

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



**Ecole Nationale Polytechnique**

Département Automatique  
Entreprise CATEL



## **Mémoire de projet de fin d'études pour l'obtention du diplôme d'ingénieur d'état en Automatique**

Thème :

**Automatisation d'une chaîne de gainage de câble téléphonique  
à base d'automate SIEMENS**

Abderrazak TAHMI

Zakaria SAADI

Présenté devant le jury composé de :

Président: O.STIHI

Rapporteur: E.M.BERKOUK

Examineur: L. ABDELOUEL

Invité : ADEL.EL ITIM

Chargé de Cours à l'ENP

Professeur à l'ENP

Chargé de Cours à l'ENP

Ingénieur à CATEL



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



**Ecole Nationale Polytechnique**

Département Automatique  
Entreprise CATEL



## **Mémoire de projet de fin d'études pour l'obtention du diplôme d'ingénieur d'état en Automatique**

Thème :

**Automatisation d'une chaîne de gainage de câble téléphonique  
à base d'automate SIEMENS**

Abderrazak TAHMI

Zakaria SAADI

Présenté devant le jury composé de :

Président: O.STIHI

Rapporteur: E.M.BERKOUK

Examineur: L. ABDELOUEL

Invité : ADEL EL ITIM

Chargé de Cours à l'ENP

Professeur à l'ENP

Chargé de Cours à l'ENP

Ingénieur à CATEL

ENP 2017

## *Je dédie ce travail*

✚ *A mes chers parents, pour leur soutien, leur patience, leur sacrifice et leur amour, vous méritez tout éloge, vous qui avez fait de moi et ma sœur et frères ce que nous sommes maintenant. Nous espérons être l'image que vous êtes fait de nous, que dieu nous garde et nous bénisse.*

✚ *Merci Maman, Merci papa*

✚ *A mes frères : Ahmed et Mohamed,*

✚ *A ma sœur : Imene*

✚ *A mon neveu : Aymen*

✚ *A mes nièces : Sirine et maria*

✚ *A mes tentes : Wahiba et Fiala*

✚ *A toute la famille SAADI et MAMANE*

✚ *A mes amies et camarades : du primaire, moyen,*

✚ *Secondaire, EPST Oron et EMP d'Alger.*

✚ *A Ali, Mouad, Ayoub, Ahmed KHALF et à tous les membres et les anciens membres du groupe « Khelil boualem »*

*Zakaria SAADI*

*Je dédie ce travail :*

✚ *A mes chers parents, pour leur soutien, leur patience, leur sacrifice et leur amour, vous méritez tout éloge, vous qui avez fait de moi et mes sœurs et frères ce que nous sommes maintenant.*

✚ *Nous espérons être l'image que vous êtes fait de nous, que dieu vous garde et vous bénisse.*

✚ *A ma grand-mère, qui m'a toujours soutenue, qu'Allah la protège.*

✚ *A mon frère, mes sœurs,*

✚ *A mes amies et camarades : du primaire, moyen, secondaire, univ-annaba, EPSJA et EMP d'Alger.*

✚ *Et à tous ceux qui m'aiment et j'aime*

*Tahmi abderrazak*

## REMERCIEMENTS

*Nous remercions Dieu le tout puissant qui nous a donné le courage et la volonté de réaliser ce travail.*

*Nous tenons à exprimer nos vifs remerciements à notre promoteur Pr. BERKOUK de l'Ecole Nationale Polytechnique pour nous avoir encadré durant notre projet de fin d'études et nous conseillé tout le long de notre travail.*

*Nous remercions également notre Co-promoteur Mr. EL ITIM ADEL, pour son encadrement et sa confiance au sein de l'entreprise.*

*Nous remercions chaleureusement les membres du jury pour l'honneur qu'ils nous ont fait en acceptant d'évaluer notre projet.*

*Nous souhaitons aussi remercier tous les enseignants de l'Ecole Nationale Polytechnique d'Alger, et en particulier, Nos professeurs d'Automatique qui nous ont encadrés auparavant et tous nos enseignants pour les connaissances qu'ils nous ont transmis, leur disponibilité et leurs efforts.*

*Enfin, tous nos remerciements à toutes les personnes qui ont contribué de près ou de loin pour l'accomplissement de ce travail.*

## ملخص:

العمل المنجز في هذه المذكرة يتمحور أساسا حول استخدام آلي مبرمج "سيمنس" بغية التحكم آليا في خط تغليف الاسلاك الكهربائية والهاتفية في شركة كاتل، وعليه استخدمنا برنامج "تيا بورتال" لكتابة وتحميل البرنامج في الآلي وكذلك لتصميم واجهات شاشات التحكم والمراقبة.

**الكلمات المفتاحية:** الآلي المبرمج "سيمنس"، "تيا بورتال"، خط التغليف.

## Abstract:

The work presented in the thesis is essentially based on the use of the programmable Logic controller of "SIEMENS" for the automation of the Cladding chain of electrical and telephonic cables at the company CATEL, by the use of the software "Portal TIA" Writing and uploading programs to the PLC and for the design of control and supervision interfaces.

**Keywords:** PLC "SIEMENS", TIA portal, Cladding chain.

## Résumé :

Le travail présenté dans ce mémoire se base essentiellement sur l'utilisation des automates programmables « SIEMENS » afin d'automatiser la ligne de gainage des câbles électriques et téléphoniques au niveau de l'entreprise CATEL, par l'utilisation du logiciel « TIA Portal » à l'écriture et au transfert des programmes vers l'API et aussi pour la conception des interfaces de commande et de supervision.

Mots clés : API « SIEMENS », TIA portal, chaîne de gainage.

# TABLE DES MATIERES

## LISTE DES FIGURES

## LISTE DES ABREVIATIONS

<i>Introduction générale</i> .....	12
<i>Chapitre 1 : Les automates programmables industriels</i> .....	14
<b>1.1 Introduction :</b> .....	15
<b>1.2 Historique :</b> .....	15
<b>1.3 Définition :</b> .....	15
<b>1.4 La structure d'un système automatisé :</b> .....	16
1.4.1 Partie commande : .....	16
1.4.2 Partie opérative (puissance) :.....	16
<b>1.5 Les types d'automate programmable industriel (API) :</b> .....	16
1.5.1 Les automates compacts : .....	16
1.5.2 Les automates modulaires : .....	17
<b>1.6 Architecture d'un API :</b> .....	17
1.6.1 Le processeur :.....	18
1.6.2 Les mémoires :.....	19
1.6.3 Modules d'entrées/sorties : .....	19
1.6.4 L'alimentation électrique : .....	20
1.6.5 Les liaisons : .....	20
1.6.6 Eléments auxiliaires :.....	20
<b>1.7 Les langages de programmation :</b> .....	21
<b>1.8 Le diagnostic de l'API :</b> .....	21
1.8.1 Vérification du minutage :.....	21
1.8.2 Dernière sortie activée :.....	21
1.8.3 La réplication :.....	21
1.8.4 Vérification d'une valeur attendue : .....	21
<b>1.9 La communication et les Réseaux industriels :</b> .....	22
1.9.1 Le Profibus : .....	22
1.9.2 Le Profinet :.....	22
<b>1.10 La Supervision :</b> .....	23
<b>1.11 Conclusion :</b> .....	23
<i>Chapitre 2 : TIA portal, logiciel de programmation et supervision</i> .....	24
<b>2.1 Introduction :</b> .....	25
<b>2.2 Présentation générale du logiciel :</b> .....	25
2.2.1 Définition :.....	25
2.2.2 Utilisation du logiciel : .....	25

2.2.3	Structure de l'interface utilisateur du TIA Portal : .....	25
<b>2.3</b>	<b>Créer et gérer des projets : .....</b>	<b>27</b>
2.3.1	Création d'un nouveau projet : .....	27
2.3.2	Ouvrir des projets : .....	28
2.3.3	Affichage des propriétés du projet : .....	29
<b>2.4</b>	<b>Editer des données de projet : .....</b>	<b>30</b>
2.4.1	Compilation des données du projet : .....	30
2.4.2	Charger des données de projet : .....	31
2.4.3	Concept de protection des données du projet : .....	33
<b>2.5</b>	<b>Utilisation de projets de référence : .....</b>	<b>33</b>
<b>2.6</b>	<b>Simuler des appareils : .....</b>	<b>34</b>
<b>2.7</b>	<b>Edition des appareils et réseaux : .....</b>	<b>34</b>
2.7.1	Configuration des appareils : .....	36
2.7.2	Configuration des réseaux : .....	40
2.7.3	Diagnostiquer les appareils et les réseaux : .....	41
<b>2.8</b>	<b>Programmation de l'API : .....</b>	<b>43</b>
2.8.1	Création du programme utilisateur : .....	43
2.8.2	Créer et gérer des tables des variables API : .....	45
2.8.3	Programmation des blocs : .....	46
2.8.4	Affichage des informations programme : .....	46
<b>2.9</b>	<b>Visualisation de processus : .....</b>	<b>48</b>
2.9.1	Créer et éditer objet HMI : .....	49
2.9.2	Créer les vues : .....	49
2.9.3	Utilisation des objets : .....	50
2.9.4	Dynamiser des vues : .....	53
2.9.5	Utilisation de plans : .....	53
2.9.6	Utilisation des variables : .....	53
2.9.7	Utiliser les archives : .....	53
2.9.8	Travailler avec des alarmes : .....	54
2.9.9	Utilisation de recettes : .....	55
2.9.10	Compiler et charger des projets : .....	56
<b>2.10</b>	<b>Conclusion : .....</b>	<b>57</b>
<b>Chapitre 3 : Présentation de la ligne de gainage BMN120.....</b>		<b>58</b>
<b>3.1</b>	<b>Introduction : .....</b>	<b>59</b>
<b>3.2</b>	<b>Câbles de télécommunications en cuivre: .....</b>	<b>59</b>
3.2.1	Les câbles urbains souterrains : .....	59
3.2.2	Les câbles urbains autoportés : .....	60

<b>3.3</b>	<b>La chaine de gainage BMN120 :</b>	<b>60</b>
3.3.1	Définitions :	61
<b>3.4</b>	<b>Les éléments de la chaine :</b>	<b>62</b>
<b>3.5</b>	<b>Le fonctionnement de la chaine :</b>	<b>64</b>
3.5.1	Le dévidoir :	64
3.5.2	La station Allupe :	64
3.5.3	L'Extrudeuse :	65
3.5.4	La Pompe à vide :	66
3.5.5	Les Bacs d'eau :	66
3.5.6	Le Sécheur :	67
3.5.7	Marqueur :	67
3.5.8	La chenille :	68
3.5.9	Le Bobinoir :	68
3.5.10	La carte électronique des vitesses :	68
<b>3.6</b>	<b>Conclusion :</b>	<b>69</b>
<b>Chapitre 4 : Conception et développement du système d'automatisation ...</b>		<b>70</b>
<b>4.1</b>	<b>Introduction :</b>	<b>71</b>
<b>4.2</b>	<b>Définition des E/S de la ligne de gainage :</b>	<b>71</b>
<b>4.3</b>	<b>Mode opératoire :</b>	<b>72</b>
<b>4.4</b>	<b>Cahier des charges :</b>	<b>72</b>
<b>4.5</b>	<b>Création du projet TIA portal :</b>	<b>77</b>
4.5.1	Objet API :	77
4.5.2	La configuration matérielle :	78
4.5.3	La hiérarchie du programme :	78
4.5.4	Ecriture du programme :	79
4.5.5	Exemples de programmation :	79
4.5.6	Objet HMI :	80
<b>4.6</b>	<b>Conclusion :</b>	<b>83</b>
<b>Conclusion générale.....</b>		<b>84</b>
<b>Bibliographie.....</b>		<b>86</b>
<b>Annexe A : les langages de programmation des APIs .....</b>		<b>87</b>
<b>Annexe B : les alarmes système et les icones de diagnostic .....</b>		<b>91</b>

# LISTE DES FIGURES

Figure 1-1 structure d'un système automatisé .....	16
Figure 1-2 : S7-300 compact (S7-314C). .....	17
Figure 1-3: S7-300 modulaire (S7-317). .....	17
Figure 1-4: la structure d'un API. ....	18
Figure 1-5: les différentes liaisons possibles d'un API [2]. .....	20
Figure 1-6: pupitres tactiles SIEMENS. ....	23
Figure 2-1: vue de portail [5]. .....	25
Figure 2-2: vue de projet [5]. .....	26
Figure 2-3: création d'un nouveau projet .....	28
Figure 2-4: ouvrir d'un projet existant .....	28
Figure 2-5 : la fenêtre de mise à niveau .....	29
Figure 2-6 : la fenêtre où on gère les propriétés d'un projet .....	30
Figure 2-7 : la fenêtre qui affiche l'état de compilation .....	31
Figure 2-8 : la fenêtre de chargement .....	32
Figure 2-9 : la fenêtre de projet de référence sous TIA portal .....	34
Figure 2-10 : la fenêtre de l'éditeur des appareils et réseaux .....	35
Figure 2-11 : la vue du réseau .....	35
Figure 2-12 : la vue des appareils .....	35
Figure 2-13 : exemple d'utilisation des châssis .....	36
Figure 2-14 : insertion d'un nouvel appareil .....	36
Figure 2-15 : insertion d'un appareil non spécifié .....	37
Figure 2-16 : insertion d'un module .....	38
Figure 2-17 : Edition des propriétés dans la vue des appareils .....	39
Figure 2-18 : voir les variables d'E/S affecté à un module .....	39
Figure 2-19 : configuration des réseaux .....	40
Figure 2-20 : configuration et paramétrage d'un réseau MPI .....	41
Figure 2-21 : la vue en ligne et diagnostic .....	41
Figure 2-22 : le tableau des informations de diagnostic des appareils .....	42
Figure 2-23 : graphes qui représentent la programmation linéaire et structuré .....	44
Figure 2-24 : les différents blocs d'organisation .....	45
Figure 2-25 : création d'une table des variables .....	46
Figure 2-26 : interface de programmation des blocs .....	46
Figure 2-27 : la vue du tableau d'affectation .....	47
Figure 2-28 : la vue qui représente la hiérarchie d'appel des blocs .....	47
Figure 2-29 : la vue qui représente la structure de dépendance .....	48
Figure 2-30 : la vue qui affiche l'utilisation des ressources matérielles de la CPU .....	48
Figure 2-31 : Ajouter un appareil HMI .....	49
Figure 2-32 : création d'une nouvelle vue .....	50
Figure 2-33 : tableau des objets simples .....	51
Figure 2-34 : Tableau des objets Eléments .....	51
Figure 2-35 : Tableau des objets de contrôles .....	52

Figure 2-36 : insérer un objet dans une vue.....	52
Figure 2-37 : la fenêtre « Archives de variables ».....	53
Figure 2-38 : la fenêtre « Archives des alarmes ».....	54
Figure 2-39 : Un graphique qui représente les types des alarmes .....	54
Figure 2-40 : la fenêtre de création des recettes .....	55
Figure 2-41 : utilisation l'objet vue de recette dans la vue de supervision .....	55
Figure 2-42 : la structure de compilation et de chargement des projets HMI .....	57
Figure 3-1 : la structure de construction du câble urbain souterrain (série 89) .....	59
Figure 3-2 : la structure de construction d'un câble autoporté (série 98) .....	60
Figure 3-3 les modes d'assemblage .....	61
Figure 3-4 : nomenclature du câble .....	62
Figure 3-5 : Les constituants d'un câble (séries 98-99) .....	62
Figure 3-6 : présentation globale de la chaîne .....	63
Figure 3-7 : le dévidoir .....	64
Figure 3-8 : la station Allupe .....	65
Figure 3-9 : principe de fonctionnement de l'extrudeuse.....	65
Figure 3-10 : l'extrudeuse.....	66
Figure 3-11 : l'image de la pompe à vide .....	66
Figure 3-12 : les bacs d'eau .....	67
Figure 3-13 : l'image du sécheur .....	67
Figure 3-14 : le marqueur .....	67
Figure 3-15: la chenille.....	68
Figure 3-16: le bobinoir.....	68
Figure 3-17 : la carte électronique des vitesses .....	69
Figure 4-1 : Grafset de l'appareil à souder.....	73
Figure 4-2; Grafset du con_evateur.....	73
Figure 4-3: la pompe à vide.....	74
Figure 4-4 : zone 1, 2, 3 et 4 de chauffage .....	74
Figure 4-5 : chauffage de la tête .....	75
Figure 4-6 : la pompe du bac d'eau 1 .....	75
Figure 4-7 : la pompe du bac d'eau 2 .....	76
Figure 4-8 : sécheur .....	76
Figure 4-9 : grafset moteur d'extrudeuse et moteur de tirage .....	77
Figure 4-10 ; configuration matérielle.....	78
Figure 4-11 : La hiérarchie du programme.....	79
Figure 4-12 : appelle du bloc fonctionnel par le bloc de données sécheur (DB10) .....	79
Figure 4-13 : le bloc fonctionnel TOR en langage CONT (Ladder) .....	80
Figure 4-14 : vue d'accueil .....	80
Figure 4-15 : vue principale.....	81
Figure 4-16 : vue zones de chauffage .....	82
Figure 4-17 : les éléments de la recette .....	82
Figure 4-18 : vue de l'archive des alarmes et la recette .....	83

# LISTE DES ABREVIATION

**API:** Automate Programmable Industriel

**CPU:** Central Processing Unit

**EPROM:** Erasable Programmable Read Only Memory

**FB :** Blocs fonctionnels

**FBD :** Function Bloc Diagram

**FC :** Fonctions

**GRAFCET :** Graphe Fonctionnel de Commande Etape Transition

**HMI :** Interface Homme Machine

**IL :** Instruction List

**LD :** Ladder Diagram

**LIST :** Le langage de liste d'instructions

**LOG :** Le langage à base de logigramme

**MPI :** Multi Point Interface

**OB :** Blocs d'organisation

**OP :** Pupitre opérateur

**PC :** Personnel Computer

**PG :** La console de programmation sur le terrain

**PLC:** Programmable Logic Controllers

**P.O:** partie opérative

**PROFIBUS:** Process Field Bus

**RAM:** Random Access Memory

**ROM:** Read Only Memory

**SFB:** Bloc Fonctionnel Système

**SFC :** Bloc programme pour le langage évolué textuel

**ST :** Structured Text

**TOR :** Toute Ou Rien

**WinCC :** Windows Contrôle Center

**CATEL :** Câblerie de Télécommunication

## **Introduction générale**

Avec l'évolution de la civilisation et depuis toujours, les besoins de l'homme et ses demandes sont en augmentation, ce qui a ouvert les portes pour lancer des entreprises et des usines et qui a engendré une forte concurrence pour satisfaire ces besoins. Le temps et la qualité des produits sont devenus parmi les plus grands enjeux de ces entreprises.

Afin de pouvoir répondre à cette contrainte de rapidité en garantissant une meilleure qualité, l'entreprise industrielle est appelée à raccourcir les délais de l'ensemble des étapes de sa chaîne de production en remplaçant les opérateurs qui ont des tâches répétitives par des robots ou des machines automatisées.

Aujourd'hui l'automatisation est devenue une solution indispensable pour remédier à ces obligations, on peut la trouver ainsi dans d'autres secteurs comme la gestion automatique d'un parking d'automobiles, la gestion des feux de signalisation ...etc.

L'automatisme est devenu une discipline importante dans tous les secteurs industriels, il permet de faciliter les tâches des opérateurs intervenants sur les sites industriels et de développer des systèmes qui assurent des tâches répétitives, dangereuses et dans des milieux hostiles pour l'homme. L'automatisation augmente la productivité et améliore la qualité du produit. Une bonne automatisation, en plus d'un fonctionnement fiable de l'installation industrielle, elle assure une détection de toute anomalie éventuelle.

Dans toute chaîne automatisée l'automate programmable industriel représente le cœur de cette chaîne, c'est l'élément le plus important et le cerveau qui assure ses performances. Il est aussi flexible et facile à la maintenance et à la réutilisation.

La diversité des processus industriels nécessite des connaissances sur l'aspect processus et les différentes technologies du domaine de l'automatisme. Les automatismes câblés qui existaient avant les automates programmables industriels (APIs) présentent des inconvénients majeurs relatifs à leurs configurations et maintenances.

Notre travail au sein de l'entreprise « CATEL » était sur l'étude et l'analyse de la chaîne de gainage « BMN 120 », on a reformulé le cahier des charges afin d'automatiser la chaîne à base d'un automate programmable « SIEMENS ».

L'objectif c'est de rendre la chaîne plus fiable, plus sécurisée et aussi de faciliter les tâches à l'opérateur en lui assurant une meilleure vision sur tous les éléments de la chaîne avec une interface HMI. Sur cette interface on présente aussi les alarmes nécessaires pour un bon fonctionnement de la chaîne.

Dans le premier chapitre on va présenter les automates programmables industriels, leur architecture, leurs langages de programmation et leur diagnostic. On va aussi expliquer le rôle de la supervision.

Dans le deuxième chapitre on va présenter TIA portal le logiciel de programmation de SIEMENS où on va décrire les différentes étapes essentielles pour créer un projet, comment choisir un API, comment le configurer, comment déclarer les entrées /sorties ...etc. afin d'introduire notre travail présenté au chapitre 4.

Nous avons réservé le chapitre 3 pour la présentation de la chaîne de gainage « BMN 120 » en décrivant tous ses équipements.

Dans Le quatrième chapitre on va décrire notre solution et on a présenté le cahier de charge détaillé, on a élaboré les Graficets et on a aussi réalisé des vues de supervision.

# **Chapitre 1 : Les automates programmables industriels**

## 1.1 Introduction :

L'automate programmable industriel (API) en anglais programmable logic controller (PLC) est le constituant principal de toute chaîne de production, avec une très grande flexibilité. On le trouve non seulement dans tous les secteurs de l'industrie, mais aussi dans les services. Dans l'industrie il fait la partie intégrante de la boucle de réglage, son rôle principale c'est de récolter les informations via ses interfaces d'entrées venantes des capteurs, les traiter pour prendre une décision qui sera transmise à la fin vers les actionneurs à travers les interfaces de sorties.

## 1.2 Historique :

Aux débuts des années 50 et jusqu'à les années 60 la technologie utilisée pour l'automatisation était la logique câblée où les composants de base étaient des relais électromécaniques, les circuits conçus portaient des milliers de relais.

Vers les 1960 la technologie des semi-conducteurs voie la lumière et les transistors sont apparus dans les automatismes sous forme des circuits intégrés, les automatismes devenus de plus en plus moins encombrants et on pouvait concevoir des fonctions plus complexes (le comptage, la temporisation ... etc.) et avec un coût décroissant.

Les API sont apparues aux États-Unis en 1969 où ils répondaient aux désirs de l'industrie des automobiles de développer des chaînes de fabrications automatisées. Les 2 premières firmes sont « Modicon » et « Allen\_Bradely ». Les premiers automates n'étaient pas rentables que pour des installations avec un certain degré de complexité, mais ceci n'a pas duré longtemps.

Aujourd'hui plusieurs APIs sont disponibles sur le marché, des nano-automates avec une dizaine d'entrées-sorties jusqu'aux grands automates avec des milliers d'entrées-sorties.

Les exigences de la complexité, de la sécurité et du confort nous ont obligés de passer de l'automatisme industriel vers l'informatique industrielle où la supervision et la commande à distance via des écrans tactiles modernes (interfaces HMI) sont devenues une nécessité.

## 1.3 Définition :

L'Automate Programmable Industriel est un appareil électronique programmable adapté à l'environnement industriel, qui réalise des fonctions d'automatisme pour assurer la Commande de pré actionneurs et d'actionneurs à partir d'informations issues des actionneurs [1].

« Appareil électronique qui comporte une mémoire programmable par un utilisateur automatique (et non informaticien) à l'aide d'un langage adapté, pour le stockage interne des instructions composant les fonctions d'automatisme comme par exemple :

- logique séquentiel et combinatoire
- temporisation, comptage, décomptage, comparaison
- calcul arithmétique
- réglage, asservissement, régulation ... etc.
- pour commander, mesurer et contrôler au moyen des modules d'entrées et de sorties (logiques, numériques ou analogiques) différentes sortes de machines ou de processus, en environnement industriel. » **Norme NFC 63-850**

## 1.4 La structure d'un système automatisé :

On peut décomposer chaque système automatisé en 2 parties [2] :

### 1.4.1 Partie commande :

C'est le centre de décision. Elle donne des ordres à la partie opérative et reçoit ses comptes rendus venus des capteurs. Ses ordres de fonctionnement sont transmis aux pré-actionneurs qui permettent de commander les actionneurs, où les pré-actionneurs assurent le transfert d'énergie entre la source et les actionneurs (la partie puissance).

Ces pré-actionneurs sont commandés par le bloc traitement des informations. Celui-ci reçoit les consignes du pupitre de commande (opérateur) et les informations de la partie opérative transmises par les capteurs / détecteurs. En fonction de ces consignes et de son programme de gestion des tâches (implanté dans l'automate programmable), elle va commander les pré-actionneurs et renvoyer des informations au pupitre de signalisation ou à d'autres systèmes de commande et/ou de supervision en utilisant un réseau et un protocole de communication.

### 1.4.2 Partie opérative (puissance) :

La partie opérative d'un système automatisé est le sous-ensemble qui effectue les actions physiques (déplacement, rotation, émission de lumière...), mesure des grandeurs physiques (température, vitesse, couple, humidité, luminosité...) et rend compte à la partie commande. Elle est généralement composée d'actionneurs, de capteurs, de pré-actionneurs, ... etc.

Plus simplement la partie opérative reçoit les ordres de la partie commande et les exécute.

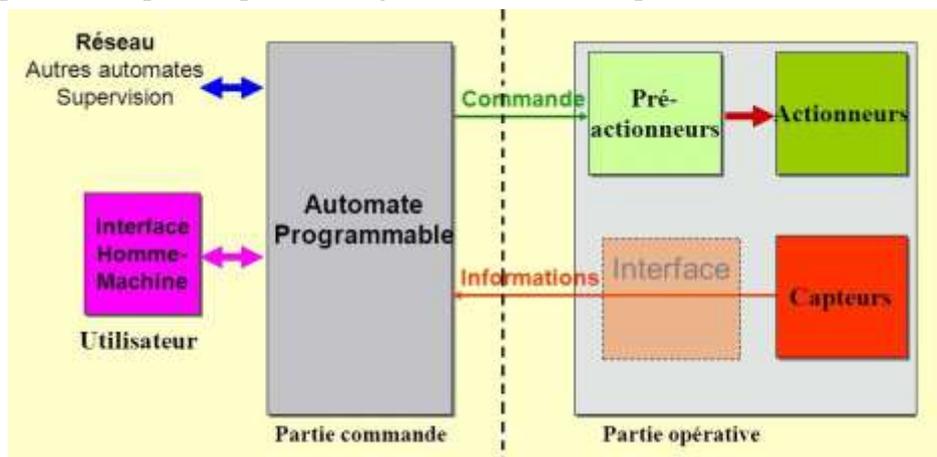


Figure 1-1 structure d'un système automatisé

## 1.5 Les types d'automate programmable industriel (API) :

Il y a plusieurs classifications des APIs distingués principalement par leur forme, leur taille, leur apparence, leur mode d'alimentation, leur puissance ou par leur langage de programmation. Généralement les automates peuvent être de type compact ou modulaire :

### 1.5.1 Les automates compacts :

Ils intègrent le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, sont des appareils avec un nombre fixe d'entrées-sorties digitales et analogiques.

