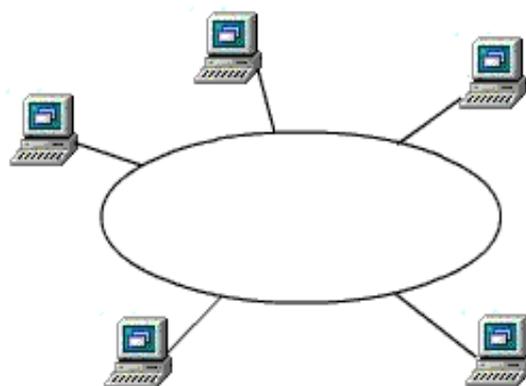


Figure II.12 : Topologie des réseaux en anneau

II.4.3 Profinet et modèle OSI [26]

Le profinet respecte l'organisation en 7 couches du modèle OSI.

Les couche 5 et 6 étant inutiles ne sont donc pas utilisées dépendent de la nature du trafic :

1 - le trafic IT classique sans rapport avec profinet, par ex. accéder a la page web de l'équipement.

2 - le trafic profinet non temps réel, par ex .configurer une station ou faire un diagnostic.

3 - le trafic profinet temps réel qui concerne les données d'entrées/sorties du procédé.

Pour les 2 premiers on utilisera Ethernet, la couche reseau IP et la couche transport TCP/UDP. Pour les échanges de données de process, on passe directement d'Ethernet (couche 1&2) à la couche applicative Profinet.

Les performances temporelles ne sont donc pas dégradées par le traitement d'informations inutiles dans le cadre d'une communication industrielle locale.

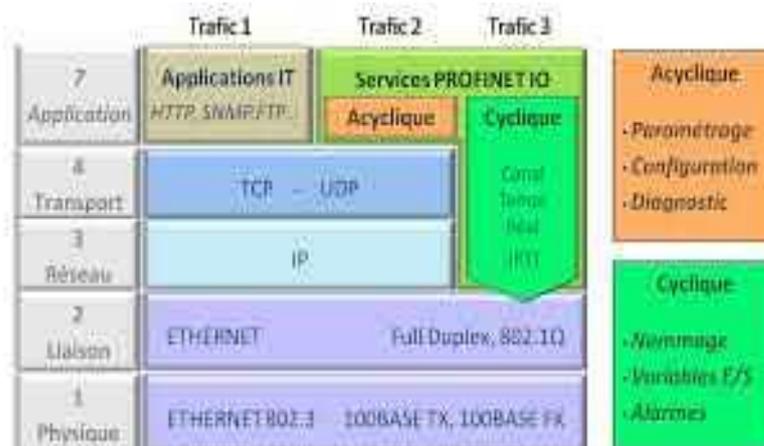


Figure II.4 : profinet dans le modèle OSI

Caractéristique de la couche physique

Le reseau profinet étant base sur Ethernet il respecte les standards correspondants.

Les stations supportent des connexions cuivre, fibre optique ou wifi [26].

- 1- La version cuivre, la plus répandue ,est basée sur la norme 802.3u (fast Ethernet) qui permet une communication 100mb/sen full duplex. essentiel des équipements dispose d'une connexion de ce type
- 2- La fibre optique est essentiellement utilisée dans le cadre de l'infrastructure et sert à interconnecter les switches entre eux.
- 3- Le wifi est utilisé ponctuellement pour des applications mobiles et/ou difficiles à raccorder physiquement.

II.5. Type profinet

II.5.1 PROFINET CBA

PROFINET CBA (Component Based Automation) est un concept d'automatisation renforçant les points suivants :

- Réalisation d'applications modulaires
- communication machine-machine

Avec PROFINET CBA vous réalisez un projet d'automatisation sur la base de composants et de sous-systèmes prédéfinis. Ce concept répond aux demandes de modularité sans cesse.

Croissante provenant du monde de la construction mécanique et d'équipements, visant à Une décentralisation poussée et à un usinage "intelligent".

Component Based Automation vous permet de mettre en œuvre des modules technologiques complets sous forme de composants standardisés utilisés dans de grandes Installations.

Les composants intelligents modulaires PROFINET CBA sont créés au moyen d'un outil d'ingénierie qui peut varier d'un constructeur d'appareils à l'autre.

Les composants construits à partir d'appareils SIMATIC, sont créés avec STEP 7 et connectés avec l'outil SIMATIC iMAP [27].

II.5.2. PROFINET IO

PROFINET IO, avec sa communication en temps réel particulièrement rapide, répond à toutes les exigences actuelles en matière de transmission de données de processus et permet une intégration facile des systèmes de bus de terrain existants.

II.5.2.1 Le fonctionnement de Profinet IO

Avec Profinet IO, la spécification Profinet propose, pour les appareils de terrain simples, un modèle d'intégration qui recoupe peu ou prou la notion habituelle de périphérie décentralisée. Les composants sont adressés sous la forme habituelle, comme dans l'environnement du bus de terrain. Par la description d'appareil ils sont intégrés dans l'outil d'ingénierie correspondant (PC Worx, Step 7, etc), où ils sont affectés à un contrôleur à l'étape de la configuration [27].

Dans le cadre de Profinet IO, on distingue les types d'appareil suivants :

II.5.2.1.1. Contrôleur d'E/S

Le contrôleur d'E/S désigne l'équipement par l'intermédiaire duquel sont adressés les périphériques d'E/S raccordés. Il s'agit généralement d'un contrôleur séquentiel au sein d'une

unité fonctionnelle décentralisée, qui échange des signaux d'entrée et de sortie avec les appareils de terrain associés [27].

II.5.2.1.2. Périphérique d'E/S

Par périphérique d'E/S on entend un appareil de terrain décentralisé (ex. un module d'E/S distant, un moteur, un îlot d'électrovannes, un switch), affecté à un ou plusieurs contrôleurs d'E/S et chargé de transmettre les données de process et de configuration, ainsi que les alarmes. Les échanges de données entre les périphériques d'E/S sont organisés par la configuration de sous-modules du type producteur/consommateur [27].

II.5.2.1.3 Superviseur d'E/S

Le superviseur d'E/S, une console de programmation ou un PC industriel, accède à toutes les données de process et de paramétrage en même temps que le contrôleur d'E/S.

Entre le contrôleur d'E/S, le superviseur d'E/S et les périphériques d'E/S, il existe des relations d'application qui comportent plusieurs relations de communication pour transmettre des données de configuration (canal standard), des données de process (canal temps réel), et des alarmes (canal temps réel) [27].

II.5.3 Attribution d'adresse

Adresses

Tous les appareils PROFINET étant conçus pour le protocole TCP/IP, leur fonctionnement sur réseau Ethernet nécessite une adresse IP [22].

Pour simplifier la configuration, il ne vous sera demandé qu'une seule fois d'attribuer une adresse IP, notamment lors de la configuration de l'IO-Controller sous STEP 7/HW Config [22].

STEP 7 affiche ici un dialogue de sélection de l'adresse IP et du réseau Ethernet. Si le réseau est isolé, vous pouvez reprendre l'adresse IP et le masque de sous-réseau proposés par STEP7. Si le réseau fait partie d'un réseau d'entreprise Ethernet, adressez-vous à l'administrateur de réseau pour obtenir ces adresses [22].

Les adresses IP des IO-Devcies sont générées par STEP 7 et ne seront affectées aux IO-Devices qu'au démarrage de la CPU. Les adresses IP des IO-Devices possèdent toujours le même masque de sous-réseau que l'IO-Controller et sont attribuées dans l'ordre croissant à partir de l'adresse IP de l'IO-Controller. Cette adresse IP peut être modifiée manuellement en cas de besoin.

Adresse IP et MAC

Adresse MAC

Une identification d'appareil, unique au niveau mondial, est attribuée en usine à chaque appareil PROFINET. Cette identification d'appareil de 6 octets est l'adresse MAC [22].

L'adresse MAC est composée de la manière suivante :

- 3 octets pour le code constructeur et
- 3 octets pour le code appareil (numéro d'ordre).

L'adresse MAC se trouve généralement sur la face avant de l'appareil : 08-00-06-6B-80-C0
p. ex.

Adresse IP

Pour qu'un appareil PROFINET puisse être adressé comme participant à Industriel Ethernet, il doit posséder en plus une adresse IP unique sur le réseau. L'adresse IP se compose de 4 nombres décimaux compris entre 0 et 255. Ces nombres décimaux sont séparés par un point [22].

L'adresse IP est constituée des éléments suivants :

- adresse du réseau et
- adresse du partenaire (généralement aussi appelé hôte ou noeud de réseau).

Masque de sous-réseau

Les bits à 1 du masque de sous-réseau définissent la partie de l'adresse IP qui contient l'adresse du réseau.

Règle générale :

- L'adresse de réseau résulte de la combinaison ET de l'adresse IP et du masque de sous-réseau
- L'adresse de partenaire résulte de la combinaison NON ET de l'adresse IP et du masque de sous-réseau.

II.6 Conclusion

Dans ce chapitre on a présenté les objectifs et les avantages de PROFINET ainsi que les topologies et les techniques de communication utilisées dans ce type de réseau.

Chapitre III

Logiciel et programmation

III.1 Introduction

En réponse à la pression internationale croissante de la concurrence, il est aujourd'hui plus que jamais important d'exploiter à fond tous les potentiels d'optimisation sur l'ensemble du cycle de vie d'une machine ou d'une installation.

Totally Integrated Automation apporte une réponse optimale à toutes les exigences et offre un concept ouvert vis à vis des normes internationales et de systèmes tiers.

Dans ce chapitre, un réseau industriel est simulé et réalisé à l'aide de deux automates programmables, un switch et une station PC dont TIA Portal est installé comme outil de configuration, de programmation et de commande.

III.2 TIA Portal V13

Pour la programmation, on a utilisé le logiciel de Siemens STEP 7 Professional V13 (TIA Portal V13).

Le portail Totally Integrated Automation, ci-après appelé Portal TIA, offre la fonctionnalité complète pour réaliser notre tâche d'automatisation, regroupée dans une plateforme logiciel globale. Le portail TIA permet également de disposer, au sein d'un cadre, d'un environnement de travail commun pour une ingénierie transparente avec différents systèmes SIMATIC. Tous les logiciels requis, de la configuration matérielle à la visualisation du processus en passant par la programmation, sont intégrés dans un cadre complet d'ingénierie.

Le logiciel STEP 7 Professional (TIA Portal V13) est l'outil de programmation des nouveaux automates comme :

- ✓ SIMATIC S7-1500
- ✓ SIMATIC S7-1200
- ✓ SIMATIC S7-300
- ✓ SIMATIC S7-400

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

- Configuration et paramétrage du matériel
- Paramétrage de la communication
- Programmation
- Test, mise en service et dépannage avec les fonctions d'exploitation et de diagnostic

- Documentation
- Génération d'écrans de visualisation pour les Basic Panels SIMATIC avec WinCC Basic intégré.
- Il est également possible de générer des écrans de visualisation pour les PC et autres Panels à l'aide de WinCC Real Time (WinCC RT).

III.2.1 Les avantages du logiciel TIA portal

- ✓ Programmation intuitive et rapide : avec des éditeurs de programmation nouvellement développés SCL, CONT, LOG, LIST et GRAPH.
- ✓ Efficacité accrue grâce aux innovations linguistiques de STEP 7 : programmation symbolique uniforme, Calculate Box, ajout de blocs durant le fonctionnement, et bien plus encore
- ✓ Performance augmentée grâce à des fonctions intégrées : simulation avec PLCSIM, télémaintenance avec TeleService et diagnostic système cohérent.
- ✓ Technologie flexible : Fonctionnalité motion control évolutive et efficace pour les automates S7-1500 et S7-1200.
- ✓ Sécurité accrue avec Security Integrated : Protection du savoir-faire, protection contre la copie, protection d'accès et protection contre la falsification.
- ✓ Environnement de configuration commun avec pupitres IHM et entraînements dans l'environnement d'ingénierie TIA Portal.

III.3 Création d'un nouveau projet SUR TIA PORTAL

Dans le but de créer un nouveau projet sur TIA PORTAL, nous devons suivre les étapes suivantes :

III.3.1 Vue du TIA Portal

La vue du portail fournit une vue d'ensemble du projet et un accès aux outils qui permettent de l'élaborer. Vous pouvez trouver rapidement ce que vous souhaitez faire, et appeler l'outil qui servira à accomplir la tâche voulue. Si vous le souhaitez, un changement vers la vue du projet s'effectue automatiquement pour la tâche sélectionnée. Cette vue simplifie donc principalement la préparation et la mise en place du projet (Figure III. 1).

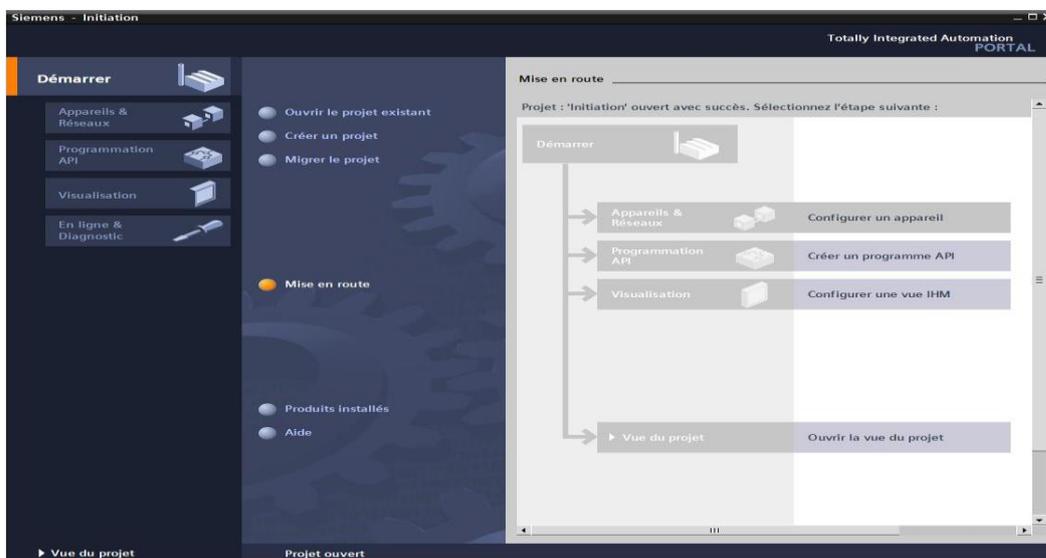


Figure III.1 : Vue du TIA Portal

III.3.2 Créer un projet

Les programmes pour SIMATIC S7-1200 sont gérés sous forme de projets. Nous allons maintenant créer un nouveau projet via la vue portail (« **Créer un projet** > Nom : *Initiation* > **Créer** ») (figure III.2)

Chapitre III : Logiciel et programmation

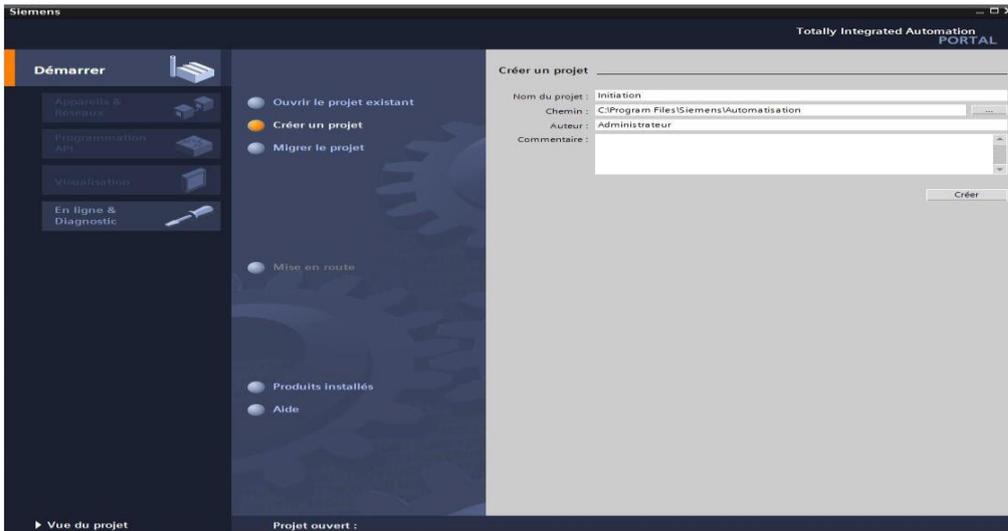


Figure III.2 : Créer un projet

III.3.3 Configurer un appareil

« Mise en route » est recommandée pour le début de la création du projet.

Premièrement, nous voulons « Configurer un appareil » (« Mise en route > Configurer un appareil »).

A partir du navigateur de projet, double-cliquez sur ajouter un appareil et sélectionnez l'automate S7 de votre choix ((Dans notre cas nous avons choisi la CPU 1214c AC/DC/RLY).

Cliquez sur le bouton "ok" pour ajouter l'automate (Figure III. 3).

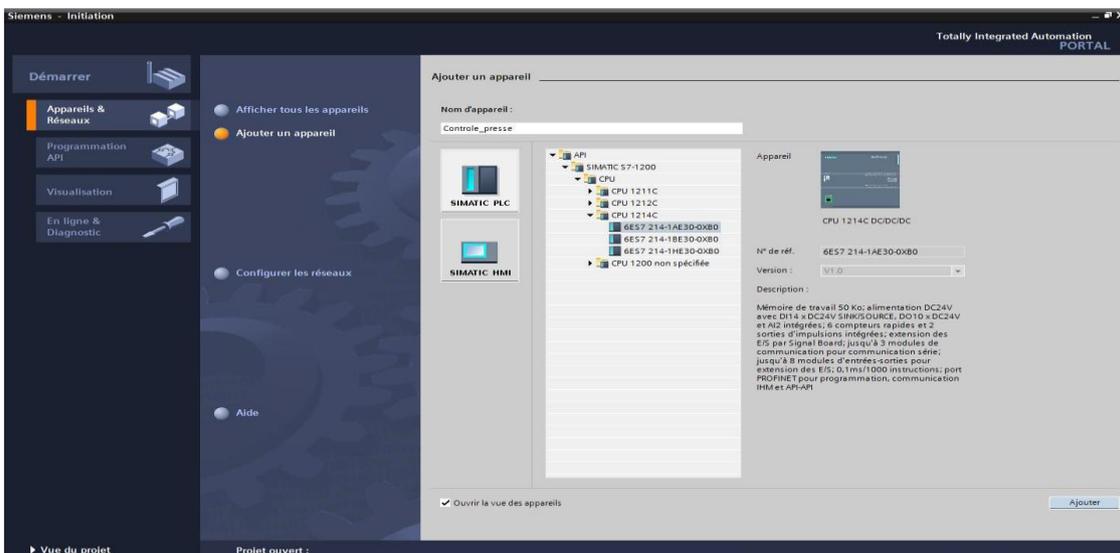


Figure III.3 : Configurer un appareil

III.3.4 Vue du projet

L'élément « Projet » contient l'ensemble des éléments et des données nécessaires pour mettre en œuvre la solution d'automatisation souhaitée (figure III.4).