Ils sont cependant extensibles par blocs jusqu'à environ 250 entrées-sorties. Ils sont principalement exploités pour des applications de complexité moyenne avec de la logique séquentielle et un traitement limité des fonctions analogiques. (Figure 1-2)



Figure 1-2 : S7-300 compact (S7-314C).

### 1.5.2 Les automates modulaires :

Le processeur, l'alimentation et les interfaces d'entrées/sorties résident dans des unités séparées (modules) et sont fixées sur un ou plusieurs racks contenant le « fond de panier » (bus plus connecteurs). Ils sont des machines rapides et puissantes qui travaillent avec des processeurs performants. Ce sont de véritables ordinateurs. Une CPU peut traiter plus de 8000 E/S. (**Figure 1-3**)



Figure 1-3: S7-300 modulaire (S7-317).

## 1.6 Architecture d'un API :

Il est constitué d'une unité centrale de traitement (CPU) qui comporte le microprocesseur, la mémoire et les entrées sorties du système.

Le CPU contrôle et exécute toutes les opérations de l'automate. Il est muni d'une horloge dont la fréquence entre 1 et 20 MHz, cette fréquence détermine la rapidité de fonctionnement de l'API. Au sein de l'API, toutes les informations sont transmises au moyen de signaux numériques, et les chemins par lesquels passent ces signaux sont appelé Bus. Au sens physique le bus est un ensemble de conducteurs sur lesquels circulent les signaux électriques (**Figure 1-4**).

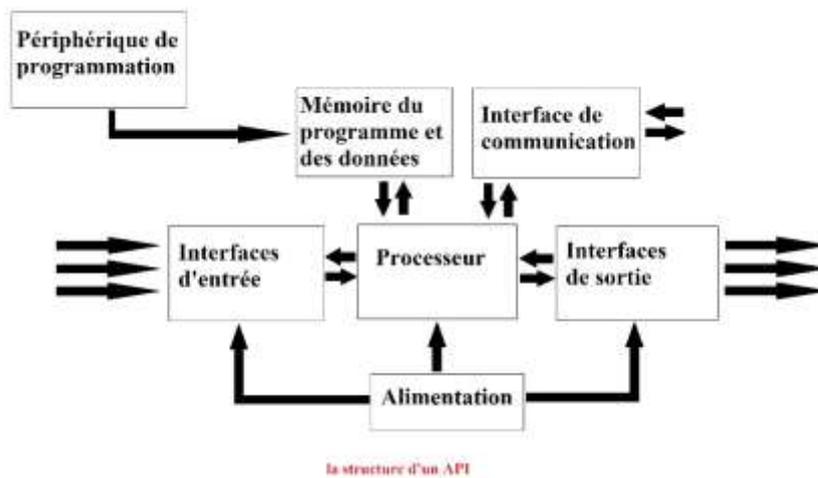


Figure 1-4: la structure d'un API.

### 1.6.1 Le processeur :

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

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

Il est organisé autour d'un certain nombre de registres, ce sont des mémoires rapides permettant la manipulation des informations qu'elles retiennent, ou leurs combinaisons avec des informations extérieures, les principaux registres existants dans un processeur sont :

#### 1.6.1.1 L'accumulateur :

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

#### 1.6.1.2 Le registre d'instruction :

Il reçoit l'instruction à exécuter et décode le code opération. Cette instruction est désignée par le pointeur.

#### 1.6.1.3 Le registre d'adresse :

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

#### 1.6.1.4 Le registre d'état :

C'est un ensemble de positions binaires décrivant, à chaque instant, la situation dans laquelle se trouve précisément la machine.

### **1.6.1.5 Les piles :**

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

### **1.6.2 Les mémoires :**

Un système à processeur est toujours accompagné d'un ou de plusieurs types de mémoires. Les automates programmables industriels possèdent pour la plupart les mémoires suivantes :

#### **1.6.2.1 Mémoire de travail :**

La mémoire de travail (mémoire vive) contient les parties du programme significatif pour son exécution. Le traitement du programme a lieu exclusivement dans la mémoire de travail et dans la mémoire système.

#### **1.6.2.2 Mémoire système :**

La mémoire système (mémoire vive) contient les éléments de mémoire que chaque CPU met à la disposition du programme utilisateur comme, par exemple, mémoire image des entrées, mémoire image des sorties, mémentos, temporisation et compteurs. La mémoire système contient, en outre, la pile des blocs et la pile des interruptions. Elle fournit aussi la mémoire temporaire allouée au programme (pille des données locales).

#### **1.6.2.3 Mémoire de chargement :**

La mémoire de chargement sert à l'enregistrement du programme utilisateur sans affectation de mnémotechnique ni de commentaires. La mémoire de chargement peut être soit une mémoire vive (RAM), soit une mémoire EPROM.

#### **1.6.2.4 Mémoire RAM non volatile :**

Zone de mémoire configurable pour sauvegarder des données en cas de défaut d'alimentation.

#### **1.6.2.5 Mémoire ROM :**

Contient le système d'exploitation qui gère la CPU.

### **1.6.3 Modules d'entrées/sorties :**

Les modules d'entrées et sorties assurent le rôle d'interface entre le procédé à commander et la CPU.

#### **1.6.3.1 Entrées/Sorties TOR :**

La gestion de ce type de variables constituant le point de départ historique des API et reste une de leurs activités majeures. Leur nombre est en général de 8,16, 24 ou 32 entrées/sorties, qui peuvent fonctionner :

- En continu : 24V, 48V.
- En alternatif : 24V, 48V, 100/120V, 210/240V

### 1.6.3.2 Entrées/Sorties analogiques :

Elle a pour rôle de fournir les tensions continues nécessaires aux composants avec de bonnes performances, notamment face aux microcoupures du réseau électrique qui constitue la source d'énergie principale. La tension d'alimentations peut être de 5V, 12V ou 24V.

D'autres alimentations peuvent être nécessaires pour les châssis d'extension et pour les modules entrées/sorties. Un onduleur est nécessaire pour éviter les risques de coupures non tolérées.

### 1.6.4 L'alimentation électrique :

Elle a pour rôle de fournir les tensions continues nécessaires aux composants avec de bonnes performances, notamment face aux microcoupures du réseau électrique qui constitue la source d'énergie principale. La tension d'alimentations peut être de 5V, 12V ou 24V.

D'autres alimentations peuvent être nécessaires pour les châssis d'extension et pour les modules entrées/sorties. Un onduleur est nécessaire pour éviter les risques de coupures non tolérées.

### 1.6.5 Les liaisons :

Elles s'effectuent :

- avec l'extérieur par des borniers, sur lesquels arrivent des câbles transportant les signaux électriques.
- avec l'intérieur par des bus, c'est les liaisons parallèles entre les divers éléments. la CPU utilise le Bus de données pour l'échange des données entre les composants .le bus d'adresse pour la transmission des adresses des emplacements permettant d'accéder aux données stockées .le Bus de contrôle pour les signaux qui concernent les actions de contrôle interne et le bus système set aux communication entre les ports d'entrées sorties et l'unité d'entrées sorties. (Figure 1-5)

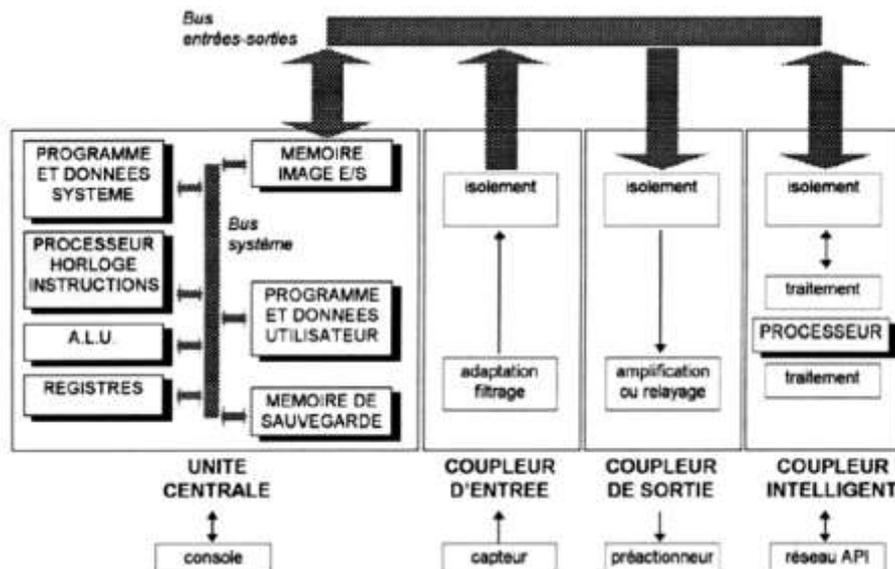


Figure 1-5: les différentes liaisons possibles d'un API [2].

### 1.6.6 Eléments auxiliaires :

On peut trouver plusieurs autres éléments qui aident au bon fonctionnement de l'automate, par exemple un ventilateur est indispensable dans les châssis comportant de nombreux modules ou dans le cas où la température ambiante est susceptible de devenir assez élevée. Le support

mécanique qu'il peut s'agir d'un rack est aussi un élément indispensable. On trouve aussi des indicateurs d'état concernant la présence de tension, la charge de la batterie, le bon fonctionnement de l'automate ...etc.

## 1.7 Les langages de programmation :

La norme **IEC 1131-3** définit cinq langages qui peuvent être utilisés pour la programmation d'applications d'automatisme. Les cinq langages sont : [3]

- SFC « sequential function chart », ou langage GRAFCET
- FBD « function block diagram », ou schéma par blocs
- LD « ladder diagram », ou schéma à relais
- ST « structured text », ou texte structuré
- IL « instruction list », ou liste d'instructions

## 1.8 Le diagnostic de l'API :

Pour tout équipement commandé par un API, la détection et la localisation des défauts constituent une exigence en matière de sécurité. Les défauts proviennent en grande majorité des capteurs, des actionneurs ou du câblage.

De nombreux API disposent de procédures d'analyse de défauts qui procèdent à des autotests et affichent les codes des défauts, parfois par un court message. Il existe plusieurs techniques de détection des défauts parmi eux : [4]

### 1.8.1 Vérification du minutage :

Le terme chien de garde désigne une vérification du minutage réalisée par l'API afin de contrôler une fonction en temps normal .si la fonction a demandé un temps plus long que la normale, un défaut est supposé et le temporisateur du chien de grade se déclenche activant une alarme et arrêtant l'API.

### 1.8.2 Dernière sortie activée :

Cette technique se fonde sur l'utilisation de témoins d'état qui indique la dernière sortie activée en cour d'un processus qui est arrêté. Ces témoins sont intégrés dans le programme de manière à être allumés lorsque les sorties se produisent. Ils indiquent donc toujours les dernières sorties qui sont activées.

### 1.8.3 La réplication :

C'est lorsque la sécurité est une question importante lors de l'apparition d'un défaut des contrôles peuvent être constamment mis en œuvre pour détecter les défauts. Une technique consiste à répliquer les vérifications, ce qui implique la duplication (2 fois ou plus pour chaque opération).

Une solution plus couteuse c'est de dupliquer tout le système API et comparer les résultats qu'ils produisent.

### 1.8.4 Vérification d'une valeur attendue :

Des erreurs logicielles peuvent être détectées en vérifiant si une valeur attendue est obtenue lorsque certaine entrées se produit .si la valeur attendue n'est pas obtenue un défaut est supposé.

## 1.9 La communication et les Réseaux industriels :

L'automate programmable industriel doit être connecté à d'autres matériels à processeur, et afin de pouvoir dialoguer avec les agents d'exploitation il faut donc se pencher sur ses liens avec son environnement et les fonctions qu'il doit assurer. Il existe plusieurs bus de terrain industriel qui sont universels ou limités (le bus est propre à l'entreprise), on note quelques protocoles de communication : Modbus, BusCAN, uni-Telway, CANopen, Profibus, Profinet, MPI, ...etc.

### 1.9.1 Le Profibus :

Profibus (Processus Field Bus) est le bus de terrain ouvert qui connaît un grand succès dans un large domaine d'application pour une production efficace et économique.

Profibus est un réseau en bus pour la communication de processus et de terrain dans les réseaux de cellules mettant en œuvre un petit nombre de stations et des appareils de terrain ainsi que pour la communication de données conforme à CEI 61158/EN 50 170.

#### 1.9.1.1 Profibus-FMS (Field bus Message Specification):

Parmi les premiers bus de terrain développés, il est utile pour les tâches hautement communicantes.

#### 1.9.1.2 Profibus-DP (Decentralized Peripherals):

Profibus (périphérie décentralisée) est un profil de communication Profibus. Il est optimisé pour la transmission rapide de données à temps critique au niveau du terrain, à des coûts de connexion minimaux. Il est plus récent que le Profibus FMS.

#### 1.9.1.3 Profibus-PA (Process Automation) :

Il est utilisé pour les équipements de mesure et surveillance, il est conçu pour les zones à risques d'explosions avec des techniques limitant les courants électriques.

### 1.9.2 Le Profinet :

Le Profinet est un standard de communication ouvert utilisé à l'automatisation industrielle actuelle, dans le cadre de Totally Integrated Automation (TIA) de SIEMENS, PROFINET principalement est le regroupement logique de :

- Profibus DP, le bus de terrain établi (l'ancien standard)
- Industriel Ethernet

Profinet IO, en tant que standard d'automatisation basé sur Ethernet de Profibus international, définit un modèle de communication, d'automatisation et d'ingénierie non propriétaire. Il utilise une technique de commutation permettant à chaque partenaire d'accéder au réseau à tout moment.

Grâce à la transmission simultanée de données de plusieurs partenaires, l'exploitation du réseau est bien plus effective. L'émission et la réception simultanées sont rendues possibles par l'exploitation de l'Ethernet.

Profinet IO est basé sur l'exploitation de l'Ethernet commuté en duplex intégral avec une bande passante de 100Mbit/s.

## 1.10 La Supervision :

Lorsque la complexité des processus augmente et que les machines et les installations doivent répondre à des spécifications de fonctionnalité toujours plus complexes, l'opérateur a besoin d'un maximum de transparence. Cette transparence s'obtient au moyen de l'Interface Homme-Machine (IHM).

Un Pupitre de commande et de supervision (**Figure 1-6**) constitue l'interface entre l'homme (opérateur) et le processus (machine/installation). Le contrôle proprement dit du processus est assuré par le système d'automatisation, la communication entre l'API et le pupitre est assurée par un bus de terrain (protocole de communication) permet ce qui en haut.

Il a pour objectif :

- Représentation du processus (les vues)
- Commande du processus
- Archivage des valeurs de processus et d'alarmes
- Gestion des paramètres



*Figure 1-6: pupitres tactiles SIEMENS.*

## 1.11 Conclusion :

Dans ce chapitre, on a présenté les automates programmables industriels (APIs), leur architecture, les modules indispensables comme les modules d'entrées sorties, les langages de programmation et des notions particulières très utilisées dans l'industrie comme le diagnostic et la communication.

## **Chapitre 2 : TIA portal, logiciel de programmation et supervision**

## 2.1 Introduction :

Dans ce chapitre on va faire une représentation générale sur le logiciel de programmation et supervision (TIA portal). On résume toutes les taches possibles à manipuler dans notre projet.

Le lecteur de ce mémoire (ex ce chapitre) vous pouvez apprendre la programmation ou la création des projets d'automatisation sous TIA portal.

## 2.2 Présentation générale du logiciel :

### 2.2.1 Définition :

La plateforme **Totally Integrated Automation Portal** est le nouvel environnement de travail Siemens qui permet de mettre en œuvre des solutions d'automatisation avec un système d'ingénierie intégré comprenant les logiciels SIMATIC STEP 7 et SIMATIC Win CC.

### 2.2.2 Utilisation du logiciel :

Totally Integrated Automation répond de façon optimale à toutes les exigences tout en étant ouvert aux standards internationaux et aux systèmes d'autres constructeurs. Avec ses six propriétés système (ingénierie, communication, diagnostic, sûreté de fonctionnement (Safety), sécurité des données (Security) et robustesse), Totally Integrated Automation prend en charge l'ensemble du cycle de vie d'une machine ou d'une installation.

### 2.2.3 Structure de l'interface utilisateur du TIA Portal :

Il y a trois vues différentes pour chaque projet d'automatisation et on peut basculer d'une vue à l'autre au moyen d'un lien.

#### 2.2.3.1 Vue de portail :

La vue du portail (**Figure2-2**) offre une vue orientée sur les tâches des outils. Vous pouvez y décider rapidement ce que vous souhaitez faire et appeler l'outil requis pour la tâche correspondante.



Figure 2-1: vue de portail [5]

1 : Portails pour les différentes tâches, Les portails mettent à disposition les fonctions élémentaires requises par chaque type de tâche. Les portails proposés dans la vue du portail dépendent des produits installés.

2 : Actions correspondant au portail sélectionné, en fonction du portail sélectionné, les actions que vous pouvez y exécuter vous sont proposées.

3 : Fenêtre de sélection correspondant à l'action sélectionnée, la fenêtre de sélection disponible dans chaque portail. Son contenu s'adapte à la sélection en cours.

4 : Le lien « **Vue du projet** » vous permet de basculer vers la vue de projet.

5 : Affichage du projet actuel ouvert, fournit des informations sur le projet actuellement ouvert.

### 2.2.3.2 Vue du projet :

La vue du projet (**Figure 2-3**) correspond à une vue structurée de l'ensemble des composants du projet.

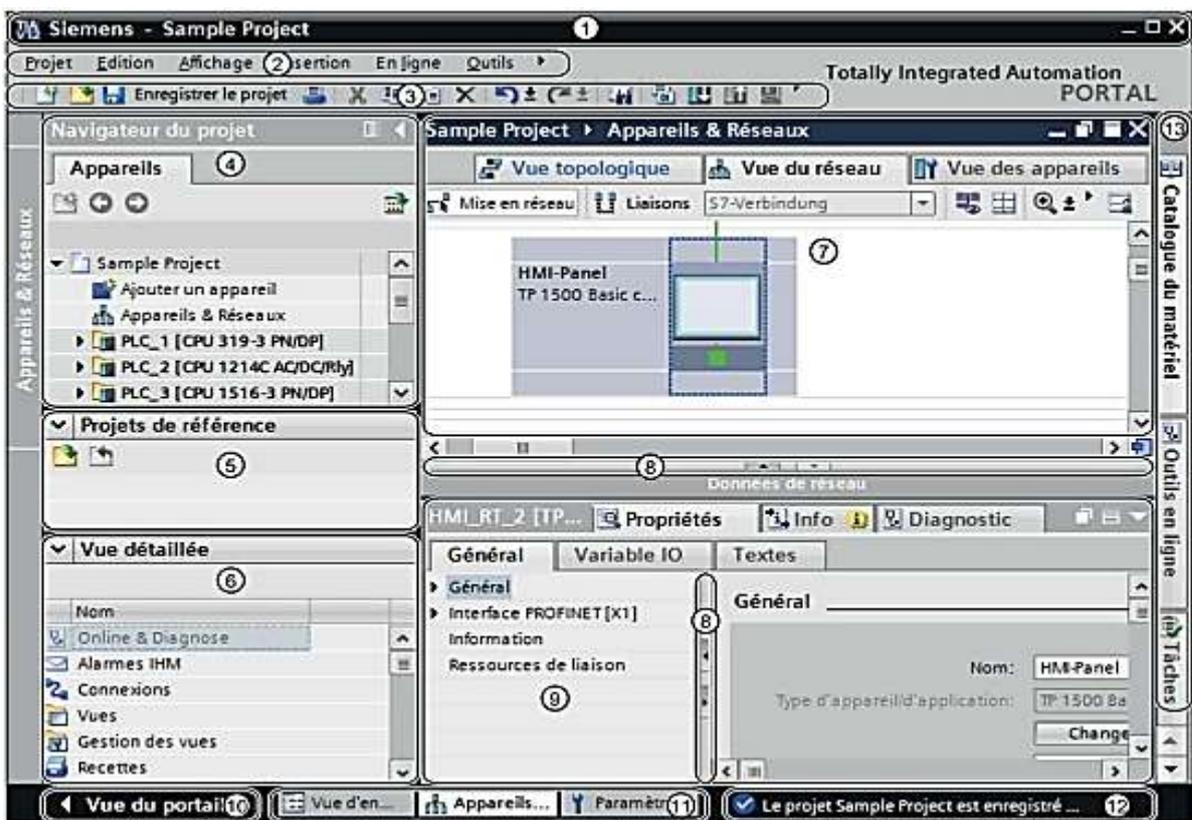


Figure 2-2: vue de projet [5]

1 : la barre de titre affiche le nom du projet.

2 : la barre des menus contient toutes les commandes dont vous avez besoin pour réaliser votre tâche.

3 : La barre d'outils met à votre disposition des boutons vous permettant d'exécuter les commandes les plus fréquemment utilisées.

4 : Le navigateur du projet vous permet d'accéder à tous les composants et les données du projet.

**5** : dans la palette « Projets de référence » on peut ouvrir des projets en plus du projet actuel, ces projets de référence sont ouverts en lecture seule et ne sont pas éditables. On peut faire glisser les objets d'un projet de référence dans notre projet en cours et les y éditer.

**6** : La vue détaillée affiche certains contenus d'un objet sélectionné dans la fenêtre de vue d'ensemble ou dans le navigateur de projet, il peut s'agir de listes de textes ou de variables.

**7** : La zone de travail affiche les objets qu'on ouvre afin de les éditer. Il s'agit des éditeurs et vues ou des tables.

**8** : Les Barres de fractionnement séparent les différents éléments de l'interface logicielle et les flèches situées sur ces barres de fractionnement permettent d'afficher ou de masquer les éléments de l'interface attenants.

**9** : La fenêtre d'inspection affiche des informations supplémentaires sur l'objet sélectionné ou sur les actions exécutées.

**10** : Le lien « Vue du portail » nous permet de basculer dans la vue du portail.

**11** : La barre des éditeurs affiche les éditeurs ouverts.

**12** : La barre d'état visualise la progression des processus qui se déroulant actuellement en arrière-plan, on peut interrompre les processus d'arrière-plan par le bouton à côté de la barre de progression.

**13** : Les « Task Cards » disponibles figurent dans une barre au bord droit de l'écran, on peut à tout moment ouvrir ou fermer cette barre, les Task Cards qui sont proposées dépendent des produits installés.

### **2.2.3.3 Vue des bibliothèques :**

La vue des bibliothèques affiche une vue d'ensemble des éléments de la bibliothèque de projet et des bibliothèques globales ouvertes. Vous pouvez basculer dans la vue des bibliothèques à l'aide de la Task Card « Bibliothèques ».

## **2.3 Créer et gérer des projets :**

### **2.3.1 Création d'un nouveau projet :**

Pour créer un nouveau projet, procédez de la manière suivante :

- 1) Choisissez la commande « Nouveau » dans le menu « Projet » (La boîte de dialogue « Créer un projet » s'ouvre).
- 2) Entrez le nom et le chemin souhaités pour le projet ou utilisez les données proposées.
- 3) Cliquez sur le bouton « Créer ». Le nouveau projet est créé et affiché dans le navigateur du projet, voir (**Figure 2-4**)

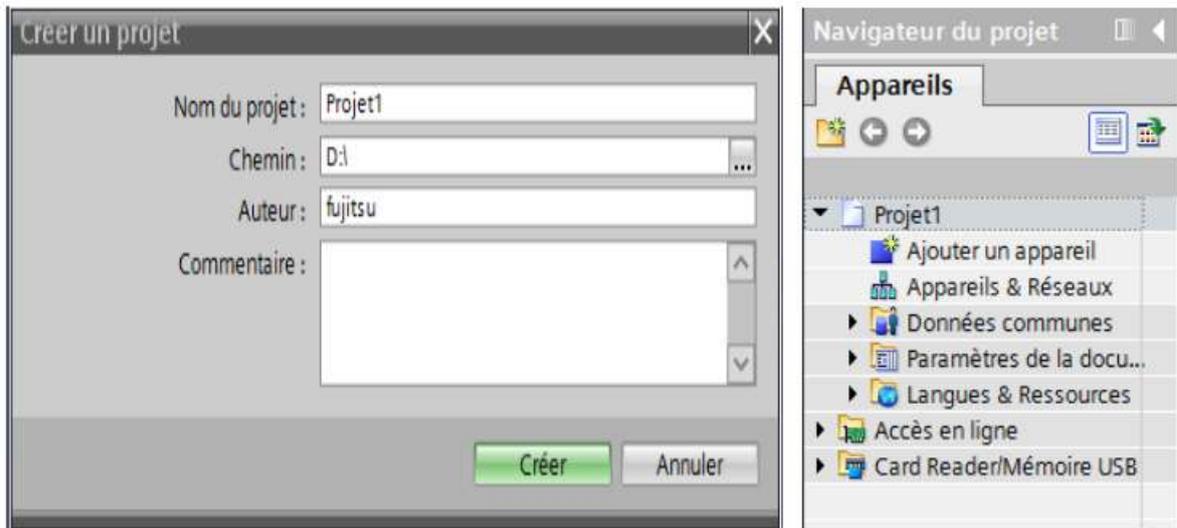


Figure 2-3: création d'un nouveau projet

### 2.3.2 Ouvrir des projets :

Tous les projets issus de la version actuelle et de versions antérieures peuvent être ouverts dans le TIA Portal. Pour ouvrir un projet existant, procédez de la manière suivante :

- 1) Choisissez la commande « Ouvrir » dans le menu « Projet » (La boîte de dialogue « Ouvrir le projet » s'ouvre et la liste des projets utilisés en dernier s'affiche).
- 2) Sélectionnez un projet dans la liste et cliquez sur « Ouvrir ».
- 3) Si le projet souhaité ne figure pas dans la liste, cliquez sur le bouton « Parcourir », naviguez jusqu'au dossier de projet souhaité et ouvrez le fichier du projet.

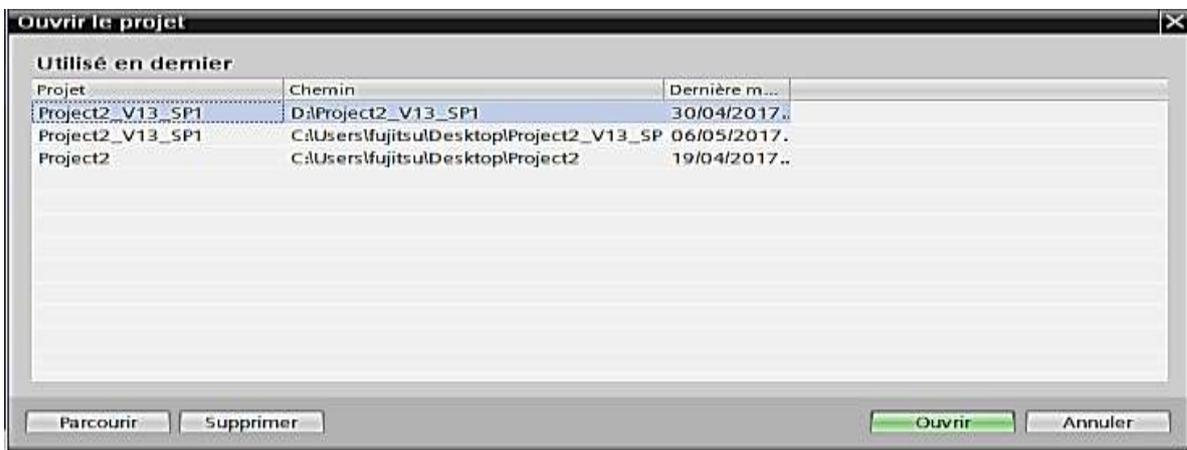


Figure 2-4: ouvrir d'un projet existant

Les projets se trouvant au format de projet actuel s'ouvrent dans la vue de projet. Si vous avez choisi un projet issu d'une version antérieure de TIA Portal, la boîte de dialogue « Mettre à niveau le projet » s'ouvre. Pour voir les versions de produit de TIA portal compatible à TIA V13 SP1 Voir (Annexe).

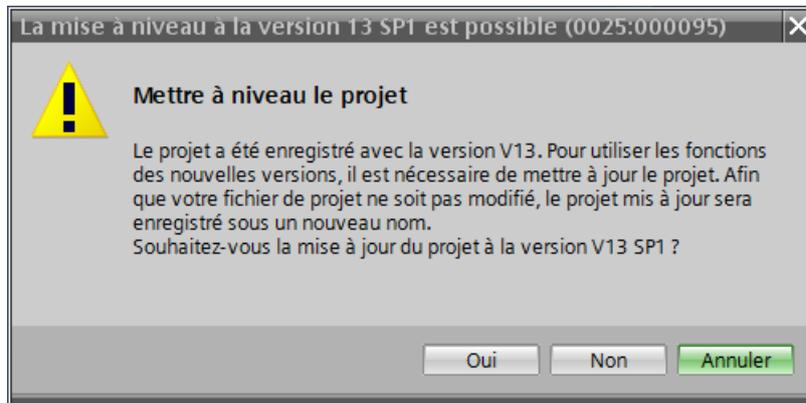


Figure 2-5 : la fenêtre de mise à niveau

### 2.3.3 Affichage des propriétés du projet :

Pour afficher les propriétés du projet, procédez de la manière suivante :

- 1) Sélectionnez le projet ouvert dans le navigateur du projet.
- 2) Choisissez la commande « Propriétés » dans le menu contextuel du projet (le dialogue avec les propriétés du projet s'ouvre).
- 3) Sélectionnez les propriétés de projet souhaitées dans la navigation d'écran pour les afficher.

Voici les points appartenant aux propriétés : [5]

- **Métadonnées du projet** : Ici, vous pouvez vous procurer par exemple des informations sur la date de création, sur l'auteur, sur le répertoire de sauvegarde, sur la taille du projet, sur les remarques de droit d'auteur, sur les langues projet, etc.
- **Historique du projet** : L'historique du projet contient un aperçu des éléments importants dans le cycle de vie d'un projet. Vous pouvez par exemple y trouver la version de TIA Portal avec laquelle un projet a été créé et savoir s'il a été entre-temps converti en une autre version, etc.
- **Support Packages dans le projet** : Une vue d'ensemble des logiciels supplémentaires requis pour le traitement de tous les appareils du projet s'affiche. On y trouve également la liste des fichiers GSD installés (fichiers de description des appareils pour d'autres appareils du catalogue du matériel).
- **Produits logiciels du projet** : Vous pouvez faire afficher une vue d'ensemble de tous les produits logiciels installés et nécessaires pour le projet.

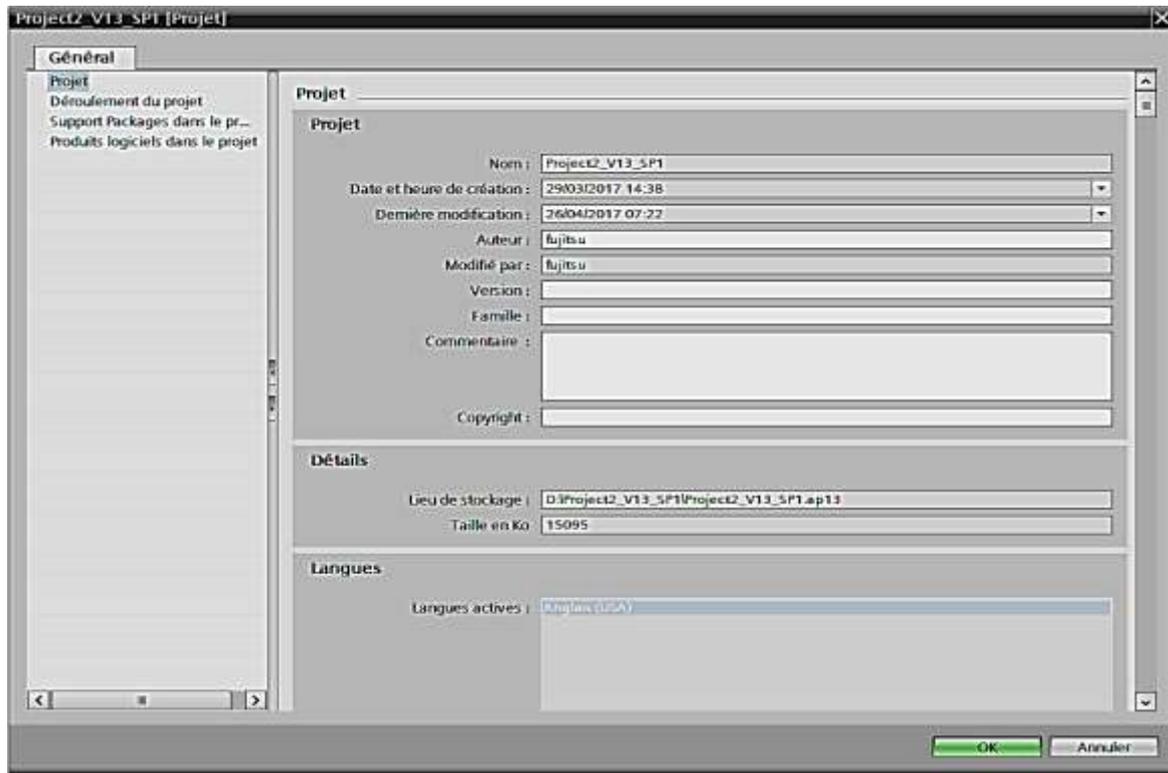


Figure 2-6 : la fenêtre où on gère les propriétés d'un projet

## 2.4 Editer des données de projet :

### 2.4.1 Compilation des données du projet :

Lors de la compilation, les données du projet sont converties de sorte à pouvoir être lues par l'appareil. Les données de configuration matérielle et les données du programme peuvent être compilées ensemble ou séparément.

Les données suivantes du projet doivent être compilées avant le chargement :

- Données matérielles du projet, par ex. les données de configuration des appareils ou les réseaux et liaisons.
- Données logicielles du projet, par ex. les blocs de programme ou les vues de processus.

Pour compiler les données du projet, procédez de la manière suivante :

- 1) Dans le navigateur du projet, sélectionnez les appareils dont vous voulez compiler les données de projet.
- 2) Choisissez l'option souhaitée dans le sous-menu « Compiler » du menu contextuel. Lorsque vous compilez les données du projet, différentes possibilités s'offrent à vous en fonction de l'appareil :
  - Matériel et logiciel (modifications seulement)
  - Matériel (modifications seulement)
  - Matériel (compilation complète)
  - Logiciel (modifications seulement)
  - Logiciel (compilation complète des blocs)
  - Logiciel (réinitialisation de la réserve de mémoire)

- 3) Les données du projet sont compilées, Donc vous pouvez contrôler la réussite de la compilation dans la fenêtre d'inspection, sous « Info > Compiler ».

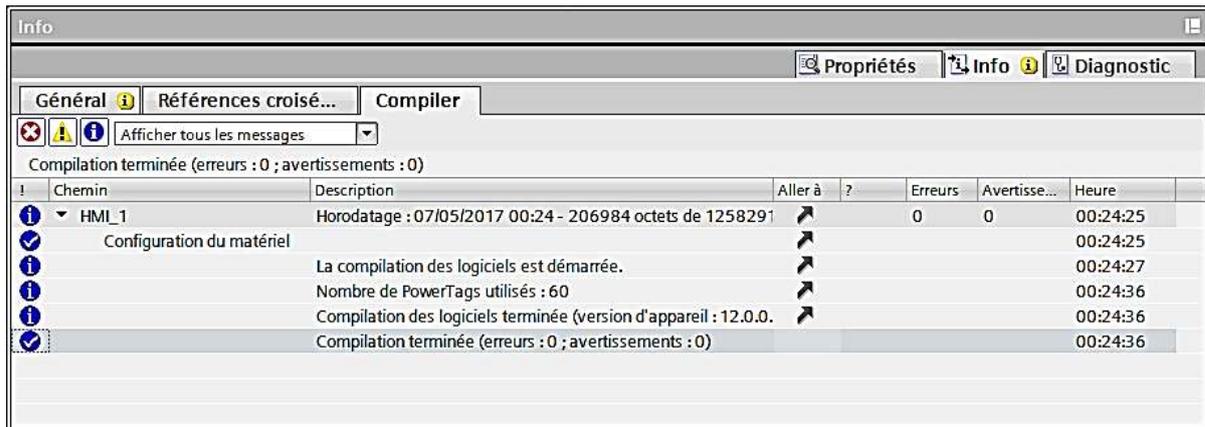


Figure 2-7 : la fenêtre qui affiche l'état de compilation

## 2.4.2 Charger des données de projet :

Pour configurer votre système d'automatisation, il est nécessaire de charger dans les appareils raccordés les données de projet générées en mode hors ligne. Ces données de projet sont générées par exemple durant la configuration du matériel, des réseaux et des liaisons, au cours de la programmation du programme utilisateur.

Lors du premier chargement, toutes les données du projet sont chargées. Par la suite, seules les modifications sont chargées. Vous pouvez charger les données de projet dans des appareils et sur des cartes mémoires.

### 2.4.2.1 Charger des données de projet dans un appareil :

Si Les données du projet sont cohérentes et chaque appareil à charger est accessible via un accès en ligne alors, vous pouvez charger les données du projet dans les appareils sélectionnés. Procédez de la manière suivante :

- 1) Sélectionnez un ou plusieurs appareils dans le navigateur du projet.
- 2) Cliquez avec le bouton droit de la souris sur un objet sélectionné, Le menu contextuel s'ouvre.
- 3) Choisissez l'option souhaitée dans le sous-menu « Charger dans l'appareil » du menu contextuel.
- 4) Contrôlez les messages dans la boîte de dialogue « Aperçu de chargement ».
- 5) Cliquez sur le bouton « Charger », Dès que le chargement est possible, le bouton « Charger » est actif.
- 6) Cliquez sur le bouton « Terminer ». Les données du projet sélectionnées ont été chargées dans les appareils.

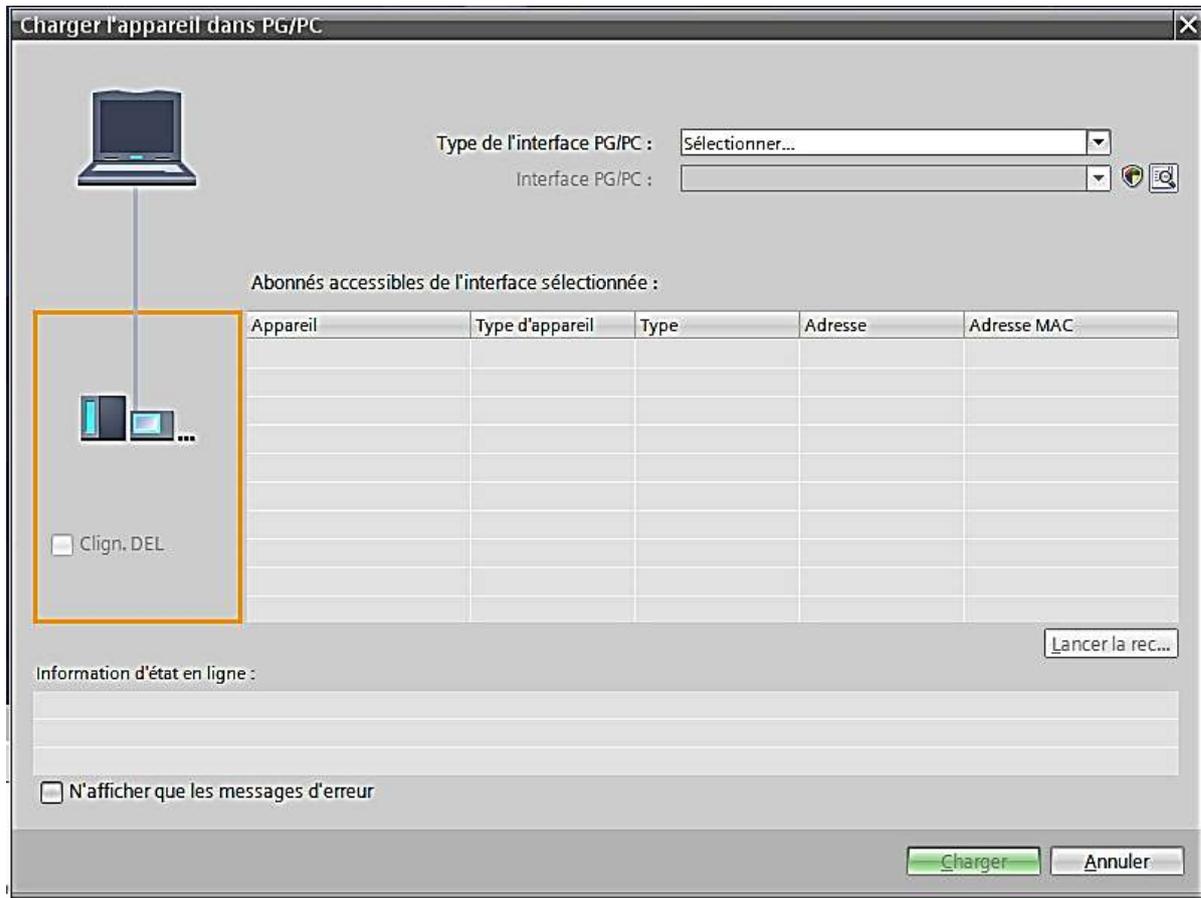


Figure 2-8 : la fenêtre de chargement

Voici les options disponibles pour l'objet que vous désirez charger :

- **Matériel et logiciel (modifications seulement) :** La configuration matérielle et le logiciel sont tous deux chargés vers la cible en cas de divergences entre les versions en ligne et hors ligne.
- **Configuration matérielle :** Seule la configuration matérielle est chargée dans la cible.
- **Logiciel (modifications seulement) :** Seuls les éléments qui sont différents en ligne et hors ligne sont chargés dans la cible.

#### 2.4.2.2 Charger les données de projet d'un appareil :

Si un objet est ouvert et Si La configuration matérielle et le logiciel devant être chargés doivent être compatibles avec TIA Portal. Vous disposez deux possibilités pour le chargement de données de projet d'un appareil vers votre projet.

- **Charger l'appareil comme nouvelle station :**

Afin de charger l'appareil complet dans votre projet, procédez comme suit :

- 1) Sélectionnez le nom de projet dans le navigateur du projet.
- 2) Dans le menu « En ligne » choisissez la commande « Charger l'appareil comme nouvelle station (matériel et logiciel) », La boîte de dialogue « Charger l'appareil dans PG/PC » s'ouvre.

- 3) Sélectionnez le type d'interface que vous désirez utiliser pour la procédure de chargement et sélectionnez l'interface qui doit être utilisée
- 4) Dans la table des abonnés accessibles sélectionnez l'abonné dont vous désirez charger les données de projet.
- 5) Cliquez sur « Charger ».

- **Charger les données de projet d'un appareil :**

Pour charger uniquement les données de projet d'un appareil dans votre projet, veuillez procéder comme suit :

- 1) Etablissez une liaison en ligne avec l'appareil à partir duquel vous souhaitez charger les données du projet.
- 2) Sélectionnez l'appareil dans le navigateur du projet.
- 3) Choisissez la commande « Charger depuis l'appareil (logiciel) » dans le menu « En ligne », La boîte de dialogue « Aperçu pour le chargement de l'appareil » s'ouvre.
- 4) Contrôlez les messages dans la boîte de dialogue « Aperçu pour le chargement de l'appareil ».
- 5) Cliquez sur le bouton « Charger de l'appareil » (Dès que le chargement est possible, le bouton « Charger de l'appareil » devient actif), la procédure de chargement s'exécute.

### 2.4.3 *Concept de protection des données du projet :*

Vous avez la possibilité de protéger vos données du projet contre des accès non autorisés. Il s'agit p. ex. de :

- Protection d'accès pour appareils.
- Protection contre la copie et l'affichage d'objets.
- Limitations pour la sortie d'objets avec protection Know-How.

## 2.5 **Utilisation de projets de référence :**

Vous avez la possibilité d'ouvrir des projets comme référence en plus du projet actuel. Vous pouvez utiliser ces projets de référence comme suit :

- Vous pouvez faire glisser des objets individuels d'un projet de référence dans le projet actuel et les y éditer.
- Vous pouvez ouvrir en lecture seule des objets donnés, par exemple des blocs de code, d'un projet de référence. Cela n'est toutefois pas possible pour tous les éléments.
- Vous pouvez comparer des appareils du projet de référence avec des appareils du projet actuel à l'aide d'une comparaison hors ligne/hors ligne.

Pour ouvrir un projet de référence, procédez de la manière suivante :

- 1) Cliquez sur « Ouvrir projet de référence » dans la barre d'outils de la palette « Projets de référence » dans le navigateur de projets, la boîte de dialogue « Ouvrir un projet de référence » s'ouvre.
- 2) Naviguez jusqu'au dossier de projet souhaité et ouvrez le fichier de projet.

- 3) Cliquez sur « Ouvrir ». Le projet sélectionné s'ouvre en lecture seule en tant que projet de référence.

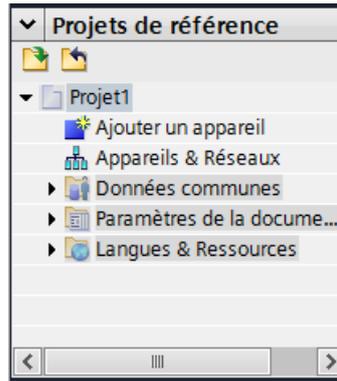


Figure 2-9 : la fenêtre de projet de référence sous TIA portal

## 2.6 Simuler des appareils :

TIA Portal vous permet de réaliser et de tester le matériel et le logiciel du projet dans un environnement simulé. La simulation s'exécute directement sur la PG/le PC. C'est pourquoi aucun matériel supplémentaire n'est requis. Procédez de la manière suivante pour démarrer la fonction de simulation :

- 1) Sélectionnez l'appareil que vous désirez simuler, par ex. dans le navigateur du projet.
- 2) Choisissez la commande « Simulation > Démarrer » dans le menu « En ligne ».

## 2.7 Edition des appareils et réseaux :

L'éditeur de matériels et de réseaux est l'environnement de développement intégré pour la configuration, le paramétrage et la mise en réseau des appareils et des modules. Dans la navigation du projet, double-cliquez sur la commande « Appareils et réseaux » pour ouvrir l'éditeur de matériels et de réseaux.

L'éditeur de matériels et de réseaux vous propose trois vues différentes de votre projet. Vous pouvez à tout moment basculer entre ces trois vues, selon que vous voulez créer et éditer des appareils et des modules individuellement, des réseaux et des configurations d'appareils complets ou la structure topologique de votre projet.

Vous sélectionnez dans le catalogue du matériel les appareils et les modules requis pour votre système d'automatisation et les insérez dans la vue des appareils, des réseaux ou topologique.

La fenêtre d'inspection contient des informations sur l'objet en cours de sélection. Vous pouvez y modifier les paramètres de l'objet sélectionné.

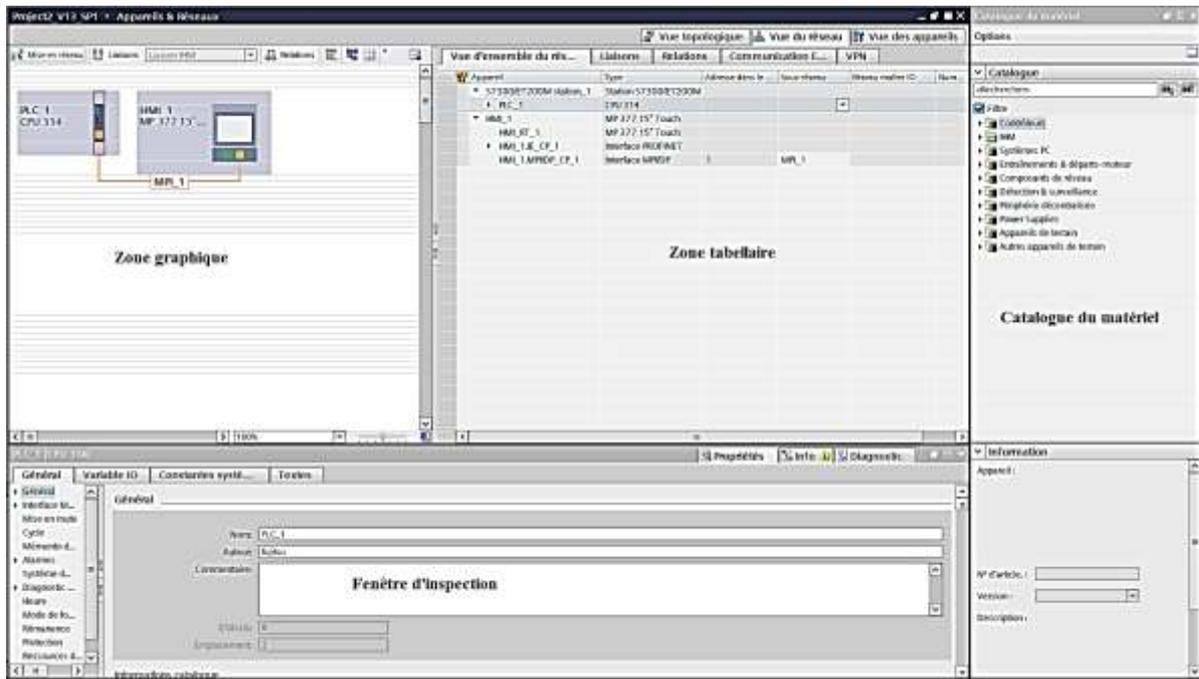


Figure 2-10 : la fenêtre de l'éditeur des appareils et réseaux

Vous pouvez Configurer et paramétrer les appareils et Mettre les appareils en réseau dans la vue de réseau.

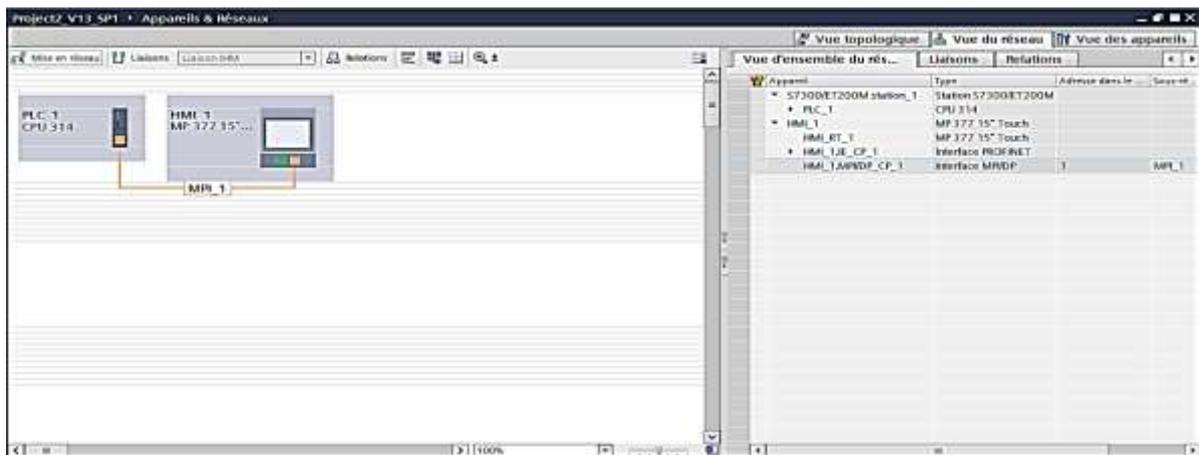


Figure 2-11 : la vue du réseau

Vous pouvez Configurer et paramétrer les appareils et les modules dans la vue des appareils.