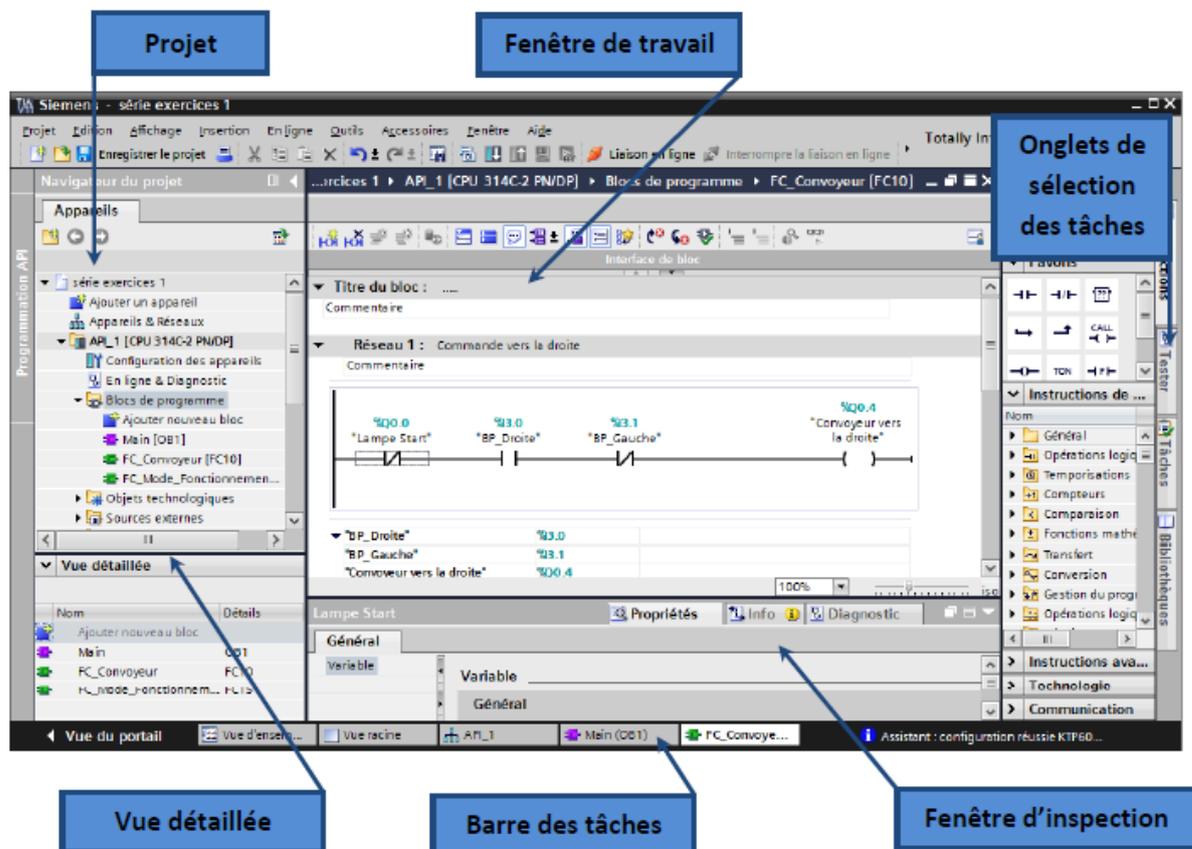


Figure III.4: Vue du projet

- ✓ **La fenêtre de travail :** permet de visualiser les objets sélectionnés dans le projet pour être traités. Il peut s'agir des composants matériels, des blocs de programme, des tables des variables, des HMI,...
- ✓ **La fenêtre d'inspection :** permet de visualiser des informations complémentaires sur un objet sélectionné ou sur les actions en cours d'exécution (propriété du matériel sélectionné, messages d'erreurs lors de la compilation des blocs de programme,...).

- ✓ **Les onglets de sélection de tâches** : ont un contenu qui varie en fonction de l'objet sélectionné (configuration matérielle, bibliothèques des composants, bloc de programme, instructions de programmation).

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

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

III.3.5 Configuration matérielle

La configuration matérielle est une étape importante. Elle consiste à disposer les châssis (rack), les modules et les appareils de la périphérie centralisée.

Dans notre cas, nous avons choisis la CPU 1214c AC/DC/RLY , un module d'entré/sortie TOR (DI 8×24VDC : 6ES72218_1BF32-0XB0) pour la configuration de notre matérielle.

III.3.5.1 Adressage des E/S

Pour connaître l'adressage des entrées et sorties présentes dans la configuration matériel, il faut aller dans « Appareil et réseau » dans le navigateur du projet. Dans la fenêtre de travail, on doit s'assurer d'être dans l'onglet « Vue des appareils » et de sélectionner l'appareil voulu. On sélectionne la CPU puis à l'aide des deux petites flèches, on fait apparaître l'onglet « **Vue d'ensembles des appareils** ». Les adresses des entrées et sorties apparaissent. On peut les modifier en entrant une nouvelle valeur dans la case correspondante (figure III.5).

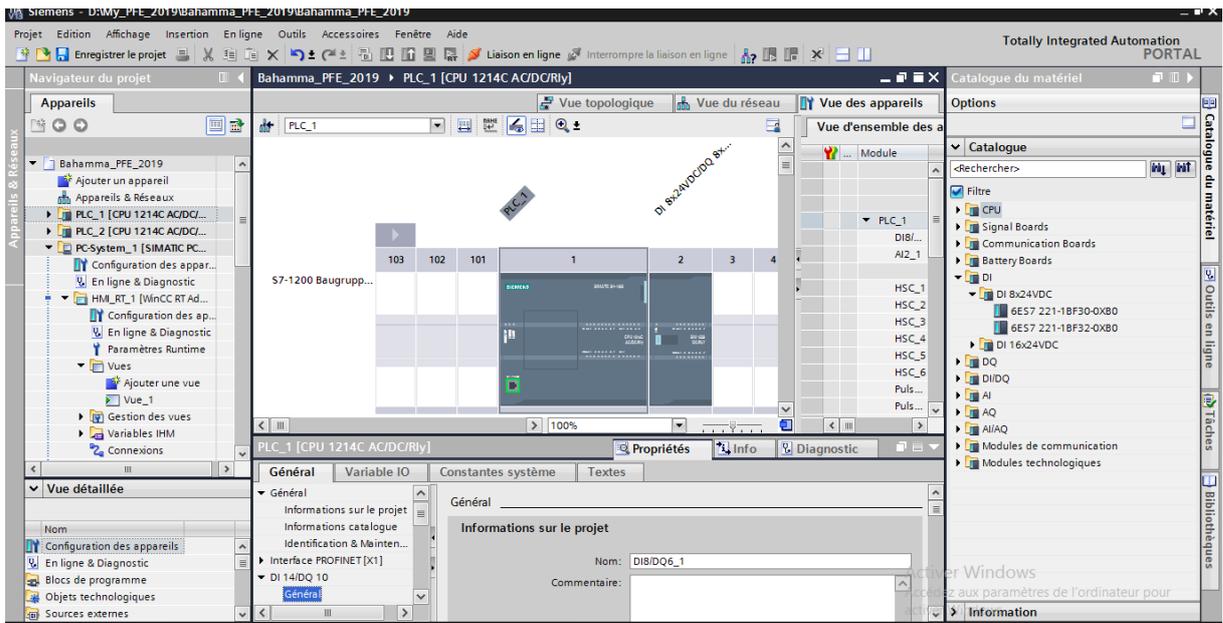


Figure III.5 : Adressage des E/S

III.3.5.2 Les variables API

III.3.5.2.1 Adresses symbolique et absolue

Dans TIA portal, toutes les variables globales (entrées, sorties, mémentos,...) possèdent une adresse symbolique et une adresse absolue.

- ✓ **L'adresse absolue** représente l'identificateur d'opérande (I, Q, M,...) et son adresse et numéro de bit.
- ✓ **Adresse symbolique** correspond au nom que l'utilisateur a donné à la variable (ex bouton marche).

Le lien entre les adresses symbolique et absolue se fait dans la table des variables API.

Lors de la programmation, on peut choisir d'afficher les adresses absolues, symboliques ou encore les deux simultanément.

III. 3.5.3 Table des variables API

C'est dans la table des variables API que l'on va pouvoir déclarer toutes les variables et les constantes utilisées dans le programme. Lorsque l'on définit une variable API, il faut définir :

- ✓ Un nom : c'est l'adressage symbolique de la variable.

Chapitre III : Logiciel et programmation

- ✓ Le type de donnée : BOOL, INT,...
- ✓ L'adresse absolue : par exemple Q 1.5

On peut également insérer un commentaire qui nous renseigne sur cette variable. Le commentaire peut être visible dans chaque réseau utilisant cette variable

III.3.6 Adresse Ethernet de la CPU

Toujours dans les propriétés de la CPU, il est possible de définir son adresse Ethernet. Un double clic sur l'icône Ethernet de la station fait apparaître la fenêtre d'inspection permettant de définir ses propriétés.

Pour établir une liaison entre la CPU et la console de programmation, il faut affecter aux deux appareils des adresses appartenant au même sous réseau. L'adresse utilisée est **192.168.0.2** de l'automate (figure III.6).