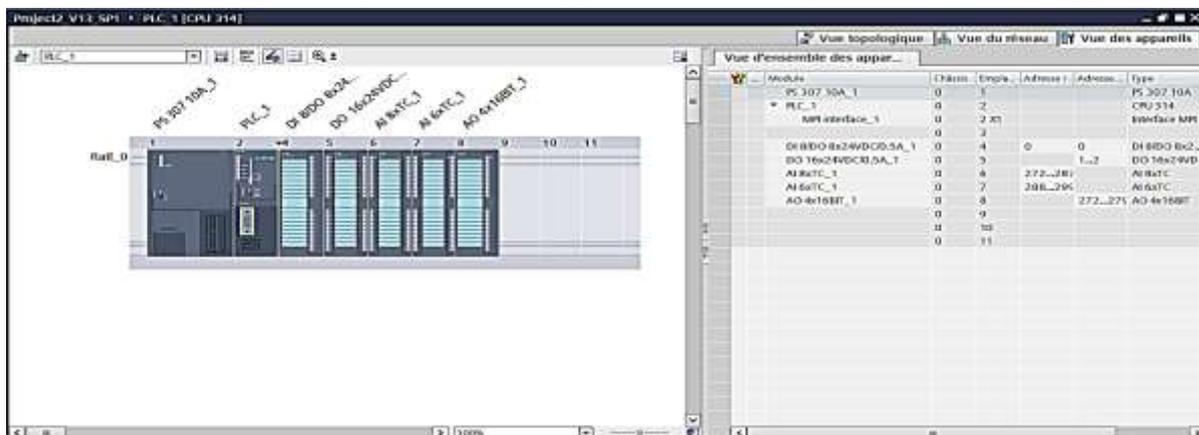


Figure 2-12 : la vue des appareils

### 2.7.1 Configuration des appareils :

Pour créer un système d'automatisation, vous devez configurer, paramétrer et relier les différents composants matériels. Au démarrage du système d'automatisation, la CPU compare la configuration logicielle prévue avec la configuration réelle de l'installation. De cette manière, il est possible de détecter et de signaler immédiatement les erreurs éventuelles.

#### 2.7.1.1 Utilisation des châssis :

Pour affecter un module à un appareil, vous avez besoin d'un châssis, vous fixez les modules sur le châssis et vous les reliez à la CPU. Lorsque vous ajoutez un appareil dans la vue de réseau, cela entraîne la création automatique d'une station et d'un châssis adapté à l'appareil sélectionné. Dans la vue de l'appareil, le châssis est affiché avec les emplacements disponibles. Le nombre d'emplacements disponibles dépend du type de l'objet.

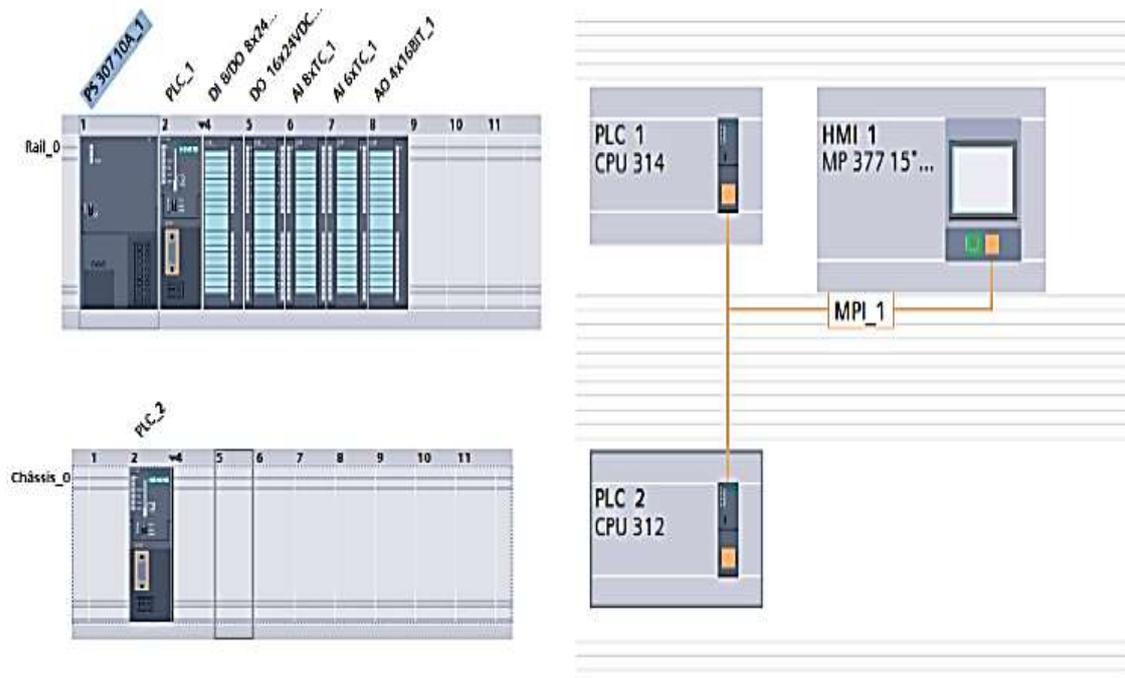


Figure 2-13 : exemple d'utilisation des châssis

#### 2.7.1.2 Ajouter un appareil à la configuration matériel :

Pour ajouter un appareil pouvant être mis en réseau à la configuration matérielle, glisser-déposer à partir du catalogue du matériel dans la vue de réseau ou de la vue topologique.

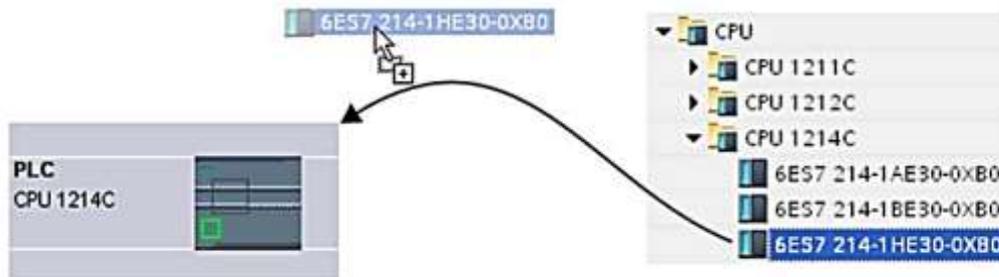


Figure 2-14 : insertion d'un nouvel appareil

Il y a d'autres possibilités pour ajouter un appareil comme :



enficher un module sur un châssis dans la vue des appareils, vous disposez des options suivantes :

- Pour un emplacement libre valide, double clic sur un module dans le catalogue du matériel.
- Glisser-déposer à partir du catalogue du matériel sur un emplacement libre valide dans la zone graphique ou tabellaire.

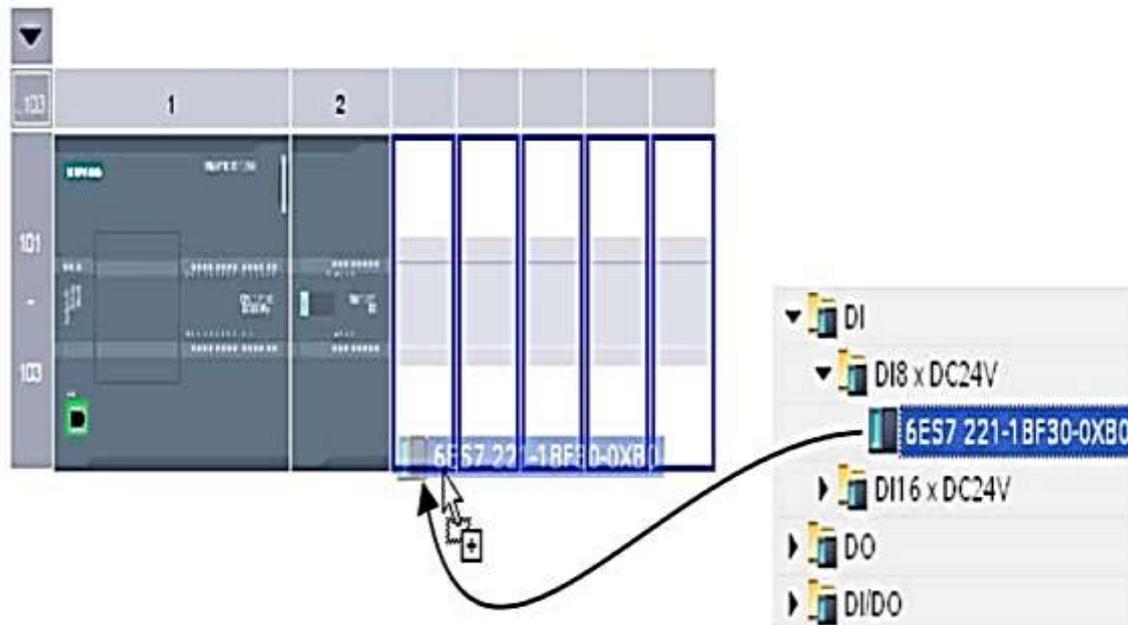


Figure 2-16 : insertion d'un module

Une fois qu'un module est enfiché sur un châssis avec une CPU déjà enfichée, les zones d'adresse sont automatiquement vérifiées. Vous pouvez copier (déplacer, supprimer et remplacer) des composants matériels dans la vue des appareils.

#### 2.7.1.4 Editer les propriétés et les paramètres :

Une fois que vous avez enfiché les composants matériels sur le châssis, vous pouvez éditer les propriétés par défaut dans la vue des appareils et de réseau, p.ex. Les paramètres et les adresses.

Pour modifier les propriétés et les paramètres des composants, procédez comme suit :

- 1) Dans la représentation graphique, sélectionnez la CPU, le module, le châssis ou l'interface à éditer.
- 2) Définissez les paramètres de l'objet sélectionné.

Dans la fenêtre d'inspection, les différents paramètres possibles sont disponibles sous « Propriétés ».

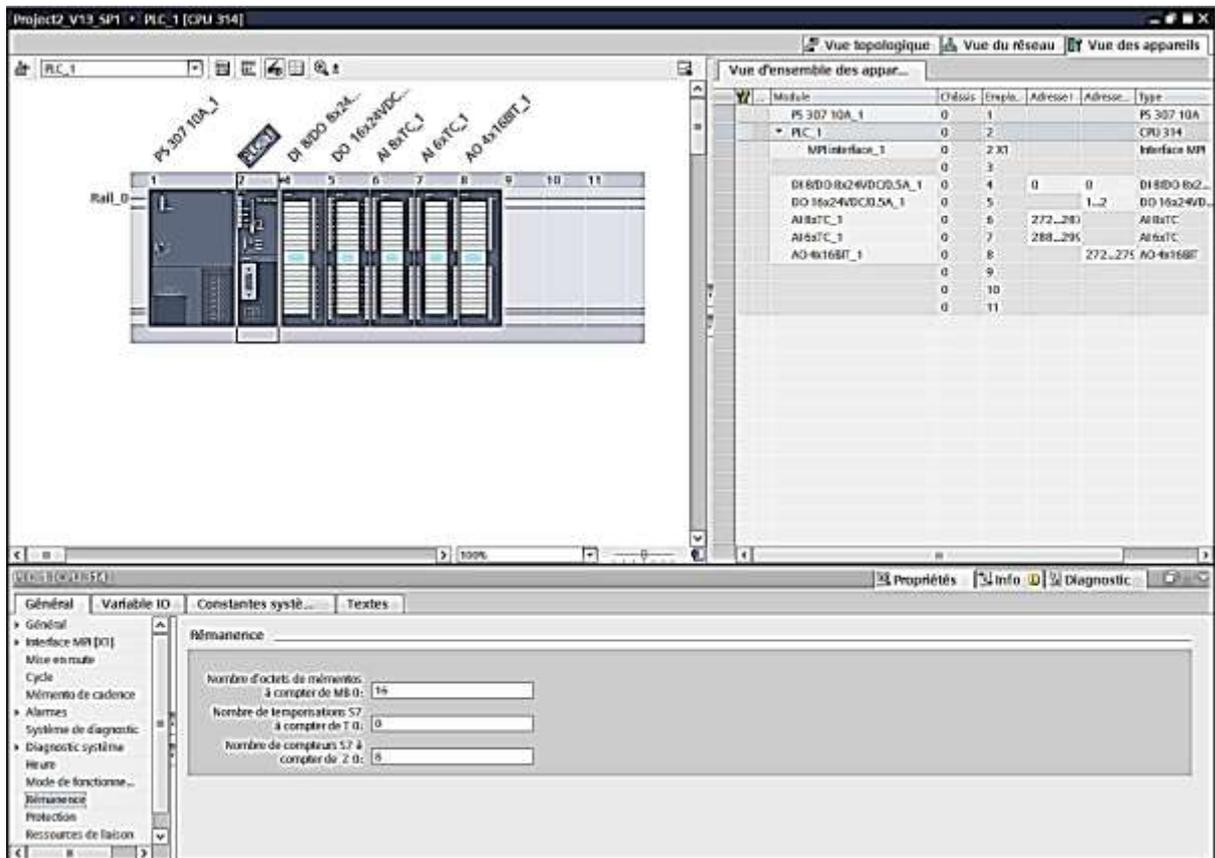


Figure 2-17 : Edition des propriétés dans la vue des appareils

Les adresses d'entrée et de sortie actuellement utilisées peuvent s'afficher sous forme de tableau dans la vue d'adresse. La vue d'adresse se trouve dans la fenêtre d'inspection de la CPU sous « Propriétés ».

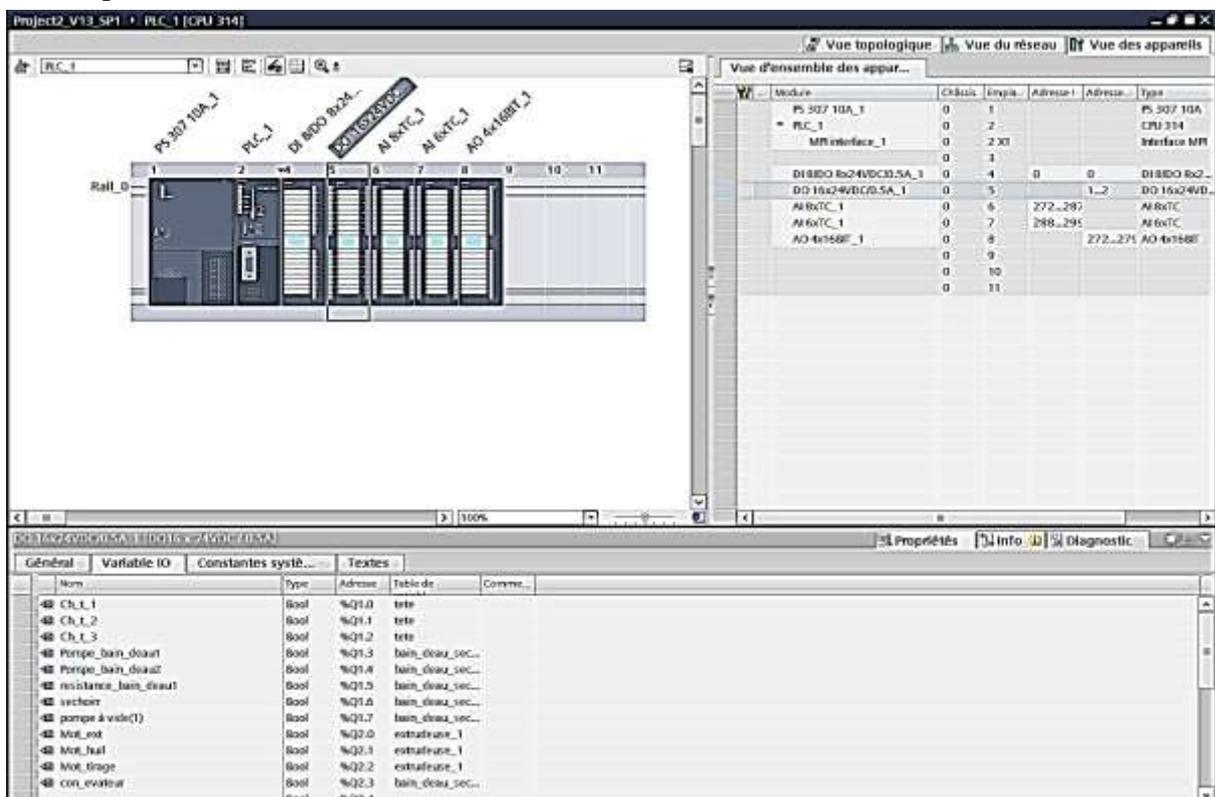


Figure 2-18 : voir les variables d'E/S affecté à un module

### 2.7.2 Configuration des réseaux :

Dans le projet, vous pouvez créer et mettre en réseau des appareils utilisant des composants aptes à la communication. Pour la mise en réseau des appareils, vous disposez des possibilités suivantes :

- 1) Vous raccordez entre elles les interfaces des composants aptes à la communication. De cette manière, vous créez un nouveau sous-réseau adapté au type de l'interface.
- 2) Vous raccordez l'interface des appareils aptes à la communication avec un nouveau sous-réseau ou avec un sous-réseau existant.
- 3) Vous configurez les liaisons via la configuration graphique des liaisons, les liaisons manquantes sont détectées et créées automatiquement ou par dialogue interactif.

En fonction des différentes tâches assignées aux appareils ou en raison de l'extension de l'installation, il peut être nécessaire d'exploiter plusieurs sous-réseaux. Ces sous-réseaux sont gérés dans un projet.

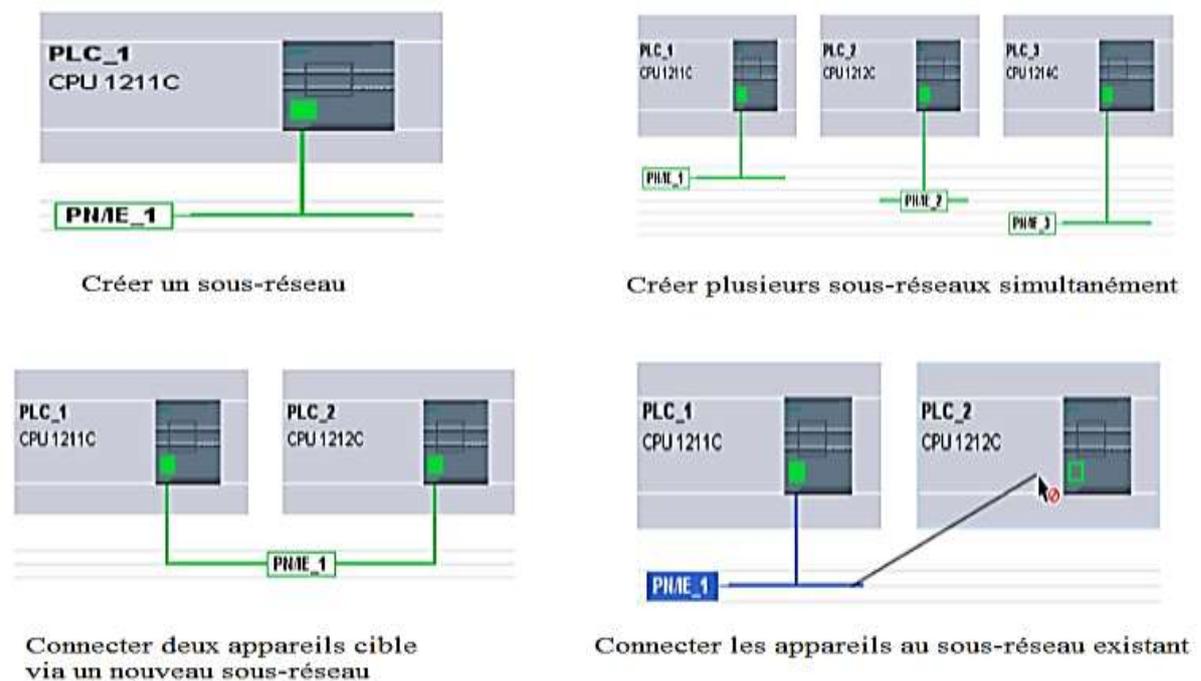


Figure 2-19 : configuration des réseaux

#### 2.7.2.1 Exemple (Configuration du réseau MPI) :

Pour les appareils avec une interface MPI, tous les participants à un sous-réseau doivent posséder des adresses de réseau différentes.

Les CPU avec adresses MPI sont livrées avec l'adresse MPI par défaut 2. Etant donné que vous ne pouvez utiliser cette adresse qu'une seule fois dans le sous-réseau MPI, vous devez modifier celle par défaut dans toutes les autres CPU. Règles pour l'attribution d'adresse MPI :

- Attribuez les adresses MPI dans l'ordre croissant.
- Réservez l'adresse MPI 0 pour une PG.

- Vous pouvez relier jusqu'à 126 abonnés (adressables) les uns aux autres dans un sous-réseau MPI (jusqu'à 8 abonnés pour une vitesse de transmission de 19,2 kbit/s).
- Toutes les adresses MPI d'un sous-réseau MPI doivent être différentes.

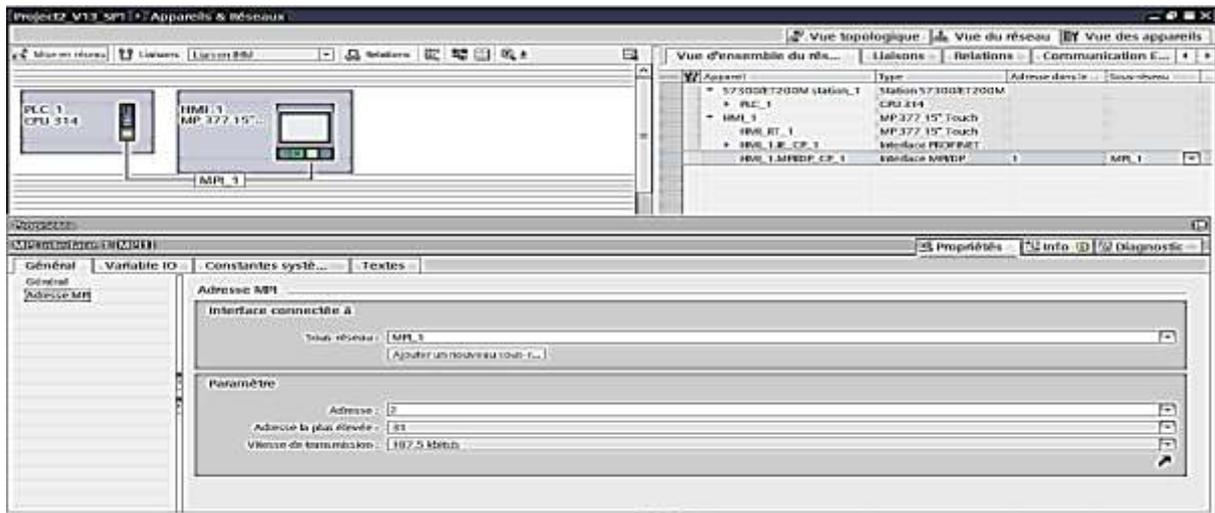


Figure 2-20 : configuration et paramétrage d'un réseau MPI

### 2.7.3 Diagnostiquer les appareils et les réseaux :

Le diagnostic des appareils et réseaux est une partie très important de l'éditations des projets sous TIA Portal, dont il fournit un aperçu de l'état du système d'automatisation.

Vous pouvez diagnostiquer le matériel dans la vue « En ligne et diagnostic », dans la Task Card « Outils en ligne » ou dans la zone « Diagnostic > Information appareil » de la fenêtre d'inspection.

#### 2.7.3.1 Structure de la vue En ligne et diagnostic :

La vue En ligne et diagnostic est composée de deux fenêtres :

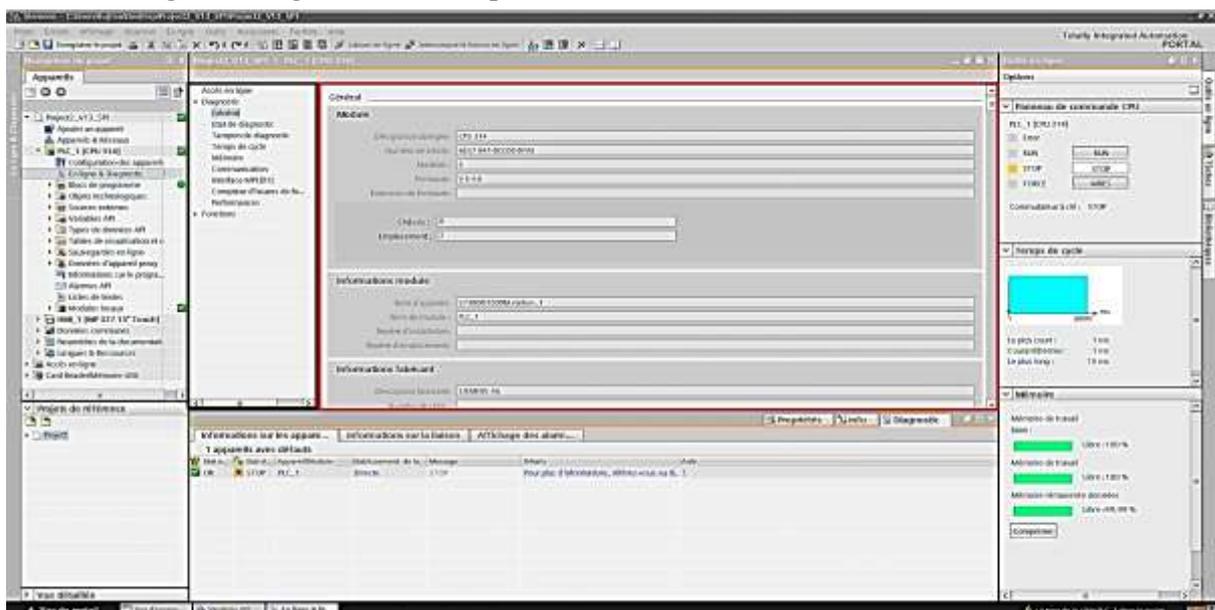


Figure 2-21 : la vue en ligne et diagnostic

La fenêtre de gauche « cadre noir » affiche une arborescence avec des dossiers et des groupes (lorsque les dossiers sont ouverts).

La fenêtre de droite « cadre rouge » contient le détail du dossier ou du groupe sélectionné. On y trouve le groupe « Accès en ligne » et les dossiers « Diagnostic » et « Fonctions » :

- Groupe « Accès en ligne » : Il indique s'il existe ou non, à un moment donné, une liaison en ligne à la cible correspondante. Vous pouvez alors établir ou suspendre la liaison en ligne.
- Dossier « Diagnostic » : contient plusieurs groupes de diagnostic pour le module sélectionné.
- Dossier « Fonctions » : contient plusieurs groupes, dans lesquels vous effectuez des paramétrages ou des commandes du module sélectionné.

### 2.7.3.2 Détecter les appareils défectueux parmi ceux qui sont en ligne :

Dans la zone « Diagnostic > Informations sur les appareils » de la fenêtre d'inspection, vous avez une vue d'ensemble des appareils défectueux pour lesquels il y a ou il y a eu une liaison en ligne.

La zone « Diagnostic > Informations sur les appareils » de la fenêtre d'inspection est composée des éléments (En-tête comportant le nombre d'appareils défectueux et Tableau avec les informations détaillées sur chaque appareil défectueux).