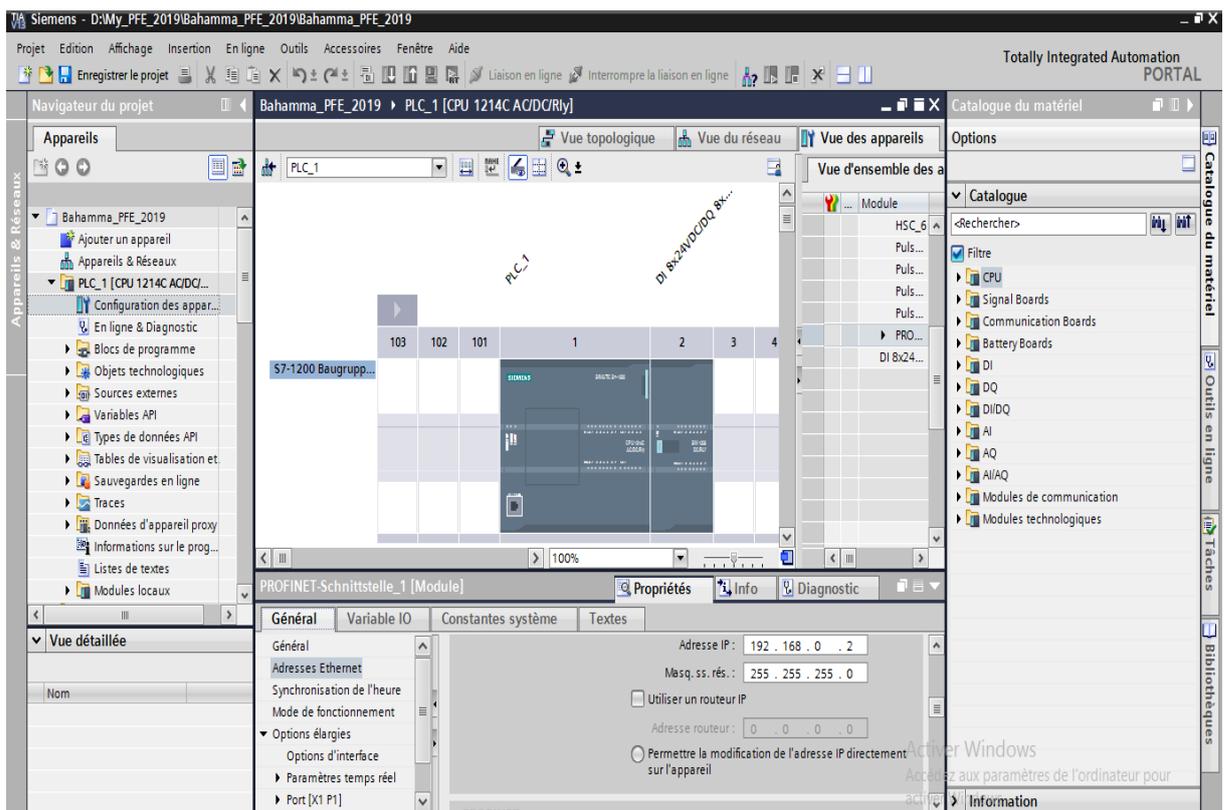


Figure III.6 : Adresse Ethernet de la CPU

III.4 Configure-t-on, dans WinCC (TIA Portal), une liaison entre un pupiter IHM ou un Wincc RunTime Advanced

Double-cliquez sur (ajouter un appareil) et sélectionnez wincc runtime Advanced (figure III.7).

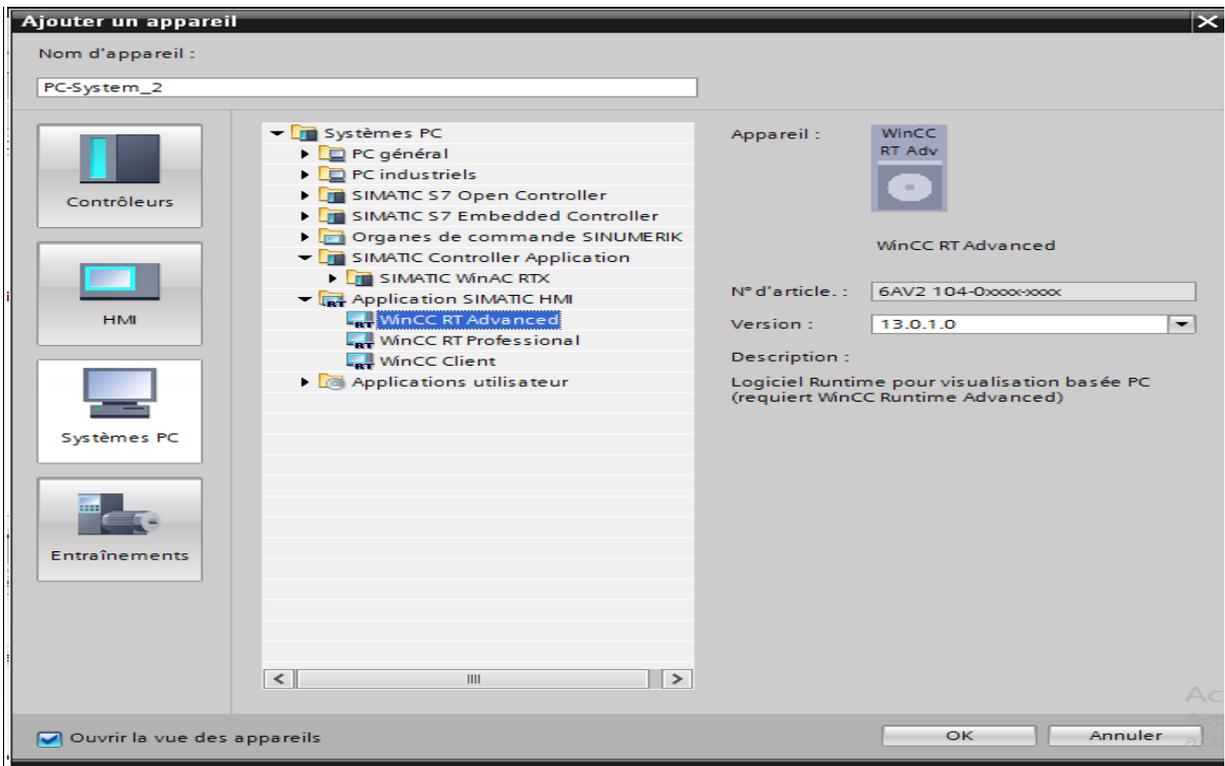


Figure III.7 : ajouter un appareil

Dans la (vue des appareils), sélectionnez à partir du (catalogue matériel), un module de communication générale (IE GENERAL) et double-cliquez dessus pour l'ajouter à votre station pc.

Allez dans les propriétés la carte (IE GENERAL) que vous venez d'ajouter .dans la partie (général) interface profinet adresses Ethernet).

Saisissez l'adresse IP et le masque de sous-réseau de la station runtime (figure III.8).

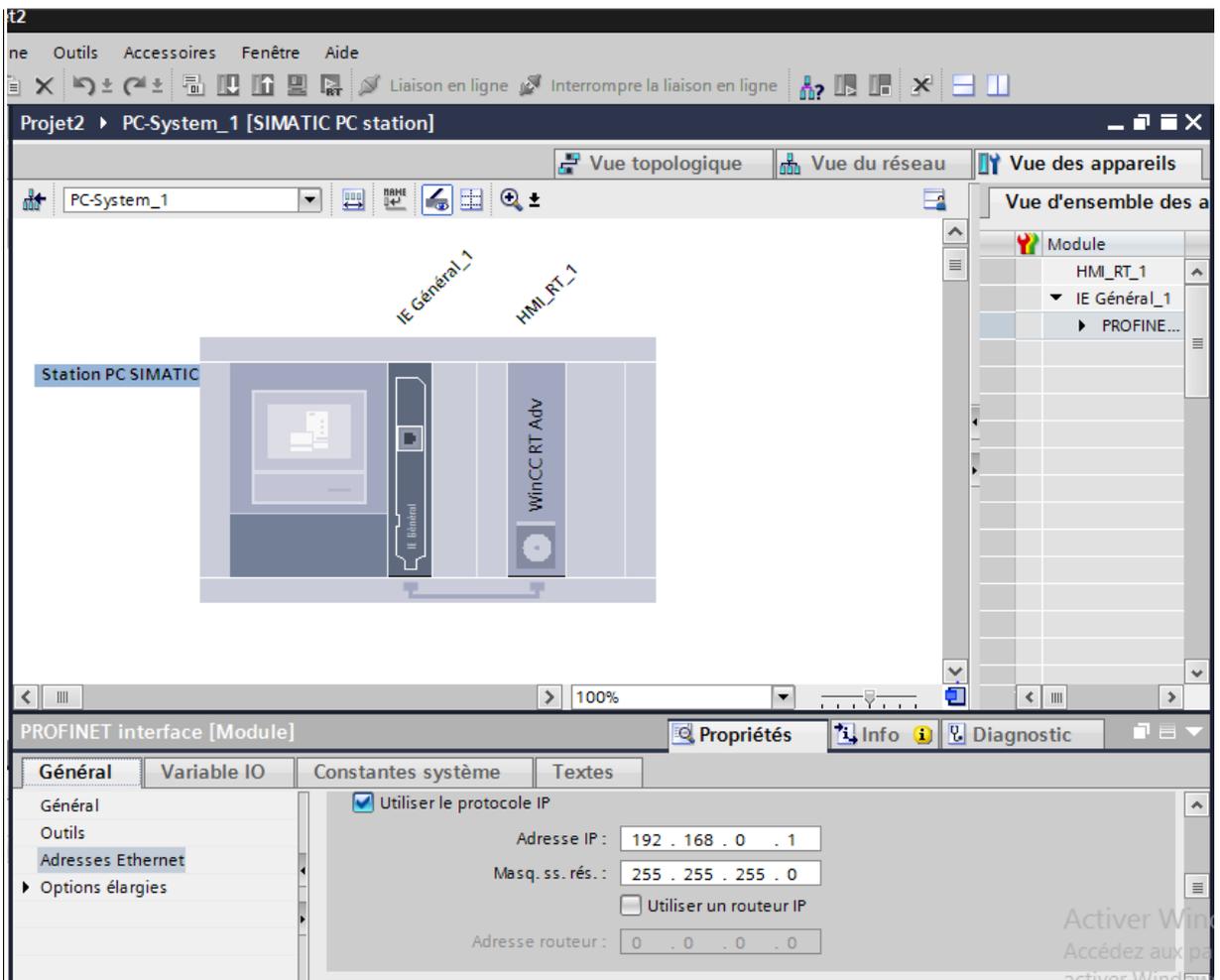


Figure III.8 : l'adresse IP et le masque de sous-réseau de la station runtime

Il faut d'abord créer une liaison HMI entre la CPU et l'HMI, cela pour pouvoir lire les données qui se trouvent dans l'automate ;

Pressez dans la vue du reseau et activez les (liaisons) pour insérer une liaison IHM. connectez l'interface des automates s7-1200 à celle de la station pc runtime. La liaison S7 devra etre choisi pour la communication entre PLC1 et PLC2 (figure III.9).