Le tableau comprend les colonnes suivantes :

- **Etat en ligne** : contient l'état en ligne sous forme d'icône de diagnostic et en mots (voir Annexe)
- **Etat de fonctionnement** : contient l'état de fonctionnement sous forme d'icône et en mots
- **Appareil/Module** : nom de l'appareil ou du module concerné
- **Message** : explique l'entrée de la colonne précédente
- **Détails** : le lien ouvre la vue En ligne et Diagnostic de l'appareil ou l'amène au premier plan. S'il n'y a plus de liaison en ligne, le lien ouvre la boîte de dialogue pour l'établissement de la liaison.
- **Aide** : le lien fournit des informations complémentaires sur la défaillance.

Etat en ligne	Etat de fonctionnement	Appareil/Module	Etat de fonctionnement de la liaison	Message	Détails	Aide
	STOP	PLC 1	Directe	STOP		Pour plus d'informations, référez-vous au di. ?

Figure 2-22 : le tableau des informations de diagnostic des appareils

## 2.8 Programmation de l'API :

### 2.8.1 Création du programme utilisateur :

Le système d'exploitation est un composant de la CPU et est déjà installé dans la CPU à la livraison. Chaque CPU contient un système d'exploitation qui organise toutes les fonctions et processus de la CPU n'étant pas liés à une tâche d'automatisation spécifique. Font partie des tâches du système d'exploitation : [5]

- Déroulement du démarrage (à chaud)
- Actualisation de la mémoire image des entrées et de la mémoire image des sorties
- Appel du programme utilisateur
- Acquisition des alarmes et appels des OB d'alarme
- Détection et traitement des erreurs
- Gestion des zones de mémoire

Le programme utilisateur contient toutes les fonctions requises pour le traitement de tâches d'automatisation spécifiques. C'est vous qui créez le programme utilisateur et le chargez dans la CPU. Font partie des fonctions du programme utilisateur : [6]

- Vérification des conditions préalables au démarrage (à chaud) à l'aide d'OB de démarrage, fin de course en bonne position ou dispositif de sécurité actif par exemple.
- Traitement des données du processus, par exemple combinaison de signaux binaires, lecture et évaluation de valeurs analogiques, détermination des signaux binaires de sortie, sortie des valeurs analogiques
- Réaction aux alarmes, par exemple alarme de diagnostic en cas de dépassement de la limite inférieure d'un module d'extension analogique
- Traitement des perturbations dans l'exécution normale du programme

#### 2.8.1.1 Programmation linéaire et structurée :

Vous pouvez résoudre de petites tâches d'automatisation en écrivant le programme utilisateur complet linéairement dans un OB cyclique. Cette démarche est recommandée uniquement pour des programmes simples.

La réalisation et la maintenance de tâches d'automatisation complexes sont plus simples si ces tâches sont divisées en plusieurs tâches partielles plus petites qui correspondent aux fonctions technologiques du processus d'automatisation ou qui peuvent être utilisées plusieurs fois. Dans le programme utilisateur, ces tâches partielles sont représentées par des blocs. Chaque bloc constitue une section indépendante du programme utilisateur.

La structuration du programme offre les avantages suivants : [4]

- La programmation de programmes volumineux est plus claire.
- Certaines parties du programme peuvent être normalisées et être utilisées plusieurs fois avec des paramètres changeants.
- L'organisation du programme est simplifiée.
- Il est plus facile de modifier le programme.

- Le test du programme est simplifié, car il peut s'effectuer section par section.
- La mise en service est simplifiée.

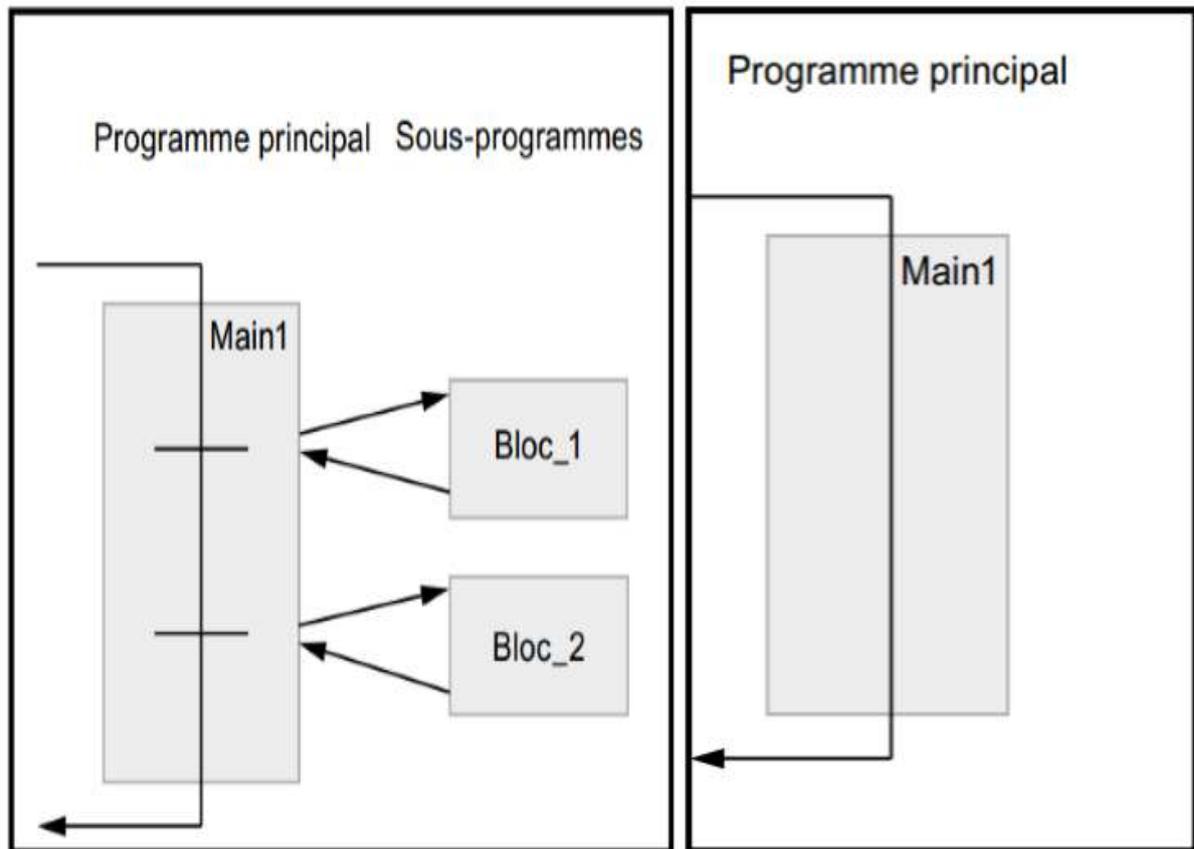


Figure 2-23 : graphes qui représentent la programmation linéaire et structuré

### 2.8.1.2 Présentation des types de blocs :

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

#### 2.8.1.2.1 Blocs d'organisation (OB) :

Les blocs d'organisation (OB) constituent l'interface entre le système d'exploitation et le programme utilisateur. Ils sont appelés par le système d'exploitation et commandent par exemple les opérations suivantes :

- Comportement de démarrage du système d'automatisation
- Traitement cyclique du programme
- Traitement du programme déclenché par alarme
- Traitement des erreurs

Vous pouvez programmer les blocs d'organisation et déterminer ainsi le comportement de la CPU. En fonction de la CPU utilisée, vous avez à votre disposition différents blocs d'organisation.

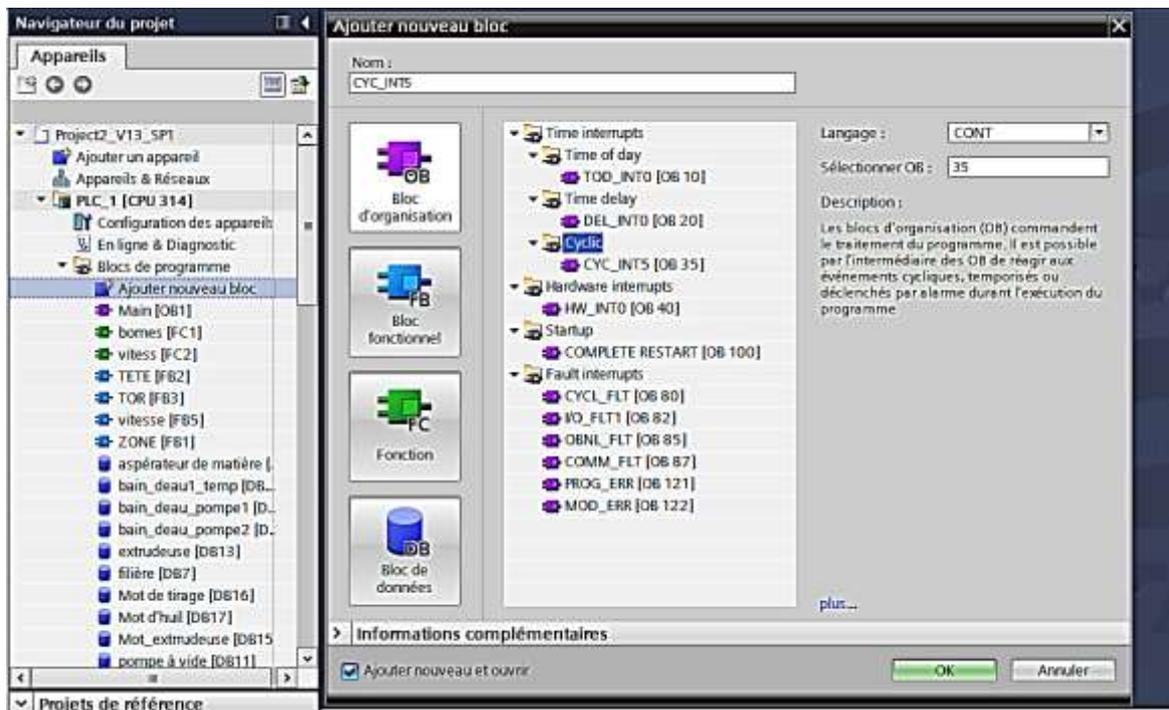


Figure 2-24 : les différents blocs d'organisation

#### 2.8.1.2.2 Fonctions (FC) :

Les fonctions (FC) sont des blocs de code sans mémoire. Elles n'ont pas de mémoire de données dans laquelle il est possible d'enregistrer les valeurs de paramètres de bloc. C'est pourquoi des paramètres effectifs doivent être fournis à tous les paramètres formels lors de l'appel d'une fonction.

#### 2.8.1.2.3 Blocs fonctionnels (FB) :

Les blocs fonctionnels sont des blocs de code qui mémorisent durablement leurs paramètres d'entrée, de sortie et d'entrée/sortie dans des blocs de données d'instance afin qu'il soit possible d'y accéder même après le traitement de blocs. C'est pourquoi ils sont également appelés « Blocs avec mémoire ».

#### 2.8.1.2.4 Blocs de données (DB) d'instances et globaux :

Un appel d'un bloc fonctionnel est désigné par le terme « instance ». Pour chaque instance d'un bloc fonctionnel, il faut un bloc de données d'instance dans lequel sont mémorisées des valeurs spécifiques à l'instance pour les paramètres formels déclarés dans le FB.

Les blocs de données globaux enregistrent des données qui peuvent être utilisées par tous les autres blocs. La taille maximale des blocs de données varie selon la CPU. Vous pouvez définir la structure des blocs de données globaux à votre gré.

### 2.8.2 Créer et gérer des tables des variables API :

Les tables des variables API contiennent les définitions des variables API et constantes symboliques valables sur l'ensemble de la CPU.

Vous pouvez créer plusieurs tables des variables API utilisateur dans une CPU. Chaque table des variables doit avoir un nom univoque sur l'ensemble de la CPU. Procédez comme suit pour créer une nouvelle table des variables API :

- 1) Dans la navigation du projet, ouvrez le dossier « Variables API » qui se trouve sous la CPU.
- 2) Effectuez un double clic sur l'entrée « Insérer une nouvelle table de variables ».
- 3) Une nouvelle table des variables API portant le nom par défaut « Table des variables-X » est créé.
- 4) Dans la navigation du projet, sélectionnez la table des variables API.
- 5) Choisissez la commande « Renommer » dans le menu contextuel.
- 6) Saisissez un nom univoque sur l'ensemble de la CPU.

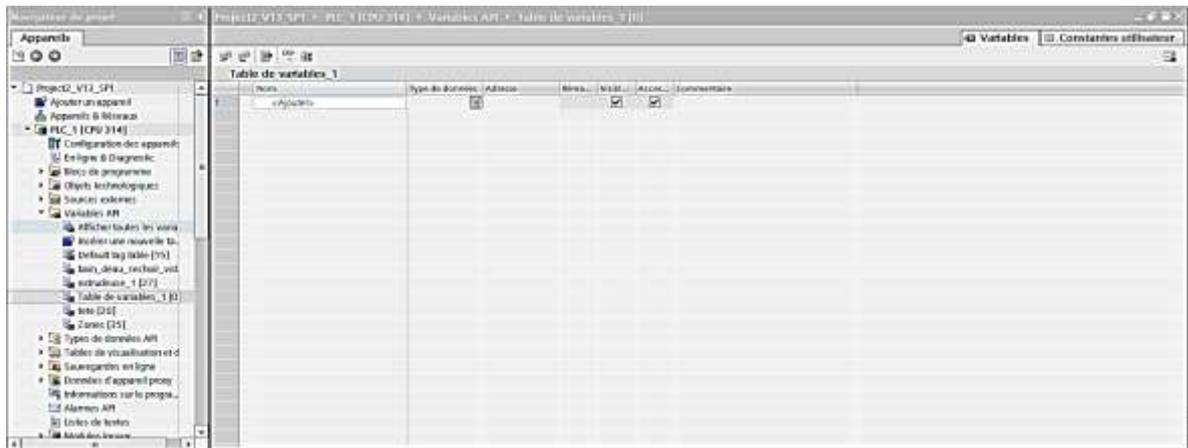


Figure 2-25 : création d'une table des variables

### 2.8.3 Programmation des blocs :

L'éditeur de programmation est l'environnement de développement intégré pour la programmation des fonctions, blocs fonctionnels et blocs d'organisation. Il offre une assistance optimale lors de la création des programmes et de la recherche des erreurs. Selon la CPU utilisée, le langage de programmation et le type de bloc utilisés, l'aspect et les fonctionnalités de l'éditeur de programmation peuvent varier.

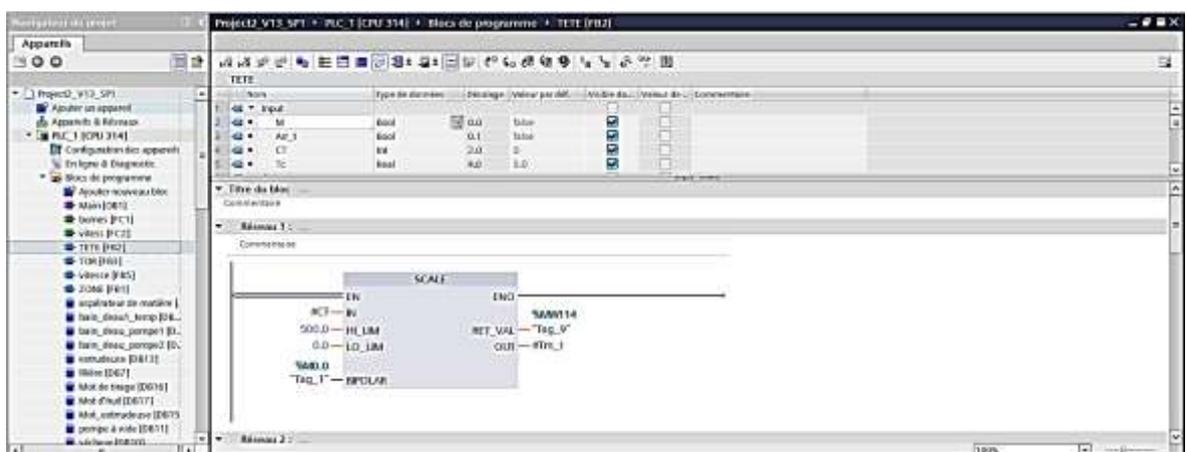


Figure 2-26 : interface de programmation des blocs

### 2.8.4 Affichage des informations programme :

Les informations programme d'un programme utilisateur contiennent les vues suivant :

2.8.4.1 « Tableau d'affectation » :

Indique quels bits des opérands des zones de mémoire I, Q et M sont déjà affectés dans le programme utilisateur.

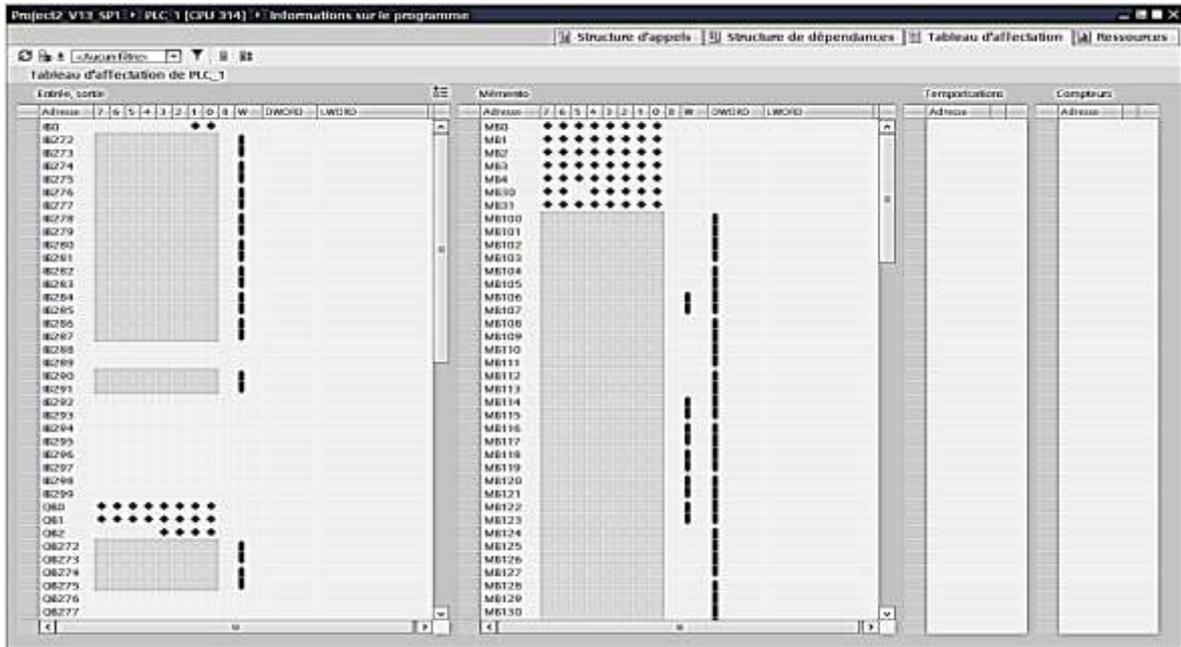


Figure 2-27 : la vue du tableau d'affectation

2.8.4.2 « Structure d'appel » :

Montre la hiérarchie d'appel des blocs à l'intérieur d'un programme utilisateur et donne une vue d'ensemble des blocs utilisés et de leurs dépendances.

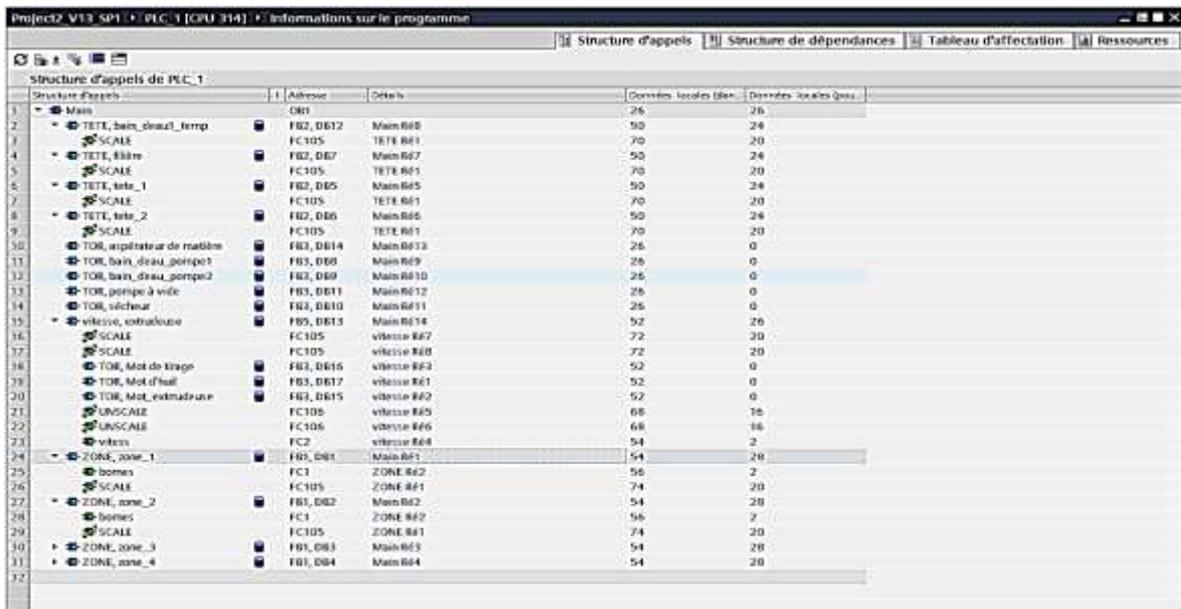


Figure 2-28 : la vue qui représente la hiérarchie d'appel des blocs

2.8.4.3 « Structure de dépendance » :

Montre la liste des blocs utilisés dans le programme utilisateur. Le premier niveau contient le bloc concerné sous lequel apparaissent, en retrait, les blocs appelant ou utilisant ce bloc. Contrairement à la présentation dans la structure d'appel, les blocs de données d'instance sont énumérés à part.

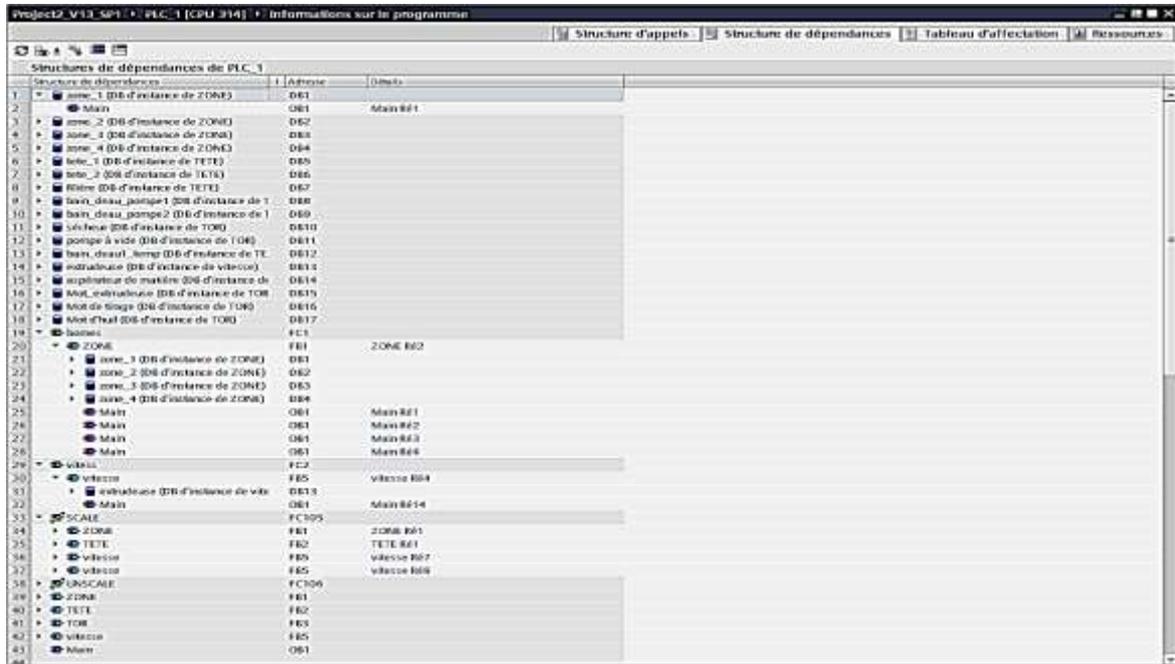


Figure 2-29 : la vue qui représente la structure de dépendance

2.8.4.4 « Ressources » :

Affiche l'utilisation des ressources matérielles de la CPU pour les objets (OB, FC, FB, DB, types de données utilisateur et variables API), les zones de mémoire de la CPU et les modules d'entrées/sorties présents.

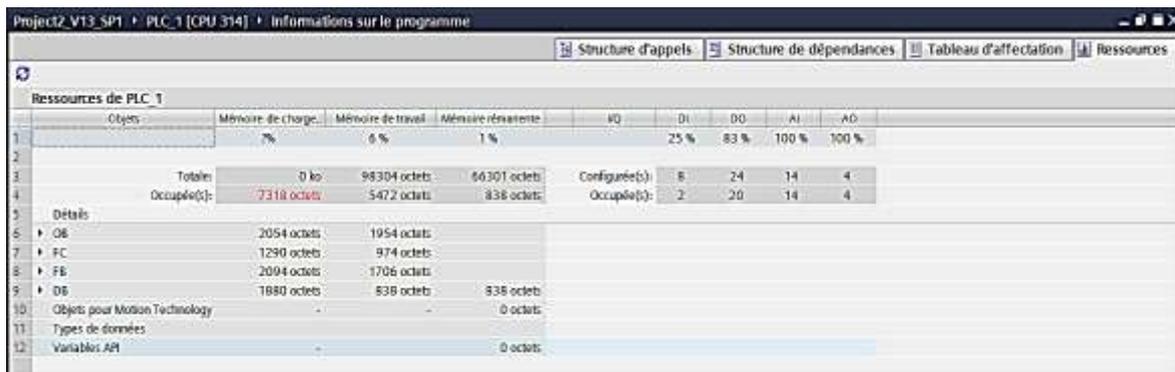


Figure 2-30 : la vue qui affiche l'utilisation des ressources matérielles de la CPU

2.9 Visualisation de processus :

Lorsque la complexité des processus augmente et que les machines et les installations doivent répondre à des spécifications de fonctionnalité toujours plus sévères, l'opérateur a

besoin d'un maximum de transparence. Cette transparence s'obtient au moyen de l'interface homme machine (IHM).

### 2.9.1 Créer et éditer objet HMI :

D'abord, il faut choisir l'appareil de supervision « appareil HMI », procédez comme suit pour ajouter un appareil HMI :

- Dans le navigateur de projet double clic sur « ajouter un appareil », la fenêtre « Ajouter un appareil » s'ouvre
- Sélectionner HMI ou systèmes PC puis parcourir jusqu'à l'appareil souhaiter, vous le sélectionnez
- Cliquez « OK », l'appareil souhaiter est ajouté.

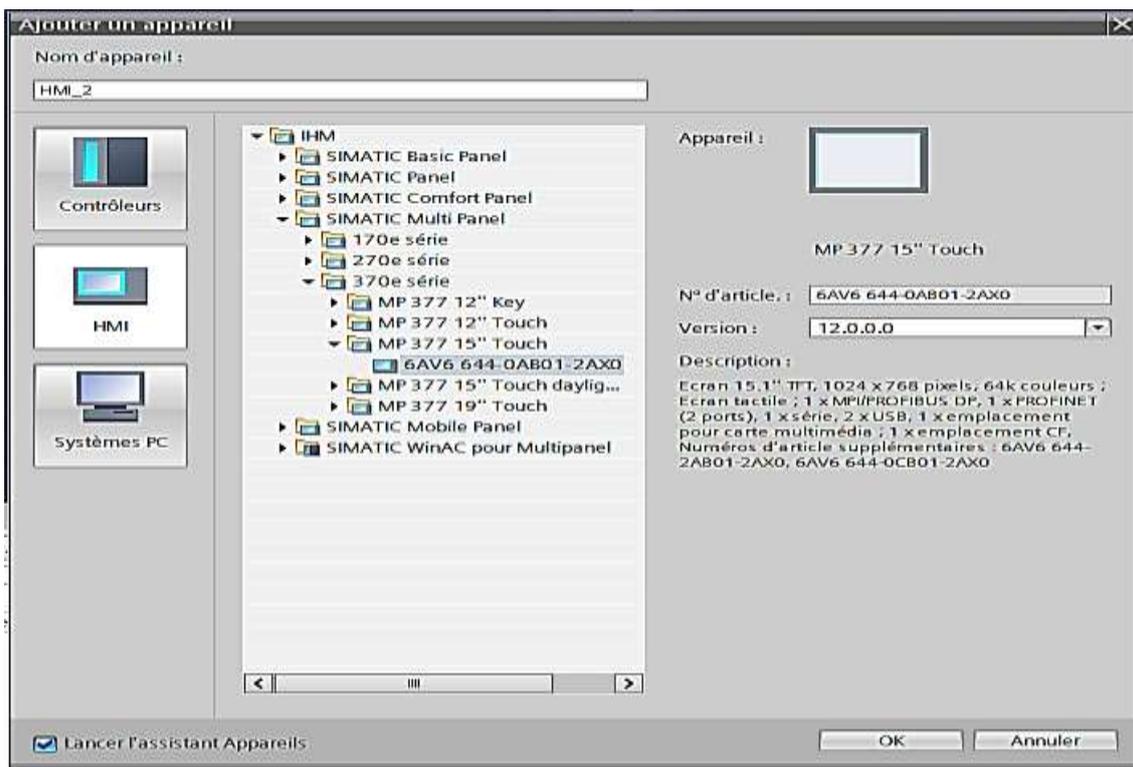


Figure 2-31 : Ajouter un appareil HMI

### 2.9.2 Créer les vues :

Dans Win CC, vous créez des vues pour le contrôle/commande de machines et d'installations. Pour créer des vues, vous disposez d'objets prédéfinis permettant de représenter votre installation, d'afficher des procédures et de définir des valeurs de processus.

Pour ajouter une nouvelle vue procédez comme suit :

- Dans la navigation du projet, effectuez un double clic sur « Vues > Ajouter nouvelle vue ». La vue est créée dans le projet et s'affiche dans la zone de travail. Les propriétés de la vue s'affichent dans la fenêtre d'inspection.
- Entrez un nom explicite pour la vue.
- Configurez les propriétés de la vue dans la fenêtre d'inspection.

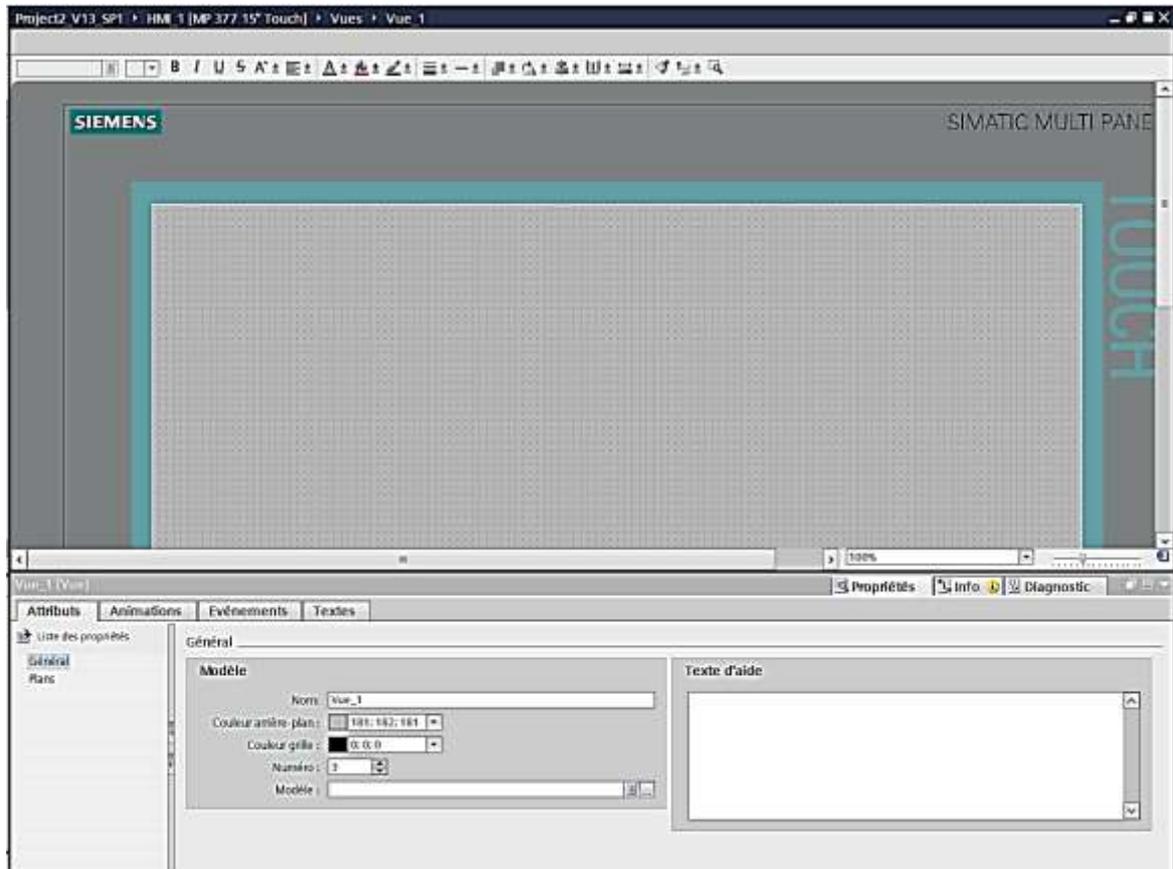


Figure 2-32 : création d'une nouvelle vue

La vue est créée dans votre projet. Dans les étapes suivantes, vous pouvez insérer des objets et des éléments de commande tirés de la Task Card "Outils" et programmer des touches de fonction.

La vue initiale est la première vue qui s'affiche en Runtime au démarrage du projet. Vous pouvez définir une vue initiale particulière à chaque pupitre opérateur. L'utilisateur appelle les autres vues à partir de la vue initiale.

Dans la navigation du projet, effectuez un double clic sur « Paramètres Runtime > Général ». Sélectionnez la vue souhaitée comme « Vue initiale ».

Configurez dans un modèle des objets qui seront représentés dans toutes les vues basées sur ce modèle.

### 2.9.3 Utilisation des objets :

Les objets sont des éléments graphiques avec lesquels vous élaborerez les vues de votre projet.

La Task Card « Outils » contient tous les objets disponibles pour le pupitre opérateur. Selon l'éditeur actuellement ouvert, la fenêtre d'outils contient différentes palettes :

- « Objets simples » : Les objets simples sont des objets graphiques fondamentaux, tels que la « ligne », le « cercle », le « Champ de texte » ou la « Vue de graphique », ...etc.

Icône	Objet	Remarques
	"Ligne"	-
	"Ellipse"	-
	"Cercle"	-
	"Rectangle"	-
	"Champ de texte"	Représente un texte à une ou plusieurs lignes. La police et la représentation sont paramétrables.
	"Vue de graphique"	Affiche des graphiques tirés de programmes graphiques externes et insère des objets OLE. Vous pouvez utiliser les formats graphiques suivants : "*.emf", "*.wmf", "*.dib", "*.bmp", "*.jpg", "*.jpeg", "*.gif" et "*.tif".

Figure 2-33 : tableau des objets simples

- « Eléments » : Les éléments sont des éléments de commande fondamentaux, tels que « Champ d'E/S », « Bouton » ou « Instrument à aiguille ».

Icône	Objet	Remarques
	"Champ d'E/S"	Fournit les valeurs d'une variable et/ou écrit des valeurs dans une variable. Vous définissez des limites pour les valeurs de variable représentées dans le champ d'E/S. Pour une saisie opérateur invisible en runtime, configurez "Saisie masquée".
	"Bouton"	Selon la configuration, exécute une liste de fonctions ou un script.
	"Champ d'E/S symbolique"	Fournit les valeurs d'une variable et/ou écrit des valeurs dans une variable. Selon la valeur de la variable, un texte d'une liste de textes s'affiche.
	"Champ d'E/S graphique"	Fournit les valeurs d'une variable et/ou écrit des valeurs dans une variable. Selon la valeur de la variable, un graphique d'une liste de graphiques s'affiche.
	"Champ date/heure"	Affiche la date et l'heure du système ou d'une variable. Permet à l'opérateur de saisir de nouvelles valeurs. Le format d'affichage est paramétrable.
	"Bargraphe"	Représente une valeur de l'automate sous forme de colonne munie d'une échelle.
	"Commutateur"	Commute entre deux états définis. La légende du commutateur peut être un texte ou un graphique.

Figure 2-34 : Tableau des objets Eléments

- « Contrôles » : Les contrôles ont des fonctions étendues. Ils représentent aussi des déroulements du procès de manière dynamique, comme le font la vue de courbe et la vue de recette.

Icône	Objet	Description
	"Vue des alarmes"	Affiche les alarmes ou événements d'alarme actuels en attente provenant du tampon d'alarmes ou de l'archive d'alarmes.
	"Vue de courbe"	Représente plusieurs courbes montrant l'évolution de valeurs provenant de l'automate ou d'une archive.
	"Vue des utilisateurs"	Permet à un administrateur de gérer les utilisateurs sur le pupitre opérateur. Permet à un opérateur sans droits d'administrateur de modifier son mot de passe.
	"HTML Browser" <sup>(1)</sup>	Affiche des pages HTML.
	"Vue de recette"	Affiche des enregistrements et permet leur traitement.
	"Vue de diagnostic système"	Permet d'avoir une vue d'ensemble de tous les appareils capables de diagnostic. Affiche les erreurs survenues sur l'installation.

Figure 2-35 : Tableau des objets de contrôles

- « Graphiques » : Les graphiques sont regroupés par thèmes dans une arborescence de dossiers. Les différents dossiers contiennent p. ex. les représentations graphiques suivantes : Parties de machines et d'installations, Appareils de mesure, éléments de commande ... etc.
- Task Card « Bibliothèques » : En plus des objets d'affichage et de commande, vous disposez également des objets de bibliothèque. Ils se trouvent dans les palettes de la Task Card « Bibliothèques ». Une bibliothèque contient des objets déjà configurés, tels que des graphiques de tuyaux, pompes, boutons préconfigurés.

Pour insérer un objet dans une vue, vous glissez l'objet souhaiter de la Task Card « Outils » à cette vue, puis le configurer ça dépend des propriétés de l'objet dans la fenêtre d'inspection.

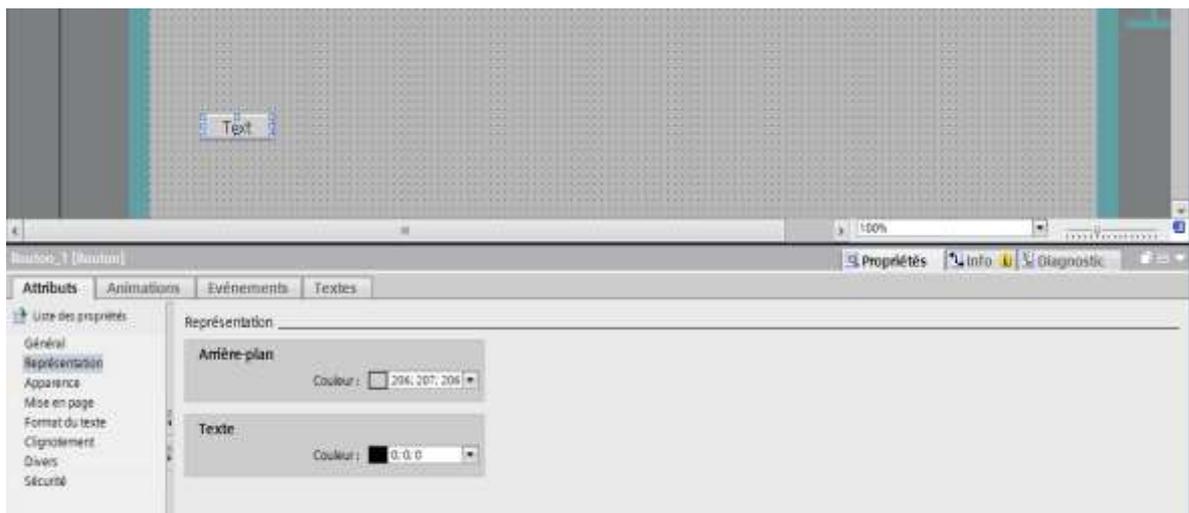


Figure 2-36 : insérer un objet dans une vue

### 2.9.4 Dynamiser des vues :

Dans Win CC, vous dynamisez des objets afin de visualiser votre installation sur des pupitres opérateur et d'afficher les étapes de traitement d'un procès. Réalisez des dynamisations avec des (Animations, Variables, Fonctions système)

Citons pour exemple l'image d'un réservoir dont le niveau de liquide augmente ou diminue suivant une valeur de procès.

### 2.9.5 Utilisation de plans :

Pour traiter les objets d'une vue de manière différenciée, vous utilisez des plans. Une vue comporte 32 plans que vous pouvez nommer comme vous le désirez. Lorsque vous affectez des objets aux plans, vous définissez ainsi la profondeur d'imbrication de la vue. Les objets du plan 0 sont à l'arrière-plan de la vue, les objets du plan 31 au premier plan.

### 2.9.6 Utilisation des variables :

Dans Runtime, les valeurs de procès sont transmises par des variables. Les valeurs de procès sont des données enregistrées dans la mémoire de l'un des automates connectés. Dans le navigateur du projet, chaque appareil IHM dispose d'un dossier "Variables IHM".

Win CC utilise deux types de variables :

- Les variables externes constituent le lien entre Win CC et les systèmes d'automatisation. Les valeurs des variables externes correspondent aux valeurs de procès de la mémoire d'un système d'automatisation.
- Les variables internes ne sont pas liées au procès et ne transmettent que des valeurs à l'intérieur de Win CC. Les valeurs de variable sont disponibles uniquement tant que Runtime est en cours.

### 2.9.7 Utiliser les archives :

Win CC met à votre disposition les types d'archives suivants pour archiver des données de procès pour HMI Runtime :

- Une archive de variables sert à archiver les données de procès d'une installation industrielle.

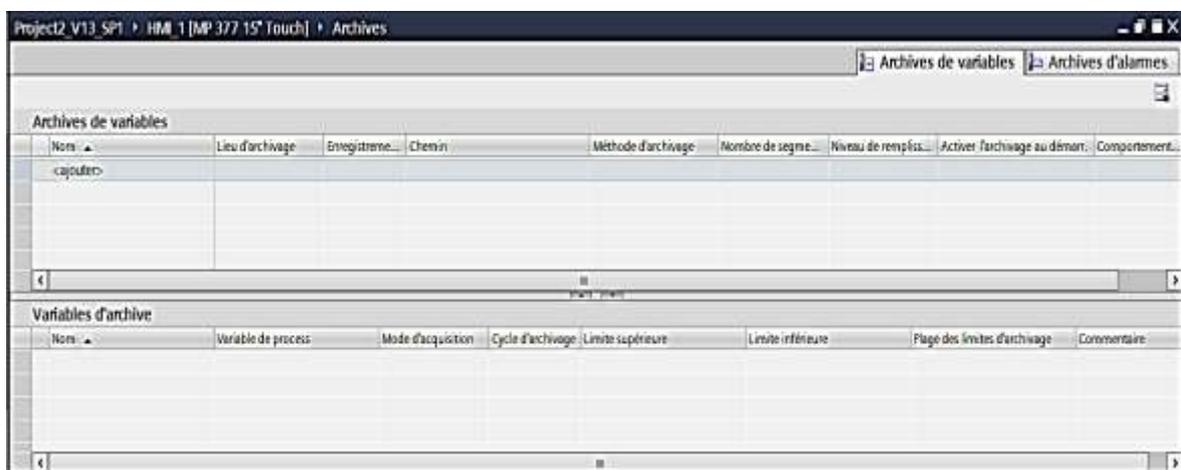


Figure 2-37 : la fenêtre « Archives de variables »

- Une archive d'alarmes sert à archiver les alarmes qui surviennent dans le procès visualisé.

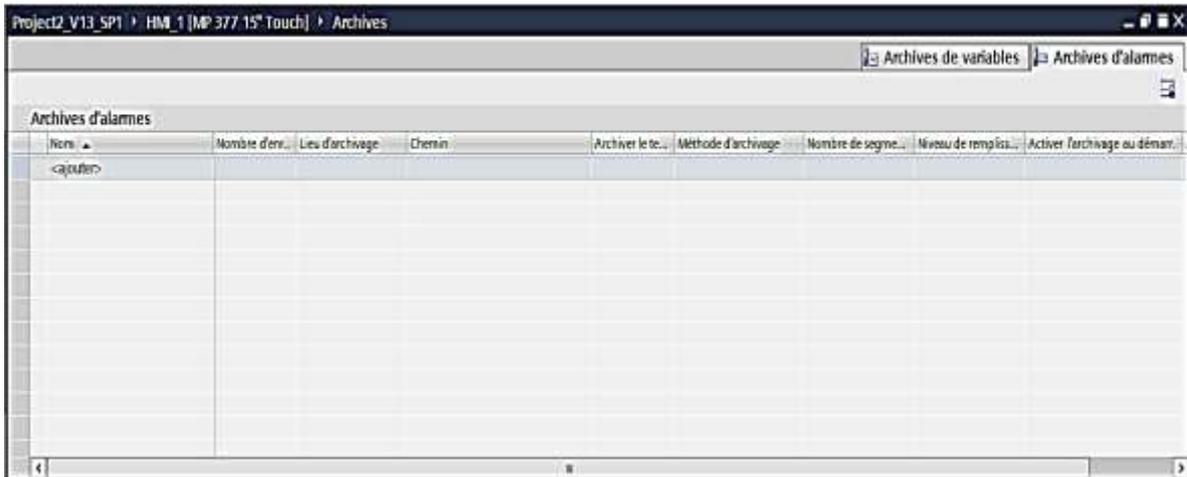


Figure 2-38 : la fenêtre « Archives des alarmes »

**2.9.8 Travailler avec des alarmes :**

Le système d'alarmes permet d'afficher sur le pupitre opérateur les états de fonctionnement et les erreurs qui se préparent ou qui surviennent dans une installation.

Le système d'alarmes traite différents types d'alarme. Il distingue les alarmes définies par le système et les alarmes définies par l'utilisateur :

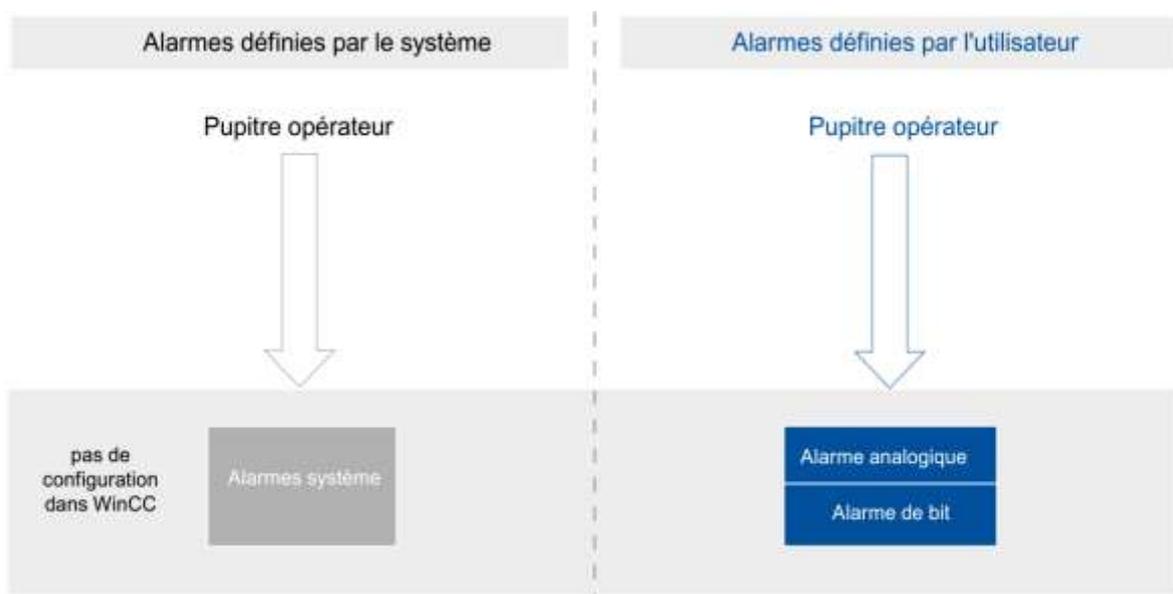


Figure 2-39 : Un graphique qui représente les types des alarmes

- Les alarmes définies par l'utilisateur servent à surveiller le procès de l'installation.
- Les alarmes définies par le système servent à surveiller le pupitre opérateur « Alarmes système » (voir Annexe).

Vous pouvez documenter les alarmes du projet dans une archive d'alarmes.

### 2.9.9 Utilisation de recettes :

Les recettes regroupent des données de même type, par ex. des paramétrages de machine ou des données de production. Exemples :

- Des paramétrages de machine qui sont nécessaires pour adapter la production à une autre variante de produit.
- Des composants qui donnent d'autres produits finaux suivant les proportions de leur combinaison.

Les recettes sont stockées sur le pupitre opérateur. Un enregistrement de recette est en principe transféré complètement en une seule étape entre le pupitre opérateur et l'automate.

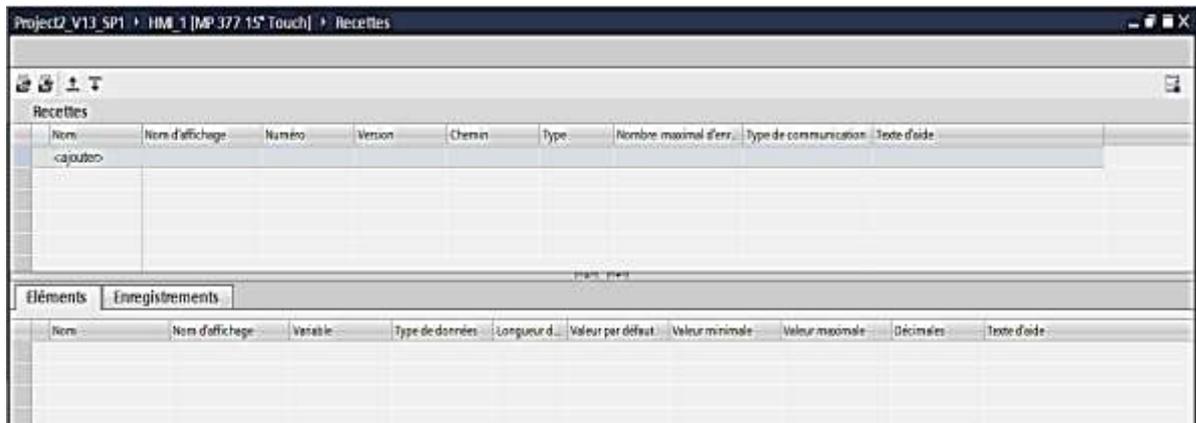


Figure 2-40 : la fenêtre de création des recettes

La vue simple de recette est un objet d'affichage et de commande prédéfini servant à gérer les enregistrements de recette. Elle présente les enregistrements sous forme de tableau.