Chapitre III : Logiciel et programmation

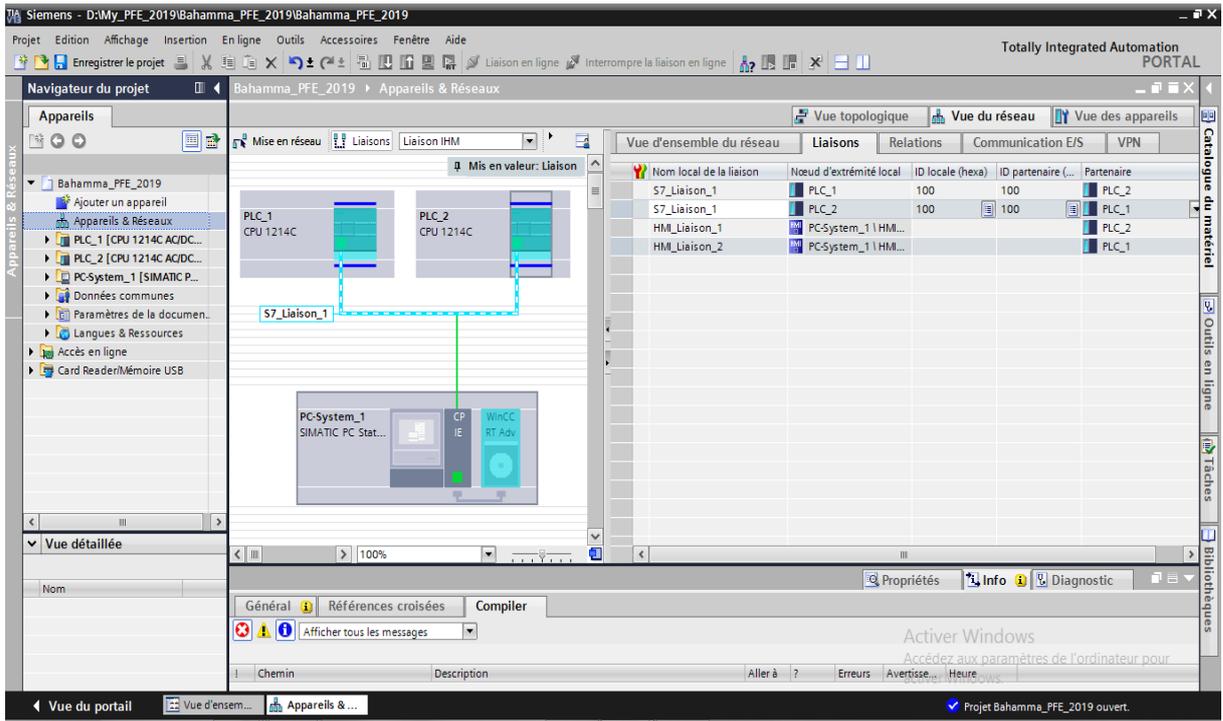


Figure III.9 : la communication entre PLC1 et PLC2 et Pc-System

III.4.1 Variables HMI

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

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

III.5 Programmation

III.5.1 Les blocs

Pour réaliser la tâche d'automatisations on doit charger dans l'automate les blocs qui contiennent les différents programmes et données. Les blocs existants sont (OB, FB, SFB, FC, SFC) qui contiennent les programmes, les blocs de données DB d'instance et DB globaux qui contiennent les paramètres du programme

En utilisant le bloc d'organisation cyclique OB1 et OB100 et autres DB et FC.

III.5.1.1 Les blocs d'organisation

Les blocs d'organisation (OB) servent d'interface entre le système d'exploitation et le programme utilisateur. Ces blocs d'organisation remplissent des tâches différentes bien précises.

✓ Programme de démarrage OB 100 :

La CPU effectue une mise en route après la mise sous tension, Démarrage complet ou à chaud : c'est le plus couramment utilisé, il est compatible avec tous les types de processeurs, seules les données rémanentes sont conservées.

Les variables qui ont été configurées comme rémanentes conservent donc leur dernière valeur, les autres variables ou blocs non rémanentes sont réinitialisées. Un OB100 est généralement utilisé pour initialiser des variables ou modules d'E/S au démarrage du programme

✓ Programme cyclique OB 1

On a utilisé le bloc d'organisation **OB1** qui est appelé par le système d'exploitation, il fait appel aux autres blocs qui constituent le programme, lorsqu'on appelle un bloc fonctionnel dans l'OB1 un bloc de donnée associé sera créé automatiquement.

Lors d'une exécution normale de programme, les traitements se font de façon cyclique. L'exécution du programme contenu dans l'OB 1 est démarrée une fois par cycle (quand il est fini, il recommence). On peut se servir de l'OB 1 pour appeler des blocs fonctionnels (FB, SFB) ou des fonctions (FC, SFC).

III.5.1.2 Blocs fonctionnelles (FB)

Le FB est un sous-programme écrit par l'utilisateur et exécuté par des blocs de code, On lui associe un bloc de données d'instance DB relatif à sa mémoire et contenant ses paramètres.

Pour ce programme on a utilisé quatre blocs de ce type, programmé en langage GRAPH,

Le FB est un sous-programme écrit par l'utilisateur et exécuté par des blocs de code, On lui associe un bloc de données d'instance DB relatif à sa mémoire et contenant ses paramètres.

III.5.1.3 Bloc fonction (FC)

La FC contient des routines pour les fonctions fréquemment utilisées. Elle est sans mémoire et sauvegarde ses variables temporaires dans la pile de données locales. Cependant elle peut faire appel à des blocs de données globaux pour la sauvegarde de ses données.

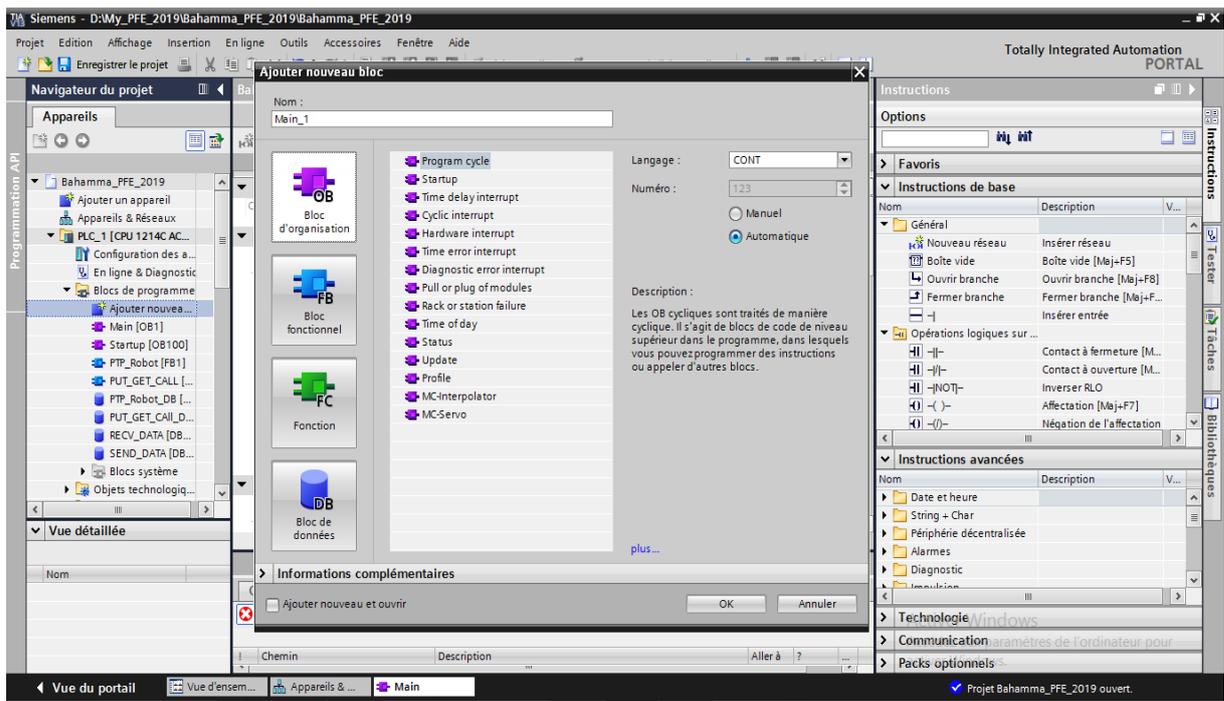


Figure III.10 : Bloc d'organisation OB1 et OB100 et BD et FB

III.5.1.4 Blocs 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 (figure III.23).

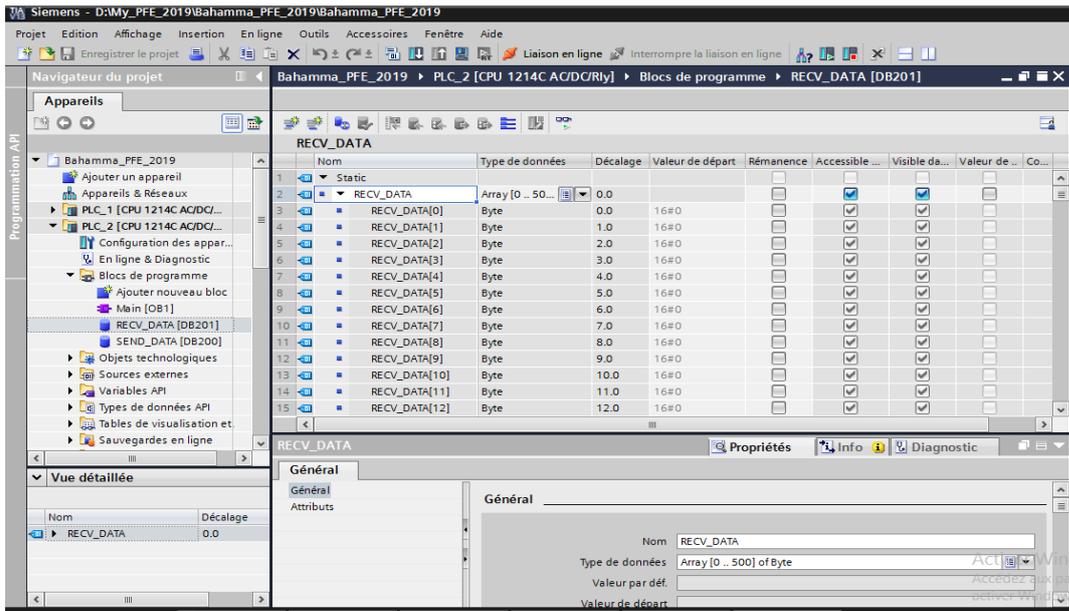


Figure III.11 : Blocs de données

III.5.2 Blocs système de communication PUT ET GET

Les blocs PUT et GET sont utilisés pour la communication via des liaisons S7.