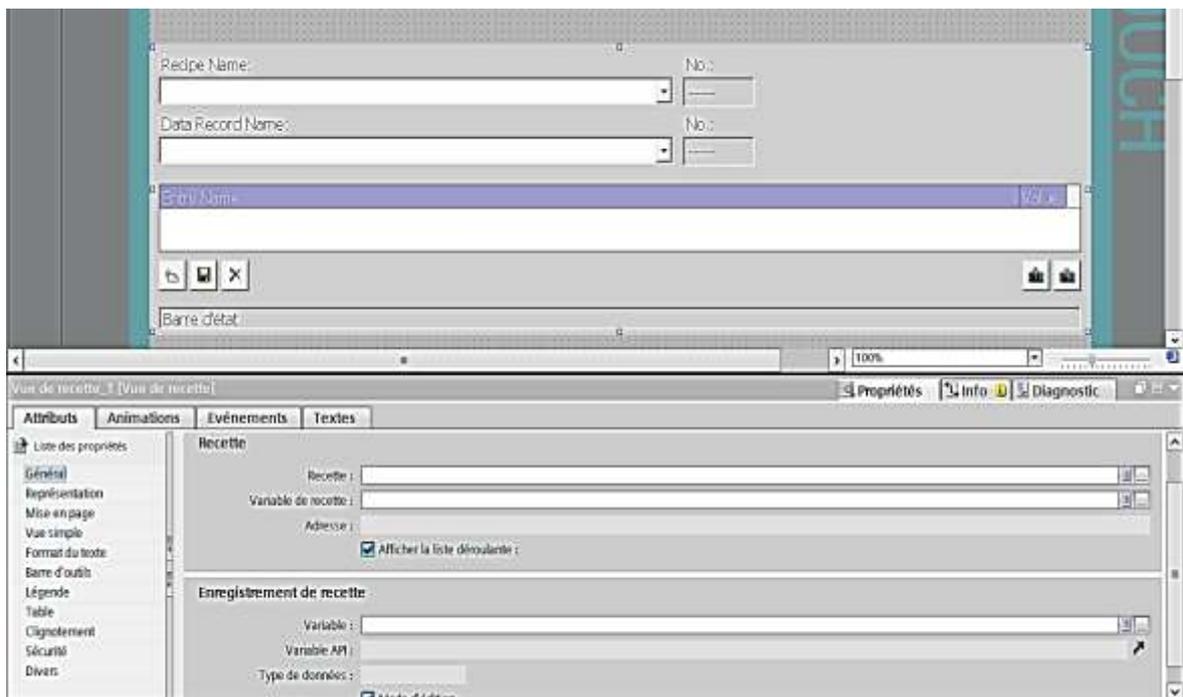


Figure 2-41 : utilisation l'objet vue de recette dans la vue de supervision

### 2.9.9.1 Création d'une recette :

Pour créer une recette procédez comme suit :

- Dans le navigateur de projet sous l'objet HMI cliquer 2 fois sur « Recettes ». L'éditeur « Recettes » s'ouvre.
- Dans l'éditeur « Recettes », cliquez sur « Ajouter » dans la première ligne vide de la table. La nouvelle recette est créée et s'affiche dans une ligne.
- Dans la zone « Général », saisissez sous « Nom » un nom explicite pour la recette. Ce nom identifie la recette de manière univoque dans un pupitre opérateur.
- Sous « Nom d'affichage », saisissez le nom qui s'affichera dans la vue de recette selon la langue.
- Sélectionnez un numéro de recette sous « Numéro ». Ce numéro identifie la recette de manière univoque dans le pupitre opérateur. La recette est munie automatiquement d'une version qui indique la date et l'heure de la dernière modification. Une autre solution consiste à saisir ici des informations à votre gré sur la recette.
- Définissez sous « Stockage des données » le lieu de stockage pour les enregistrements de recette. Les possibilités offertes dépendent du pupitre opérateur utilisé.
- Saisissez une info-bulle qui s'affichera pour l'opérateur dans Runtime.
- Pour accorder à la vue de recette, dans Runtime, les variables de recette qui sont configurées dans des champs d'E/S, activez « Synchroniser vue de recette et variables de recette » sous « Attributs > Synchronisation » dans la fenêtre d'inspection.
- Pour que les variables de recette soient transférées automatiquement dans l'automate lorsqu'elles sont éditées dans les champs d'E/S, désactivez « Transfert manuel de valeurs individuelles modifiées (mode Apprentissage) ».
- Pour que le transfert des données de recette soit surveillé au moyen d'un pointeur de zone dans Runtime, activez « Transfert coordonné des enregistrements ».
- Sélectionnez sous "Synchroniser avec" la connexion appropriée à l'automate pour ce transfert coordonné.

### 2.9.10 Compiler et charger des projets :

Déjà au moment où vous configurez un projet dans Win CC, le projet est compilé en permanence en arrière-plan. Cela permet de réduire la durée de la compilation finale. Lorsque vous lancez la compilation, un fichier exécutable sur le pupitre opérateur concerné est créé. Chargez le projet compilé sur les pupitres opérateur sur lesquels vous souhaitez utiliser le projet.

Si vous utilisez dans votre projet des variables IHM qui sont associées à des variables API, compilez aussi tous les blocs S7 modifiés avant le chargement dans le pupitre opérateur IHM avec la commande « Compiler > Logiciel » dans le menu contextuel.

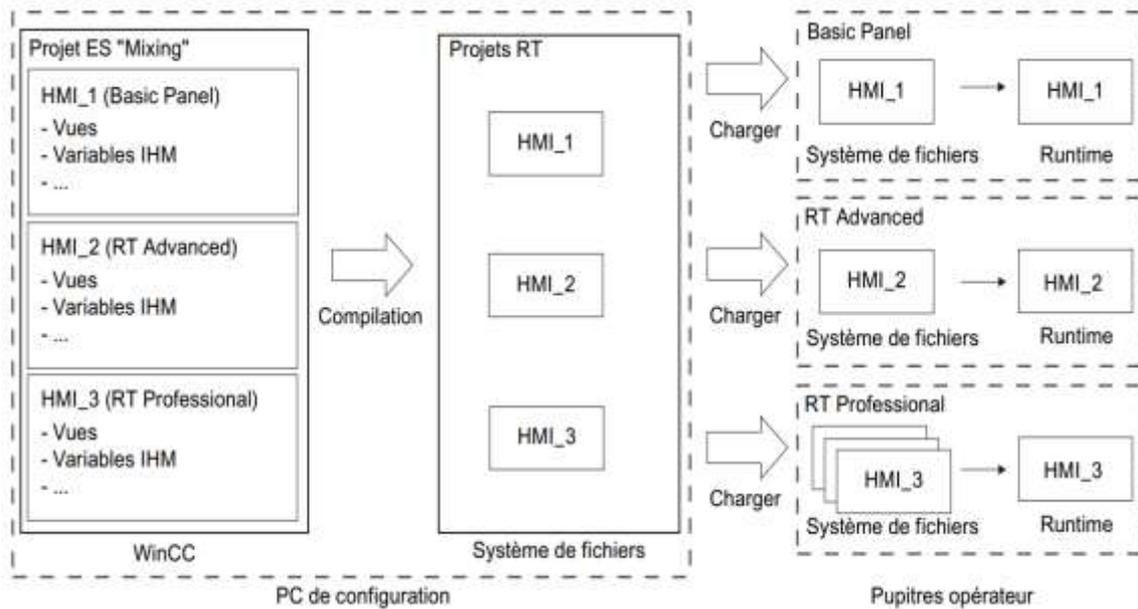


Figure 2-42 : la structure de compilation et de chargement des projets HMI

## 2.10 Conclusion :

Dans ce chapitre, nous avons présenté une description générale sur le TIA portal, où le lecteur de notre travail peut comprendre comment on a utilisé TIA portal.

Le STEP7 TIA offre la possibilité de programmer et de tester, tels que la visualisation du programme afin de remédier à d'éventuelles erreurs commises et les modifications appropriées avant de passer à l'implémentation dans l'automate.

Le WinCC TIA offre le maximum de transparence et essentiel pour l'opérateur qui travaille dans un environnement où les processus sont de plus en plus complexes.

## **Chapitre 3 : Présentation de la ligne de gainage BMN120**

### 3.1 Introduction :

Les Câbleries de Télécommunications d'Algérie « CATEL » est une société par actions spécialisée dans la production et la commercialisation des fils et câbles de télécommunications en cuivre et en fibre optique et dans la production et la commercialisation des câbles de signalisation, d'énergie de basse tension, des câbles de commande, des câbles coaxiaux, des câbles d'instrumentation, des câbles anti incendie et des câbles spécifiques.

L'usine a été créée en 1928 sur décision de la société française « Lignes Télégraphiques et Téléphoniques » de France. Sa nationalisation par l'état algérien est intervenue en 1968. Elle a été rattachée aussitôt à la SNMETAL et puis à la SONELEC en 1969.

L'entreprise nationale des industries du câble « ENICAB » a rattaché en 1983 l'Ex LTT à son patrimoine et a donné naissance, après sa restructuration en 1997, à la société CATEL qui avait pour principale activité : Le tréfilage, la production et la commercialisation des câbles de télécommunications en cuivre. Elle a créé en 2005, en partenariat avec la Holding Cableco Sal, la société CATEL FIBROPTIC, spécialisée dans la production et la commercialisation des câbles à fibre optique. [7]

### 3.2 Câbles de télécommunications en cuivre:

Le câble de télécommunication en cuivre est l'un des plusieurs supports importants de télécommunication, il existe sous différentes formes et structures tout dépend de son utilisation.

CATEL fabrique 4 catégories de câbles de télécommunications en cuivre (les câbles urbains souterrains, les câbles urbains autoportés, les câbles unipolaires de raccordement aérien, câbles de raccordement et d'installations privées), où chaque catégorie contient plusieurs séries. Ces exigences de production ont obligé l'entreprise à travailler avec des différentes chaînes de productions, par exemple la chaîne de gainage BMN120 qui est réservé uniquement pour les deux premières catégories. [7]

#### 3.2.1 Les câbles urbains souterrains :

Les câbles souterrains sont utilisés pour les réseaux téléphoniques locaux, en transport entre le central et les sous répartiteurs et en distribution finale jusqu'au point de raccordement de l'utilisateur final. Il y a deux séries, la série 88 et la série 89 qui a des caractéristiques de transmission amélioré. (Figure 3-1)

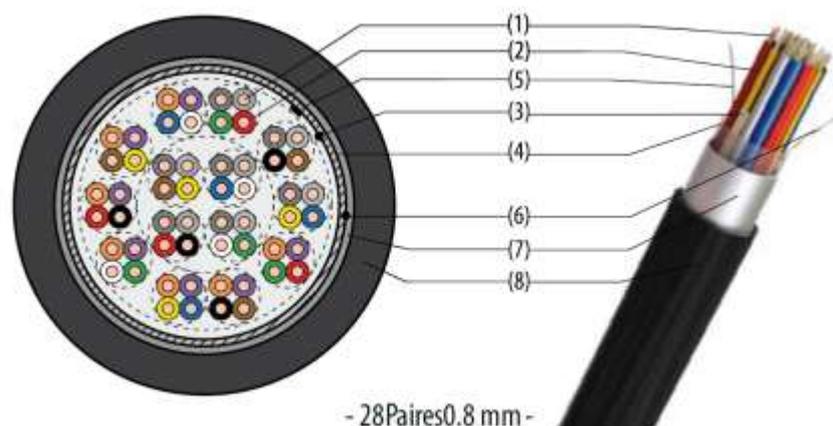


Figure 3-1 : la structure de construction du câble urbain souterrain (série 89)

1. Conducteur : âme massive en cuivre recuit de diamètre 8/10 mm
  2. Isolant : Polyéthylène massif
  3. Bande métrée en papier
  4. Deux (02) rubans polyester disposés en hélice
  5. Fil de continuité en cuivre étamé de 0.53 mm de diamètres
  6. Filin de déchirement placé entre l'âme du câble et le ruban d'aluminium
  7. Ruban en aluminium contre-couché polyéthylène formant une barrière d'étanchéité
  8. Gaine extérieure en polyéthylène basse densité de couleur noire.
  9. Marquage de la gaine : Année Client série et type.
- Les constituants (5, 6, 7, 8,9) sont ajoutés au niveau de la chaine de gainage.

### 3.2.2 Les câbles urbains autoportés :

Les câbles urbains autoportés sont utilisés pour les réseaux aériens de distribution urbaine et rurale et pour relier les centraux entre eux ou des centraux à des sous répartiteurs. Il y a deux série (série 98 et la série 99). (**Figure 3-2**)

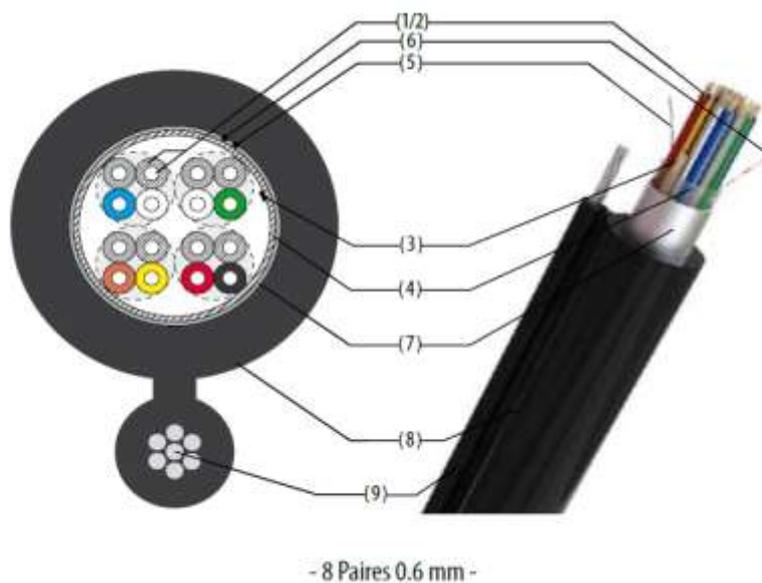


Figure 3-2 : la structure de construction d'un câble autoporté (série 98)

1. Conducteur : âme massive en cuivre recuit de diamètres 4/10 et 6/10 mm
  2. Isolant : Polyéthylène massif
  3. Bande métrée en papier
  4. Deux (02) rubans polyester disposés en hélice
  5. Fil de continuité en cuivre étamé de 0.53 mm de diamètre
  6. Filin de déchirement placé entre l'âme du câble et le ruban d'aluminium
  7. Ruban en aluminium contre-couché polyéthylène formant une barrière d'étanchéité
  8. Gaine extérieure en polyéthylène basse densité de couleur noire.
  9. Porteur en acier.
  10. Marquage de la gaine : Année Client série et type
- Les constituants (5, 6, 7, 8, 9,10) sont ajoutés au niveau de la chaine de gainage.

### 3.3 La chaine de gainage BMN120 :

Au niveau de la chaine de gainage le câble prend sa couche externe et finale, on a à l'entrée un assemblage qui va sortir par la suite comme un câble fini, prêt à la livraison.

La chaine de gainage est une installation qui contient plusieurs composants (capteur, actionneur, afficheur,...etc.

Avant de citer les éléments de la chaîne, on commence par présenter quelques définitions sur les mots techniques utilisés. (Figure 3-5)

### 3.3.1 Définitions :

**Ame conductrice (ou conducteur) :** c'est la partie dont la fonction est de véhiculer le signal d'un point à un autre. Elle est constituée d'un fil en cuivre et elle prend généralement une forme circulaire.

**Isolant :** Enveloppe diélectrique couvrant l'âme du conducteur dont la fonction est d'interdire le passage de tout courant électrique vers l'extérieur. Elle est soit (en polychlorure de vinyle PVC, en polyéthylène PE de haute ou basse densité, ou autre comme: XLPE, silicone, EPR, ...etc.

**Conducteur isolé :** Ensemble de l'âme et de l'isolant.

**Assemblage :** Ensemble de conducteurs isolés assemblés entre eux, le plus souvent par une disposition en hélice, ou en plusieurs couches. (Figure 3-3)



Figure 3-3 les modes d'assemblage

**Revêtement d'assemblage :** Un ou plusieurs rubans en polyester recouvrant les conducteurs assemblés.

**Fil de continuité :** C'est un fil de cuivre étamé de 0.5mm de diamètre et disposé longitudinalement ou en hélice à pas long. Ce fil est destiné à assurer la continuité électrique de l'écran.

**Ecran :** Constitué par un ruban ALUPE (aluminium + copolymère d'éthylène) adhérent à la face interne de la gaine de protection et dont la fonction est d'isoler le conducteur ou le câble des champs électromagnétiques extérieurs pouvant perturber son fonctionnement

**Filin de déchirement :** Placé longitudinalement entre l'âme du câble et le ruban d'aluminium. Il est utilisé pour la reconnaissance et le déchirement de la gaine.

**Gaine d'étanchéité :** Revêtement tubulaire continu, uniforme, de matériaux non métalliques, le plus souvent extrudé et recouvrant l'écran ou l'assemblage des conducteurs et bourrages éventuels.

**Blindage électromagnétique :** Est un blindage en bande de cuivre ou tresse qui consiste à réduire le champ électromagnétique au voisinage d'un objet en interposant une barrière entre la source du champ et l'objet à protéger.

**Armature :** Revêtement constitué de feuillards d'acier et posé de différentes façons. Il est utilisé comme protection mécanique.

**Gaine de protection** : Revêtement en matière thermoplastique, appliqué pour assurer la protection externe du câble.

**Nomenclature du câble** : elle représente l'identifiant du câble et contient le numéro de série, le nombre de paires et le diamètre nominal du conducteur (**Figure 3-4**)



Figure 3-4 : nomenclature du câble

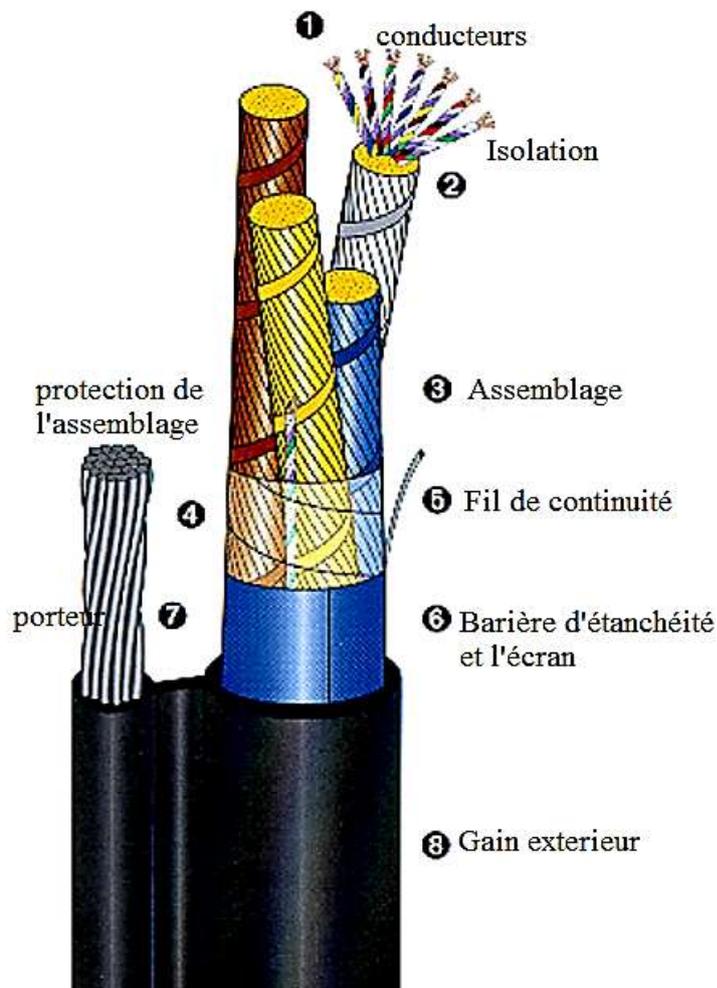


Figure 3-5 : Les constituants d'un câble (séries 98-99)

### 3.4 Les éléments de la chaîne :

En effet la chaîne se décompose en plusieurs parties où chaque partie a une tâche élémentaire bien précise (**Figure 3-6**).

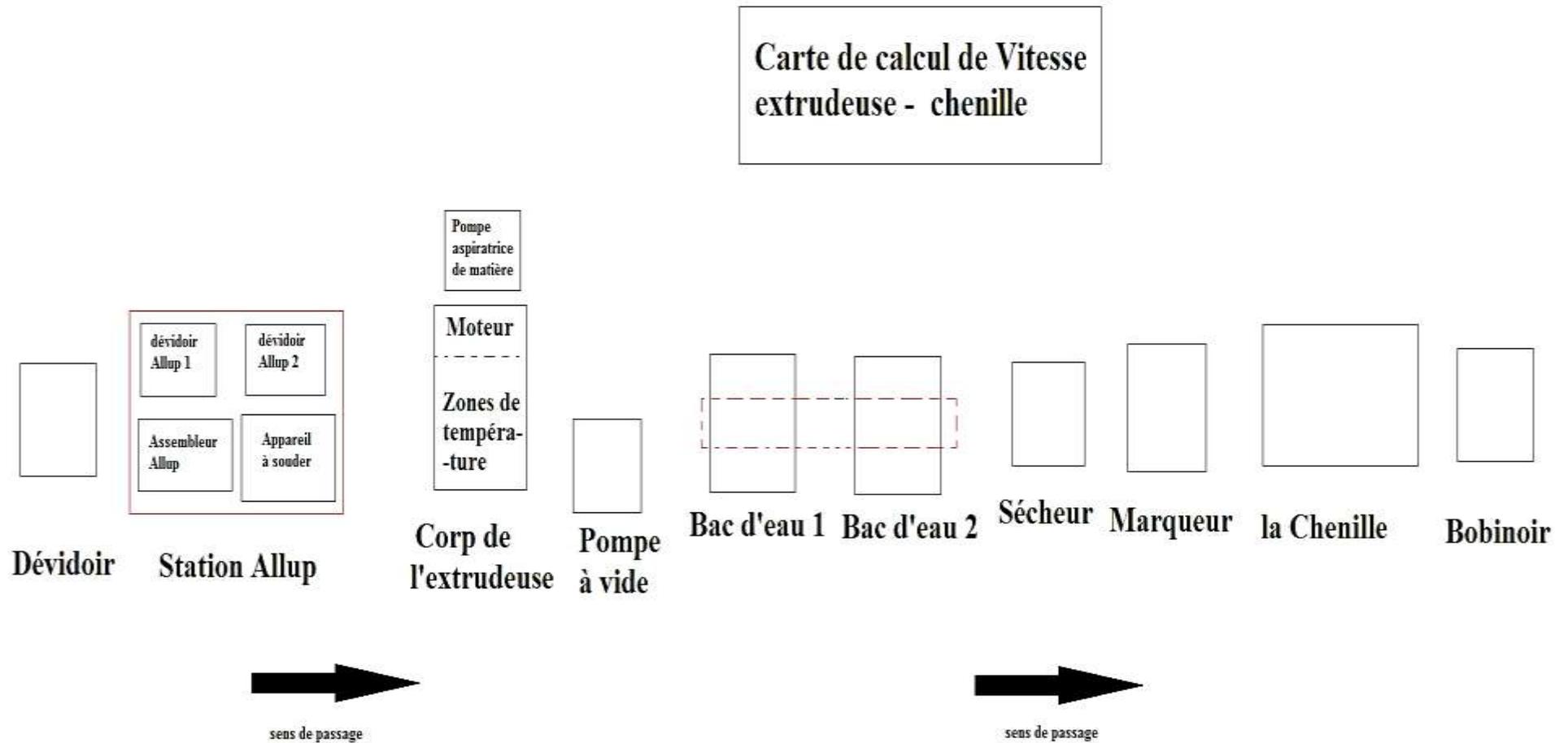


Figure 3-6 : présentation globale de la chaîne

### 3.5 Le fonctionnement de la chaîne :

Le câble assemblé sort du bobinoir et passe par la station Allupe où on met l'isolation électromagnétique, un fil de continuité du courant et un fil de déchirement. Tout cet ensemble passe par la trempette de formage pour le couvrir par la bonde ALU/PE. Ensuite l'ensemble passe par l'extrudeuse pour prendre la gaine extérieure. Si le câble est de la série 98/99 l'ensemble passe par l'extrudeuse avec un porteur d'acier.

L'extrudeuse chauffe la matière de revêtement dans 4 zones de températures situées sur la même ligne afin de l'extruder vers sa tête. La tête garde la matière chauffée et la guide afin de former un tube pour couvrir le câble. Cette circulation de matière est assurée par la rotation du vis-sans-fin attaché au moteur de l'extrudeuse qui tourne par une vitesse synchronisée avec celle du tirage (chenille). Après cette étape le câble sera refroidit en passant par les 2 bacs d'eau où le premier est régulé en température afin d'éviter le choc thermique de la gaine.

A la fin du bac 2 il y a un sécheur pour sécher le câble. Ensuite il passe par le marqueur qui tape sur le son matricule. Après ça le câble sera bobiné par le bobinoir. Ces bobines après son contrôle de qualité seront prêtes à la livraison.

On mentionne que Les 2 vitesses de la chenille et l'extrudeuse sont calculées et synchronisées par une carte électronique

#### 3.5.1 Le dévidoir :

Il sert à porter la bobine qui contient le câble assemblé (semi fini) et assure une vitesse de rotation tout en gardant une certaine tension du câble fixe.

Le dévidoir se compose de 3 moteurs (un moteur pour varier la distance entre les deux armatures du dévidoir selon la largeur de la bobine, un moteur pour faire monter la bobine à certaine hauteur et un moteur pour le déplacement pas à pas afin d'éviter l'inclinaison du câble) et un frein pour garder la tension du câble constante. Tout cet ensemble est commandé par un automate SIEMENS S7-1200. **(Figure 3-7)**



Figure 3-7 : le dévidoir

#### 3.5.2 La station Allupe :

Dans cette zone, on met les accessoires du câble, elle est constituée de 2 dévidoirs (le premier porte la bonde ALU/PE et le deuxième porte le filin de déchirement) et un appareil à souder

qui sert à coller les 2 bornes de la bonde ALU/PE sur l'ensemble de l'assemblage et le filin de déchirement. (**Figure 3-8**)



Figure 3-8 : la station Allupe

### 3.5.3 L'Extrudeuse :

L'extrudeuse a comme rôle de mettre la gaine extérieure du câble. Elle est constituée de 2 parties principales :

#### 3.5.3.1 Le moteur de l'extrudeuse :

Il est réglé en vitesse afin d'asservir le débit de la matière. Ce moteur fait tourner le **Vis-sans-fin** pour que la matière puisse circuler à travers les zones de températures. La matière de revêtement est assurée par le Con\_evateur. (**Figure 3-9**)

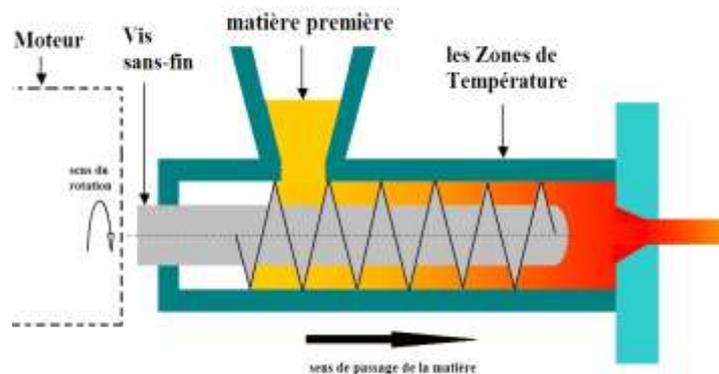


Figure 3-9 : principe de fonctionnement de l'extrudeuse

#### 3.5.3.2 Les zones de températures :

On a 7 zones de températures qui sont réglées pour assurer une augmentation de température graduelle de la matière. 3 zones sont au niveau de la tête seulement et 4 zones au niveau du corps de l'extrudeuse qui ont, en plus des résistances, des ventilateurs. Tous les résistances et ventilateurs fonctionnent en tout ou rien (TOR). (**Figure 3-10**)



Figure 3-10 : l'extrudeuse

#### 3.5.4 La Pompe à vide :

Une pompe qui peut être branchée au niveau de la tête de l'extrudeuse qui sert à éliminer le vide d'air entre la gaine et le câble, afin d'avoir un câble bien fini. (**Figure 3-11**)



Figure 3-11 : l'image de la pompe à vide

#### 3.5.5 Les Bacs d'eau :

2 bacs d'eau liés aux 2 réservoirs d'eau pour le refroidissement du câble, Où chaque bac d'eau est muni d'une pompe qui assure la circulation d'eau dans son bac. Le premier bac est régulé en température pour éviter le choc thermique. (**Figure 3-12**)



Figure 3-12 : les bacs d'eau

### 3.5.6 Le Sécheur :

Le sécheur est un appareil qui sert à sécher et à enlever les gouttes d'eau après le refroidissement du câble. (**Figure 3-13**)



Figure 3-13 : l'image du sécheur

### 3.5.7 Marqueur :

Le marqueur est un appareil qui sert à mettre le matricule sur le câble (la date, le nom de l'entreprise, la série ...etc.) cette étape vient juste avant le bobinage du câble. (**Figure 3-14**)



Figure 3-14 : le marqueur

### 3.5.8 La chenille :

C'est elle qui tire le câble de toute la chaîne, elle est alimentée par deux sources de puissances (tension pour le moteur de tirage et pression pour les vérins d'attraction), elle est régulée en vitesse. Son rôle est de tirer le câble sur toute une distance afin d'éviter de l'endommager. (Figure 3-15)



Figure 3-15: la chenille

### 3.5.9 Le Bobinoir :

A un rôle inverse du dévidoir, tel qu'à la place du frein il y a un moteur qui fait tourner la bobine du câble. Le bobinoir met le câble sur une bobine finale où il sera prêt à être livré au client. Il assure une vitesse de rotation tout en gardant une certaine tension du câble fixe. (Figure 3-16)



Figure 3-16: le bobinoir

### 3.5.10 La carte électronique des vitesses :

Cette carte calcule et transmet les consignes des vitesses du moteur de l'extrudeuse et de la chenille sous forme de 2 signaux (de 0 à 10 Volt). Où ces signaux sont envoyés aux variateurs des vitesses de ces 2 moteurs. (Figure 3-17)



*Figure 3-17 : la carte électronique des vitesses*

### **3.6 Conclusion :**

Dans ce chapitre on a présenté l'entreprise CATEL et les câbles de télécommunication en cuivre. On a aussi expliqué le fonctionnement de la chaîne de gainage BMN120 dont on a cité les différentes parties de la chaîne en définissant le rôle de chaque partie.