✓ PUT :

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

S7-1200 : 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) P#DB201.DBX0.0 BYTE 160 et les données (SD_i) P#DB200.DBX0.0 sont envoyés à la CPU partenaire PLC2 (figure III.12).

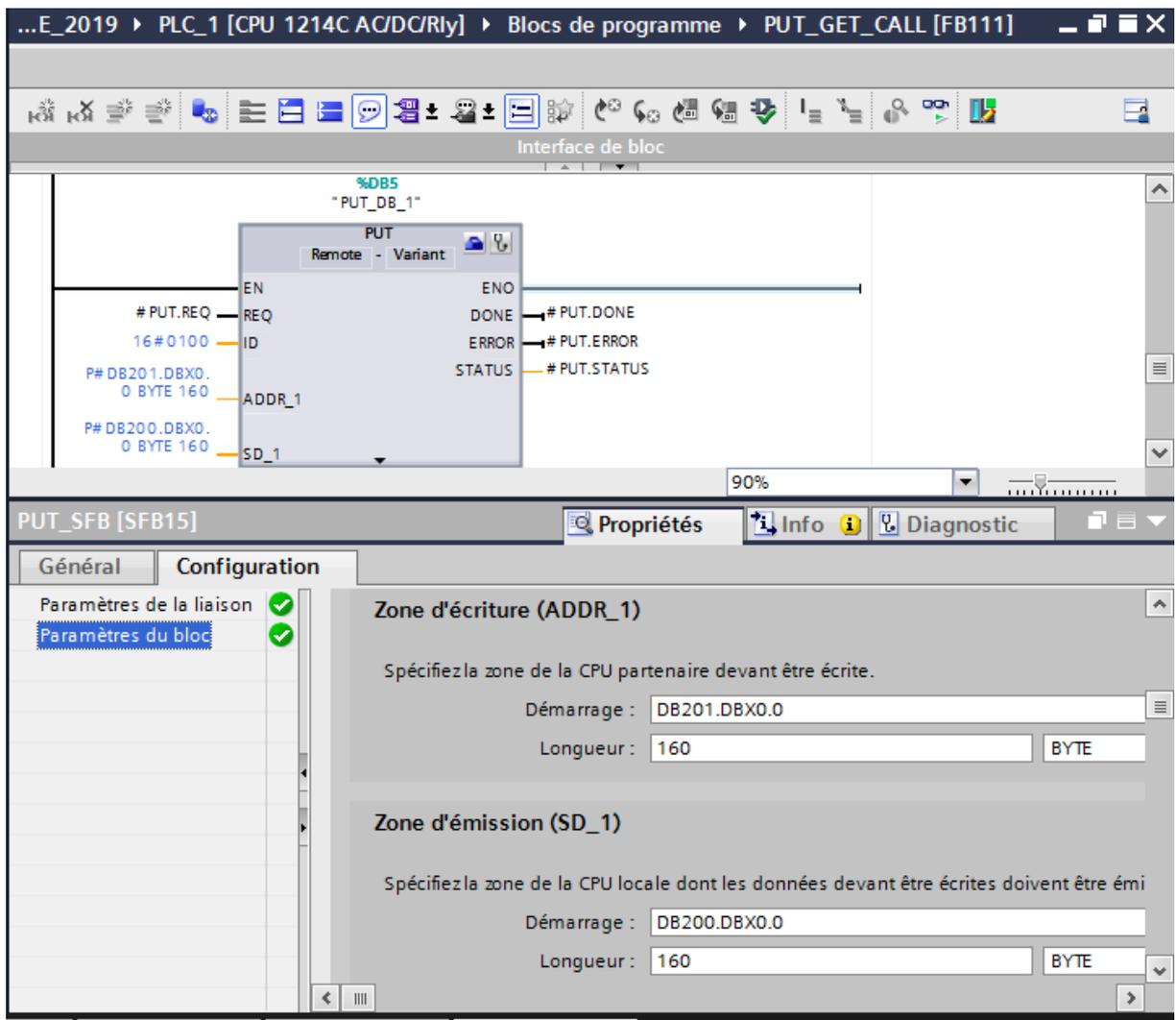


Figure III.12 : bloc PUT

✓ GET :

Bloc "GET" 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) P#DB200.DBX0.0 BYTE 160 sont envoyés à la CPU partenaire. La CPU partenaire peut être à l'état de fonctionnement Marche ou Arrêt.
- La CPU partenaire renvoie le contenu des données : Le dépassement de la taille maximale des données utiles par une réponse est signalé par le code d'erreur "2" au paramètre STATUS. Lors de l'appel d'instruction suivant, les données reçues sont copiées dans les zones de réception configurées (RD_i) P#DB201.DBX0.0 BYTE 160.

Chapitre III : Logiciel et programmation

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

Pour activer de nouveau une lecture, il faut attendre que la lecture précédente soit achevée. Si des problèmes d'accès aux données se sont produits pendant la lecture ou si le contrôle du type de données a révélé une erreur, les paramètres ERROR et STATUS émettent des erreurs et des avertissements (figure III.13).

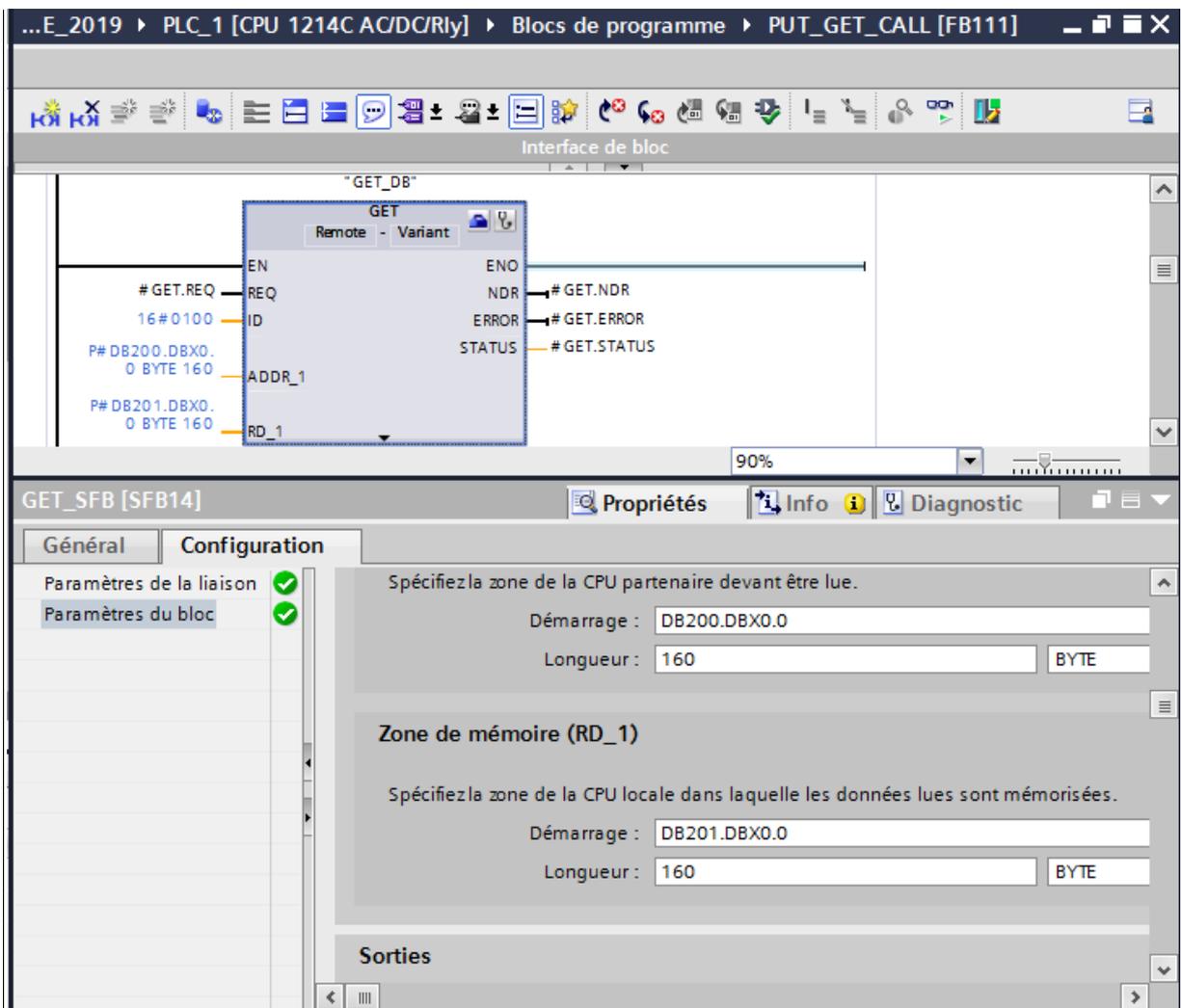


Figure III.13 : bloc GET

III.5.3 Programmation de la communication Client/serveur :

La communication S7 avec les services PUT et GET est utilisé dans ce projet d'application pour l'échange de données.

Dans ce projet, L'automate PLC_1 est considérée comme client car les blocs PUT et GET sont configure dans celle-ci.

Alors que la deuxième automate PLC_2 est considérée comme serveur.

L'automate PLC_1 sera ici le client pour la communication S7. la configuration de la liaison est programmée dans le client qui établit activement la liaison, ainsi que la réalisation des services PUT et GET.

Le bloc fonctionnel utilisateur FB4 (GET_PUT) est appelé dans le programme applicatif du L'automate PLC_1 .celui-ci encapsule l'appel des blocs fonctionnels systemes SFB14(GET) et SFB15(PUT) pour la transmission des données et fournit une interface simple à l'utilisateur par des paramètres.

Les données utiles sont envoyées et reçues cycliquement ou commandées sur événement avec le bloc fonctionnel utilisateur FB4 (GET-PUT).le nombre de données utiles et cibles pour les modules fonctionnels système SFB14(GET) et SFB15(PUT) sont fournis par les paramètres d'entrée du bloc utilisateur FB4(GET_PUT).

L'automate PLC_2 est ici le serveur la communication S7, qu'elle est associée passivement lors de l'établissement de la liaison. Dans L'automate PLC_2, aucune configuration de liaison, ni d'appel de blocs fonctionnels de communication ou de blocs fonctionnels systemes ne sont nécessaires pour la réalisation des services d'écriture et de lecture.

III.5.4 Liaison avec l'automate

Il faut maintenant charger la configuration de l'automate dans celui-ci. Pour cela, il faut tout d'abord connecter l'automate au PC en utilisant l'interface profinet adapté.

Ensuite, après avoir sélectionné la vue « En ligne et diagnostique », sélectionnez les options suivantes :

III.5.4.1 Paramétrage de l'interface PG/PC

Cet outil sert au paramétrage de l'adresse locale des PG/PC, la vitesse de transmission dans le réseau MPI ou PROFIBUS ou profinet en vue d'une communication avec l'automate et le transfert du projet.

Cet outil sert à paramétrer l'adresse locale des PG/PC, la vitesse de transmission dans le Réseau MPI (Multi-Point Interface ; protocole de réseau propre à SIEMENS) ou PROFIBUS en vue d'une communication avec l'automate et le transfert du projet.ux comme d'habitude sous forme de liaison S7 ou de liaison Send/Receive (figure III.14).

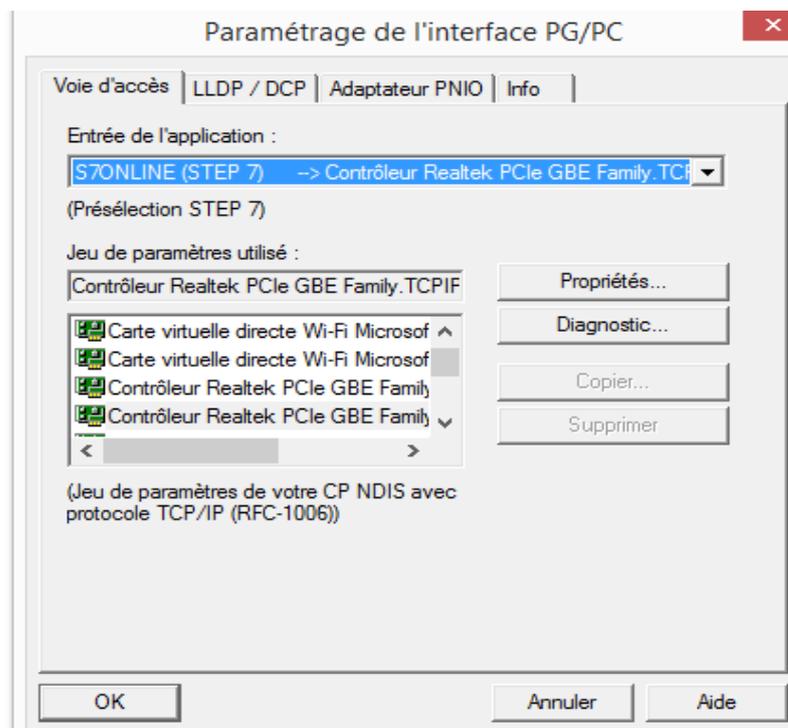


Figure III.14 : Paramétrage de l'interface PG/PC

III.5.5 WinCC sur TIA portal

WinCC (TIA portal) est un logiciel d'ingénierie pour la configuration de pupitres SIMATIC, de PC industriel SIMATIC et de PC standard par le logiciel de visualisation. Le SIMATIC WinCC dans le TIA portal fait partie d'un nouveau concept d'ingénierie intégré qui offre un

Chapitre III : Logiciel et programmation

environnement d'ingénierie homogène pour la programmation et la configuration de solution de commande, de visualisation d'entraînement, c'est le logiciel pour toutes les applications IHM allant de solutions de commande simples avec basic panels aux applications SCADA pour système multipostes basé sur PC (figure III.15) [28].

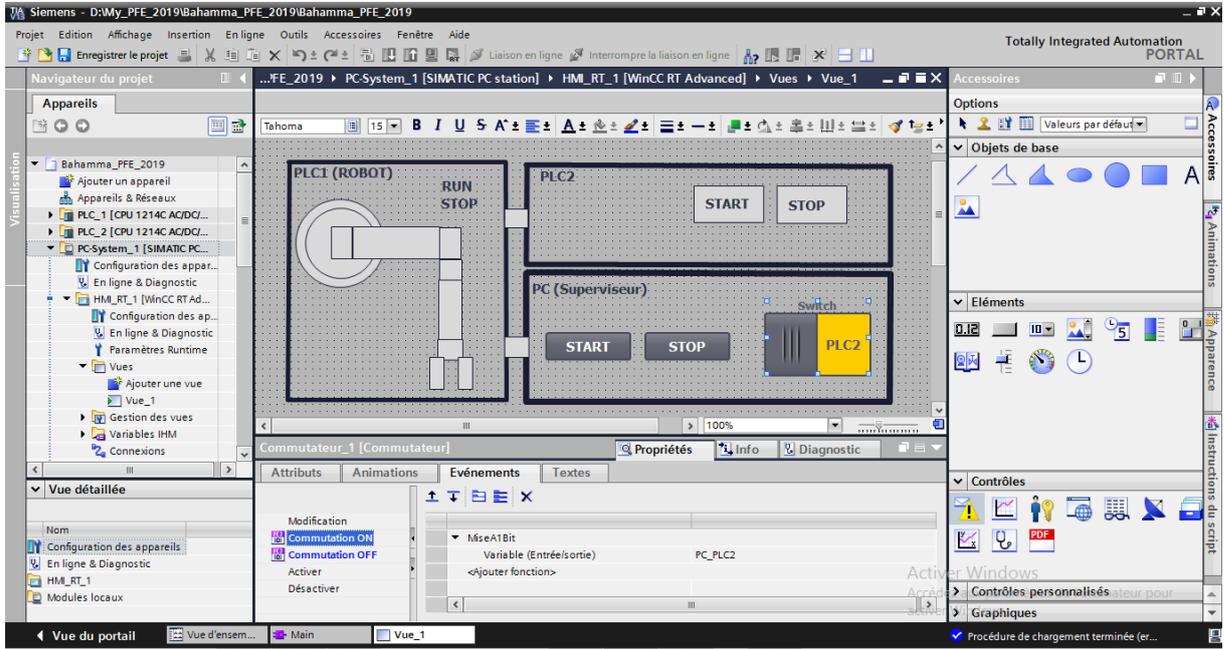


Figure III.15 : WinCC sur TIA portal

✓ Animation :

Chaque vanne est associé à une sortie automate, lorsque la variable de sortie est à « 0 » ou X_Adv_Ret est représenté en bleu (vanne fermé), lorsque elle est à « 1 » la X_Adv_Ret est en vert (figure III.16).

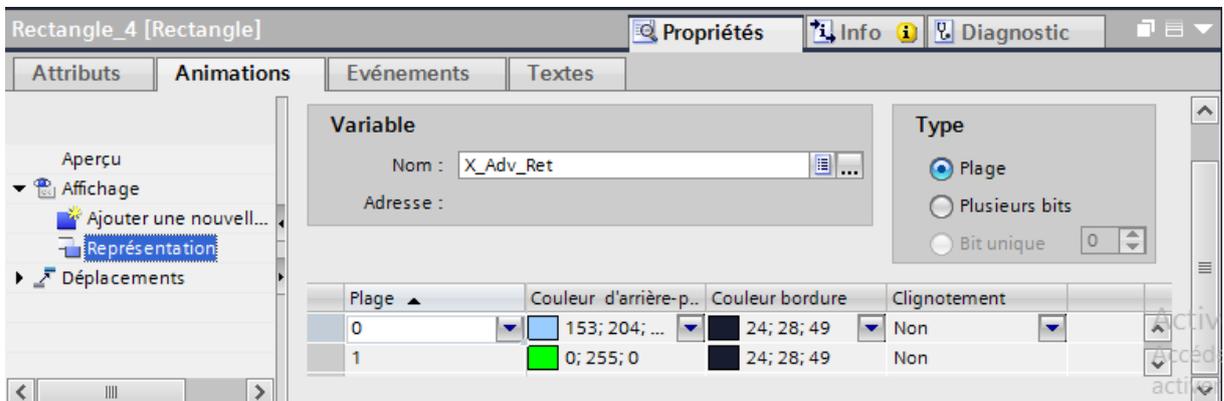


Figure III.16 : Animation d'une X_Adv_Ret

- ✓ **Evènement** : Chaque bouton est associé à un memento, en appuyons sur le bouton on change l'état de ce memento dans l'automate soit à « 1 » ou à « 0 ».

Affichage du nom de l'étape active

- ✓ **Visibilité** :

Le nom d'étape apparaît en vert clignotant si l'étape est active, sinon il est invisible.

III.6 Compilation et simulation

III.6.1 PLCSIM

En l'absence de l'automate et des moyens est nous n'avons pas pu réaliser l'armoire électrique et tester réellement l'exécution du programme.

Nous avons utilisé un logiciel optionnel de TIA Portal, ce logiciel nommé PLCSIM permet de simuler un automate de la famille SIEMENS avec tous ces modules. Le simulateur présente une interface simple et accessible, en effet pour changer l'état d'une entrée, il suffit de cocher la case correspondante, les états des sorties changent automatiquement selon l'évolution du programme. Lors de la simulation et dans la fenêtre de programmation(CONT), chaque contact représentant une variable active est affiché en vert (Les contacts non actifs en pointillé).Ceci permet de suivre l'évolution du programme en détails. La simulation nous a permis de tester les différentes situations que peut affronter le système.

Nous concluons à la fin que notre programme répond exactement aux exigences du cahier des charges et qu'il peut donc être transféré du PC vers l'automate qui lui correspond.

L'application de simulation S7-PLCSIM nous a permis d'exécuter et de tester notre programme qu'on a simulé sur ordinateur. La simulation a été complètement réalisée au sein du logiciel TIA portal, cette application permet de tester des programmes destinés aux CPU S7, et de remédier à d'éventuelles erreurs.

Pour la CPU, la simulation est complètement réalisée au sein du logiciel TIA Portal V13. En effet, S7-PLCSIM dispose une interface comportant une CPU S7-1214C virtuelle et des modules d'entrées/sorties qui permettent de visualiser et forcer les différents états du programme [29].

III.6.2 RUNTIME

Après avoir créé le projet et terminé sa configuration, il est indispensable de vérifier la cohérence du projet, et de détecter les erreurs, à l'aide de la commande sur la barre du menu compiler, après la compilation, le système crée un fichier de projet compilé.

La simulation permet de détecter des erreurs logiques de configuration, par exemple, des valeurs limites incorrectes, et cela à l'aide du simulateur SIMATIC WinCC RT Advanced.

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

- Communication avec les automates
- Affichage de vue à l'écran.
- Commande du processus, par exemple, spécification de consignes ou ouverture et fermeture de vannes.
- Affichage des données de Runtime actuelles, des valeurs processus et événement d'alarme (**Figure III.17**) [30].

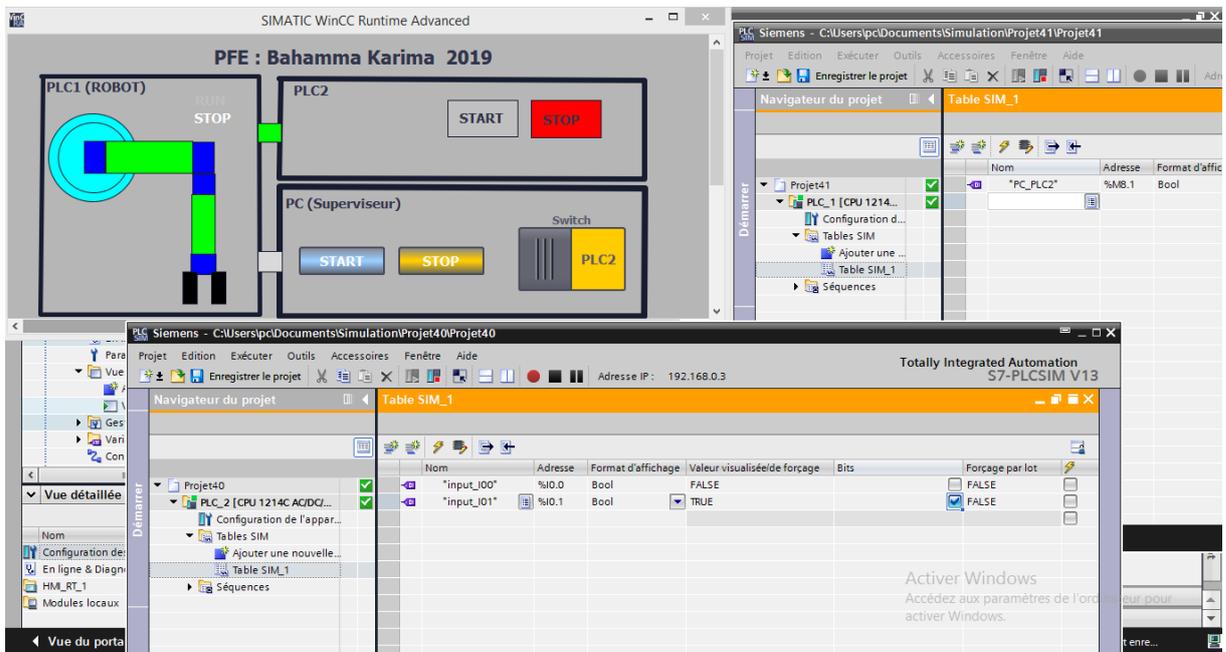


Figure III.17: Le simulateur de S7-1214C

III.7 Réalisation pratique :

Le système est constitué des éléments suivants :

S7-1200 CPU 1214 AC/DC/RLY (PLC1) avec module d'E/S (8 E/S).

S7-1200 CPU 1214 AC/DC/RLY (PLC2) avec module d'E/S (16 E/S)

Switch

TIA Portal (**figure III.18**)



Figure III.18 : Réalisation pratique

III.8 Conclusion

Dans ce chapitre, nous avons présenté le logiciel de programmation des automates SIEMENS et le logiciel de la supervision le WINCC Communication CPU Sous PROFINET IO, vous simulé et réalisé à l'aide de deux automates programmables, un switch et une station PC dont TIA Portal est installé comme outil de configuration, de programmation et de commande.

Conclusion générale

L'étude menée dans ce projet nous a permis d'appréhender un nouveau monde qui est celui des systèmes de contrôle instrumenté.

Dans l'industrie manufacturière, l'automatisation industrielle est le processus d'intégration de machines et d'équipements industriels pour effectuer automatiquement des tâches telles que le soudage, la manutention des matériaux, l'emballage, la palettisation, la distribution, le découpage, etc... L'utilisation de matériels et de logiciels d'automatisation permet d'améliorer productivité, sécurité et rentabilité. L'automatisation apporte de nombreux avantages lorsqu'elle est intégrée correctement.

Ce travail est basé un réseau industriel est simulé et réalisé à l'aide de deux automates programmables, un switch et une station PC dont TIA Portal est installé comme outil de configuration, de programmation et de commande.

On a établi le programme en langage contact sous Tia Portal et on a réalisé la supervision de notre système avec le Logiciel de supervision WinCC Advanced, et ce après avoir réalisé une connexion entre deux automates et le PC contenant le WinCC RT. Le bus utilisé dans ce réseau est PROFINET.

On a pu aussi faire une simulation du travail sur des automates de simulation virtuelle à l'aide de PLCSIM intégré dans le logiciel Tia Portal, et la visualisation sur WINCC RT.

Finalement on peut dire que la combinaison Tia Portal logiciel de programmation et WinCC RT logiciel de supervision sont des outils très puissants et très utilisés qui permettent une bonne gestion et un suivi des systèmes de production.

Bibliographie

- [1] Henri Nussbaumer, « informatique industrielle III », Press polytechniques de Romandes 1987
- [2] MICHEL Groult et Patrick Salam, Instrumentation industriel : 3eme édition
- [3] Bertrand. M. 'Automate programmable industriels'. Technique de l'ingénieur, S8015, page 14, 2001.
- [4] UGERGAZI Khaled, SAHRAOUI Hamza « étude de la tréfileuse et sa commande automatique », thèse d'ingénieur de Electrotechnique, Biskra 2006. [5] Gilles MICHEL, Claude LAURGEAU, Bernard ESPIAU « les automates programmables industrielle », édition DUNOD. 1979.
- [6] [http://www.courselec.free.fautomatisme/Automate programmable.htm](http://www.courselec.free.fautomatisme/Automate%20programmable.htm)
- [7] Y.Felfoul & N.Bekkouche, 'Etude de la commande de la station de préparation de colle de la chaîne de fabrication du panneau de particules à base d'automate SIEMENS', PFE ingénieur université de Bejaia, 2006.
- [8] : Document technique: *Rapport d'entreprise BYA electronic*, 07-2011.
- [9] P. Bonnet, *Cours informatique industrielle, E/S pour API*, USTL 2012.
- [10] Slim BEN SAOUD, *Automatismes*, INSAT, 2015.
- [11] Jean-Claude Humbolt, « Automate programmables Industriels » édition DUNOD. 1993.
- [12] F. Benchallalet A.Saadi, « *Étude et automatisation d'une conditionneuse sous vide THERA 450 colimatic* », Mémoire fin d'étude, USTHB, 2016.
- [13] G .Michel, « Architecture et application des automates programmable industriels » DUNOD , 1987
- [14] William Bolton, AUTOMATES PROGRAMMABLES INDUSTRIELS, chapitre 1 p 16, 2^{ème} Edition
- [15] : ALAIN GONZAGA " LES AUTOMATES PROGRAMMABLES
- [16] : André SIMON, « Automates programmables, programmation, et logique programmé », Edition L'ELANE, 1983
- [17] SIMATIC S7-1200 ; N° de référence E20001-A1860-P272-X-7700 , 2011.
- [18] Siemens MANUAL_SITOP-PSU8600-MP.
- [19] Raimond pigan et Marck metter, Automating with PROFINET, siemens A&D translation services , 22 octobre 2008.

Bibliographie

- [20]S71500_communication_function-manual,en-us_en-us.pdf.
- [21] Luca Lachello et all, «Industriel Ethernet FACTS», Volume 6, Numéro 2, 2ème Edition, Février 2013, p 7
- [22] Manuel système, PROFINET Description du système. SIEMENS 03/2012
- [23] PROFINET, L'Ethernet industriel comme standard ouvert pour une rentabilité immédiate de l'automatisation, SIEMENS, 2008
- [24]Communication industrielle pour l'automatisation.pdf
- [25] Solutions de réseaux pour PROFIBNET, Brochure technique, Avril 2008
- [26] <https://fr.wikipedia.org>
- [27]52000543.Profinet8_Basics.16.pages_16.pdf
- [28] WinCC flexible 2008 Compact / Standard / Advanced 396 User's Manual, 07/2008, 6AV6691-1AB01-3AB0
- [29] Manuel SIEMENS, « STEP7 PLCSIM », Testez vos Programme, 2002.
- [30] Hans Berger, "Automating with SIMATIC", second edition, 2003.