## **Chapitre 4 : Conception et développement du système d'automatisation**

## 4.1 Introduction :

Après avoir présenté la ligne de gainage et déceler ses manquements, nous allons y apporter les solutions correctives et amélioratrices qui répondent au cahier des charges donné par l'entreprise d'où l'automatisation de l'ensemble du système. Et cela parce que la fonction globale de tout système automatisé c'est de conférer une valeur ajoutée à ce terme.

## 4.2 Définition des E/S de la ligne de gainage :

Tout d'abord, nous allons décrire toutes les entrées et les sorties de la ligne de gainage, pour y arriver, nous allons décrire les E/S de chaque composant de la ligne comme suit :

**Dévidoir, Marqueur et Bobinoir :** on a besoin d'une entrée à l'API pour voir l'état de fonctionnement du dévidoir (enclenché/déclenché), c'est la même chose pour le bobinoir et le marqueur.

**Appareil à souder, Pompe à vide, Pompe réservoir et Sécheur :** on a besoin de deux entrées à l'API (bouton pour enclencher / déclencher l'appareil à souder et une entrée pour voir l'état (marche/Arrêt)) et une sortie à le contacteur de puissance de l'appareil. Pompe à vide, pompe réservoir 2 et sécheur ont même E/S comme l'appareil à souder.

**Con\_evateur :** on a besoin d'un bouton pour mise le Con\_evateur en service/hors service et deux entrées logiques (détecteur niveau bas et détecteur niveau haut) pour enclencher/déclencher la pompe d'aspiration, une entrée pour voir l'état de la pompe (marche/Arrêt), une sortie à le contacteur de puissance du con\_evateur.

**Zones de chauffage :** il y a 4 zones, on utilise un commutateur de 3 états pour mettre les zones en service, pour forcer le refroidissement des zones, pour mettre les zones hors service, on a besoin aussi d'une entrée analogique (thermocouple types J) pour chaque zone afin de mesurer la température de la zone, une entrée logique pour voir l'état de chaque actionneur, une sortie pour enclencher/déclencher la résistance de chauffage et une sortie pour enclencher/déclencher le moteur du ventilateur de chaque zone.

**Têtes de chauffage :** il y a 3 zones au niveau de la tête, on utilise un interrupteur de 2 états pour mettre ces zones en service ou hors service, on a besoin aussi d'une entrée analogique (thermocouple type J) pour chacune des zones enfin de mesurer sa température, une entrée logique pour voir l'état de chaque partie et une sortie à chaque actionneur des zones.

**Moteur extrudeuse et moteur chenille (de tirage) :** chaque moteur a son variateur de vitesse qui va communiquer avec l'API, donc on utilise un interrupteur de 2 états pour mettre le moteur d'extrudeuse en service/hors service, idem pour le moteur de tirage , deux entrées analogiques pour recevoir la mesure de vitesse de chaque moteur , deux entrées logique pour recevoir l'état de chaque moteur(marche/Arrêt) , deux sorties analogiques pour transmettre une consigne de vitesse à chaque variateur.

**Réservoir d'eau 1 :** se compose de deux systèmes, le système de chauffage et le système de pompage. Le système de chauffage prend la même interface d'E/S que chacune des têtes, le système de pompage prend la même interface d'E/S que l'appareil à souder.

On utilise aussi des boutons d'arrêt d'urgence au niveau du dévidoir, le bobinoir, la chenille et l'armoire de l'extrudeuse qui sont représenté par une seule entrée.

### 4.3 Mode opératoire :

L'opérateur doit suivre les étapes suivantes afin de mettre la chaîne BMN120 en service :

- Vérifier et mettre les armoires électriques sous tension en actionnant le sectionneur de la position 0 à 1.
- Préparer l'installation : montage de la vis sans fin, montage de la tête extrudeuse, placer le guide câble et la filière de gainage adéquat et placer la trempette de formage adéquat sur son port.
- Entrer les températures de chauffage pour chaque zone puis mettre l'extrudeuse en chauffe en actionnant sur les boutons « chauffage » et « chauffage\_tete ».
- Placer la bobine du câble à gainer dans le dévidoir et une bobine vide dans le bobinoir suivant le câble à gainer.
- Placer la bobine de la bonde ALU/PE et le filin de reconnaissance sur ses ports.
- Placer le marqueur adéquat sur son port.
- Placer la bobine de toron d'acier adéquat, si le câble à gainer est de la série 98 ou 99.
- Faire passer le câble, le filin de reconnaissance et la bonde ALU/PE à la trempette et la filière de formage.
- Vérifier si la bonde ALU/PE est bien fermée puis attacher la fin du câble à gainer au câble de tirage.
- Faire passer le fil de tirage tout au long du bac de refroidissement, chenille et en fin le bobinoir.
- Régler la pression d'air de la chenille.
- Faire fonctionner la pompe d'aspiration de la matière.
- Mettre le sécheur en marche.
- Ouvrir les robinets du bac de refroidissement et faire fonctionner les pompes.
- Mettre la pompe à vide en marche si le câble est de la série 98 ou 99.
- Démarrer la machine en actionnant sur les deux boutons (« Moteur\_Ext » et « Moteur\_tirage »).
- Augmenter la vitesse de la ligne, l'extrudeuse et de tirage en parallèle jusqu'à atteindre le diamètre et l'épaisseur voulu suivant le câble à gainer.

### 4.4 Cahier des charges :

**Appareil à souder :** au début l'appareil est à l'arrêt, si on tourne le levier « App\_à\_souder » en ON, l'appareil passera à l'état marche. Le voyant qui indique si l'appareil marche/arrêt s'allume. Si on tourne le levier en OFF ou on appuie sur l'arrêt d'urgence « AU », l'appareil passera à l'arrêt.

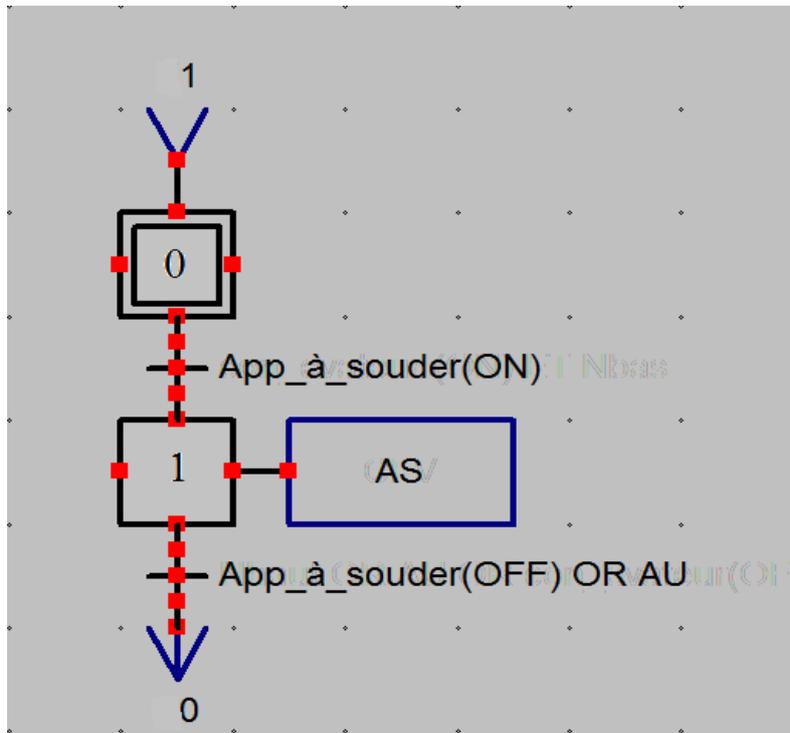


Figure 4-1 : Grafcet de l'appareil à souder

**Le Con\_evateur :** au début la pompe d'aspiration de la matière est à l'arrêt. Si on tourne le levier « con\_evateur » en « ON », le con\_evateur mise en service. Si la matière au-dessous du niveau bas la pompe passe à l'état marche. La case qui indique si la pompe marche s'allume. Si on tourne le levier en « OFF » ou la matière au-dessus du niveau haut ou on appuie sur AU la pompe d'aspiration passera à l'arrêt.

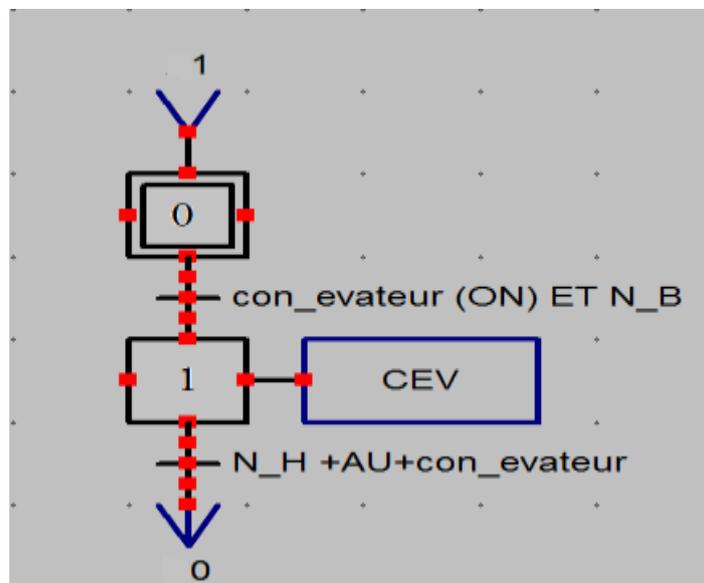


Figure 4-2; Grafcet du con\_evateur

**Pompe à vide :** si le câble est de la série 99 ou 98, il faudra d'utiliser la pompe à vide. Si on tourne le levier « pompe à vide » en « ON » la pompe passe à l'état marche. Le voyant qui indique si la pompe marche s'allume. Si on tourne le levier en OFF ou on appuie sur « AU » la pompe passera à l'arrêt.



**Chauffage d'une tête :** On joue sur l'enclenche et le déclenche de la résistance de chauffage pour réguler la température. Initialement le système est hors service, l'opérateur doit entrer la consigne de température. Si on clique sur le bouton « chauffage\_tete » et la température est inférieure à «  $T_c - 5^{\circ}\text{C}$  », la résistance passera à l'état marche. On utilise une case ou un graphique pour indiquer si la résistance est allumée.

Si on clique sur le bouton « Arrêt\_tête » ou la température est supérieur ou égal à «  $T_c_t$  » ou on appuie sur « AU », la résistance passera à l'arrêt.

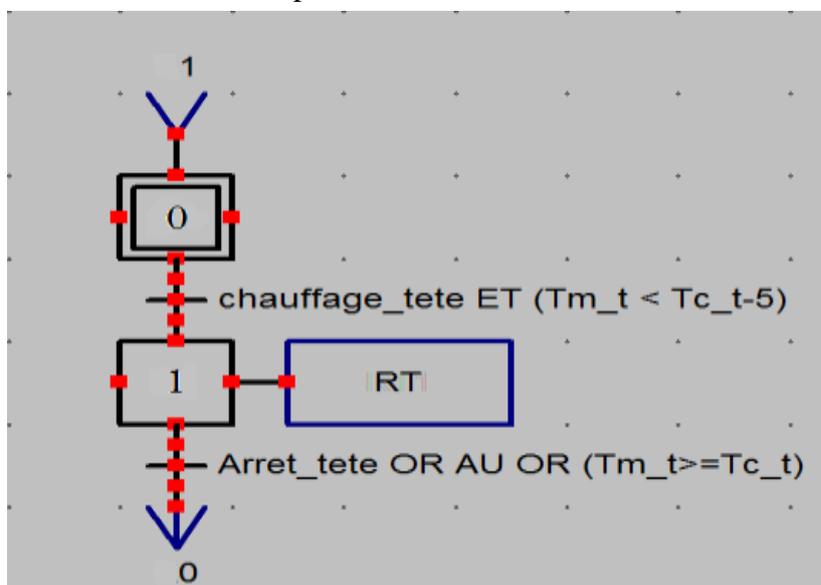


Figure 4-5 : chauffage de la tête

**Réservoir 1 (bac d'eau) :** il faut chauffer l'eau à la température voulue et circuler l'eau dans un bac. Si on tourne le lever « chauffage\_eau » en « ON » et la température est inférieure à «  $T_c_t - 5^{\circ}\text{C}$  », la résistance de chauffage passera à l'état marche. On utilise un graphique pour indiquer si la résistance est allumée. Si on tourne le lever en « OFF » ou la température est supérieur ou égal à «  $T_c_t$  » ou on appuie sur "AU", la résistance passera à l'arrêt.

Si on clique sur un bouton (ou un graphique), la pompe 1 sera mettre en service, on réserve un voyant pour indiquer si la pompe est allumé. Mais si on clique une autre fois sur le bouton ou on appuie sur « AU », la pompe passera à l'arrêt.

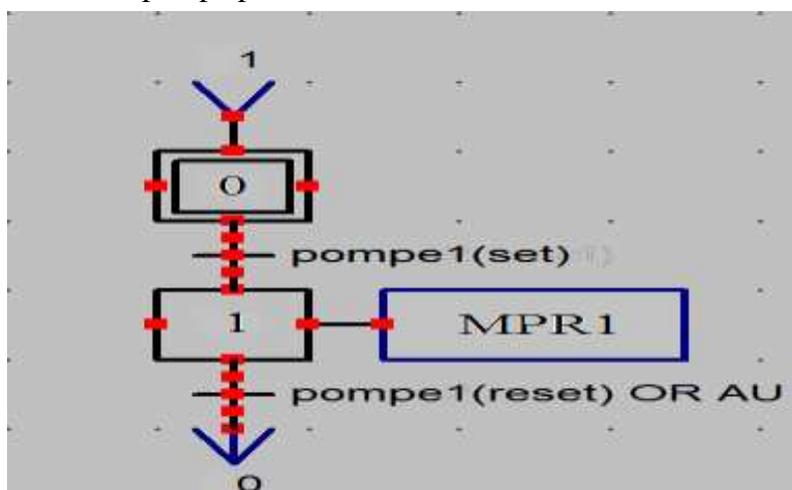


Figure 4-6 : la pompe du bac d'eau 1

**Réservoir 2** : on a besoin de circuler l'eau dans un autre bac d'eau d'une température ambiante. Si on clique sur un bouton (un graphique), la pompe 2 se mettra en service. On réserve un voyant pour indiquer si la pompe est allumée. Mais si on clique une autre fois sur le bouton ou on appuie sur « AU », la pompe passera à l'arrêt.

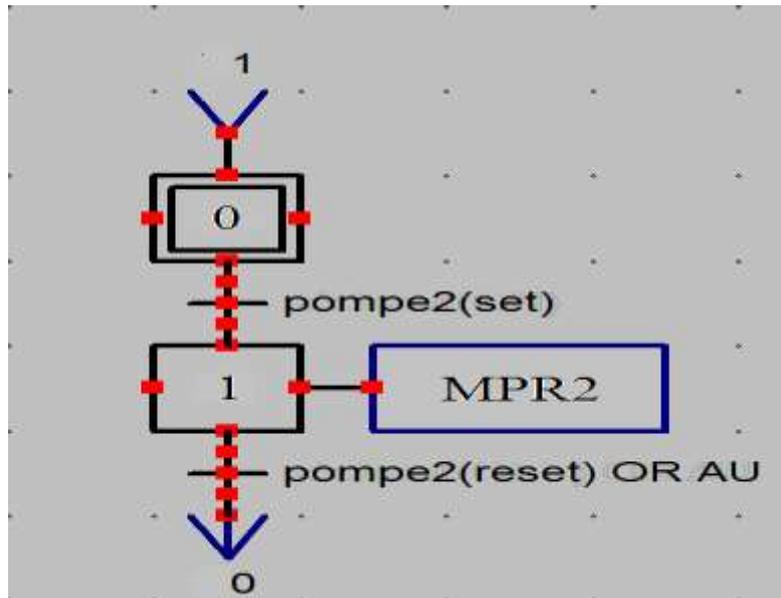


Figure 4-7 : la pompe du bac d'eau 2

**Le sécheur** : quand le câble quitte le bac d'eau 2, il faut sécher le câble à fin de marquer la matricule du câble. Si on tourne le lever « sécheur » en « ON » le sécheur passera à l'état marche. On utilise un voyant qui indique si le sécheur est allumé. Si on tourne le lever en « OFF » ou on appuie sur « AU », le sécheur passera à l'arrêt.

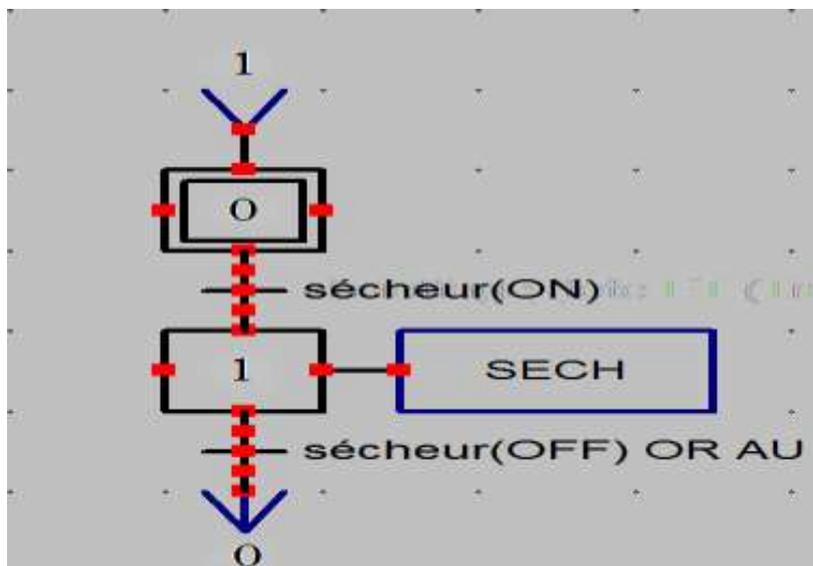


Figure 4-8 : sécheur

**Moteur d'extrudeuse et moteur de tirage** : il y a une relation de synchronisation entre ces deux moteurs. L'opérateur varie la vitesse d'un par rapport à l'autre jusqu'à atteindre un diamètre acceptable. Chaque moteur a un levier à fin de l'enclenché ou le déclenché.

On a deux régimes de marche pour le moteur d'extrudeuse, marche de démarrage et marche de production. On utilise le mode marche de démarrage pour purger l'extrudeuse jusqu'à atteindre une matière prise à l'utilisation. Puis on lance le mode marche de production ou on

met les deux moteurs en marche et on varie leur vitesse en parallèle jusqu'à atteindre le diamètre et l'épaisseur voulu en agissant sur « consigne\_V\_Ext » et « consigne\_V\_tirage ». Puis on augmente la vitesse de production en agissant sur « consigne\_V\_master » qui fait varier les deux vitesses en synchronisation (en sauvegardant le même rapport de synchronisation).

Si on tourne le lever « Moteur\_Ext » en « ON » et l'extrudeuse est chauffé «  $T_m > T_{min}$  », le moteur d'extrudeuse passera en marche. Si on appuie sur « AU » ou une des zones est en panne ou on tourne le lever « Moteur\_Ext » en « OFF », le moteur d'extrudeuse passe à l'arrêt et la valeur de « consigne\_V\_Ext » devient nulle. Si on tourne le lever « Moteur\_tirage » en « ON », le moteur de tirage passe en marche. Maintenant on peut varier la vitesse « consigne\_V\_tirage ». Si on tourne le lever « Moteur\_tirage » en « OFF » ou on appuie sur « AU » le moteur de tirage passe à l'arrêt. On utilise des voyants pour indiquer si les moteurs sont en marche/arrêt.

En cas de panne dans le mode de production normal qui a lancé par (dévidoir, bobinoir, reservoir1, reservoir2, chenille, sécheur, appareil à souder) les vitesses de consigne reviennent à zéro.

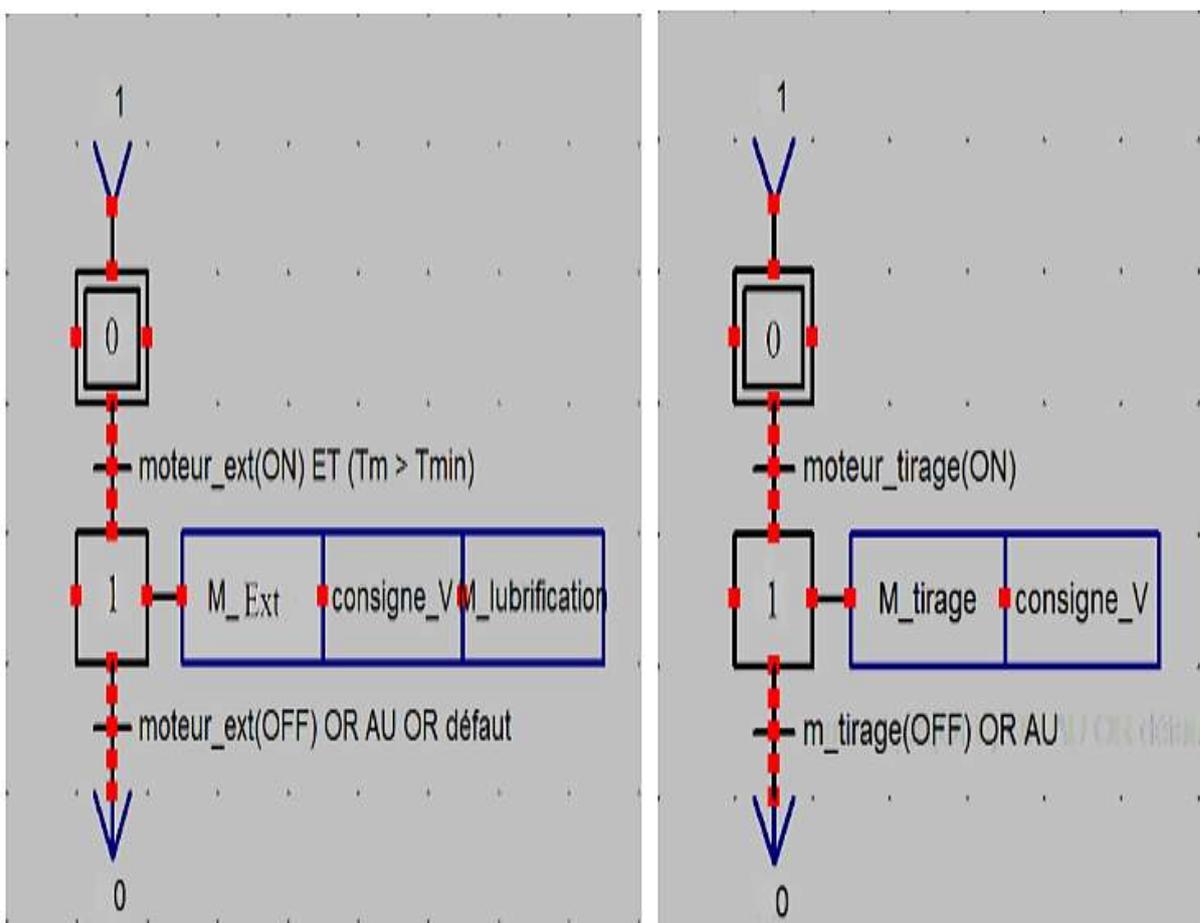


Figure 4-9 : grafcet moteur d'extrudeuse et moteur de tirage

## 4.5 Création du projet TIA portal :

### 4.5.1 Objet API :

Après la création d'un projet TIA portal (revoir chapitre 2), on ajoute un appareil API, la configuration des matériels, les mnémoniques, ainsi que l'écriture du programme.

Tout d'abord, nous citons les critères suivants pour le choix d'un API :

- La mémoire de travail et la vitesse de traitement des instructions du programme : il faut que la mémoire du travail de l'API choisi supporte la taille du projet et la vitesse doit être suffisante pour gérer toutes les tâches du système d'automatisation sans retard et sans décalage.

Exemple : CPU 314 (Mémoire de travail de 96 ko, 0.1 ms/kilo-instructions)

- Nombre d'entrées/sorties : il faut que le châssis de la CPU choisi puisse prendre en charge tous les modules d'E/S nécessaires pour notre application.

Exemple : CPU 314 (configuration multi rangée pouvant comporter jusqu'à 32 modules)

- La communication : L'automate doit pouvoir communiquer avec les autres systèmes de commande (API, supervision ...) et offrir des possibilités de communication par Profibus, MPI, Profinet...etc.

- La disponibilité du matériel (API) : Sur le marché, l'existence de la documentation et le savoir-faire du personnel sur Le matériel.

#### 4.5.2 La configuration matérielle :

Après l'insertion de la CPU 314 nous configurons le matériel utilisé :

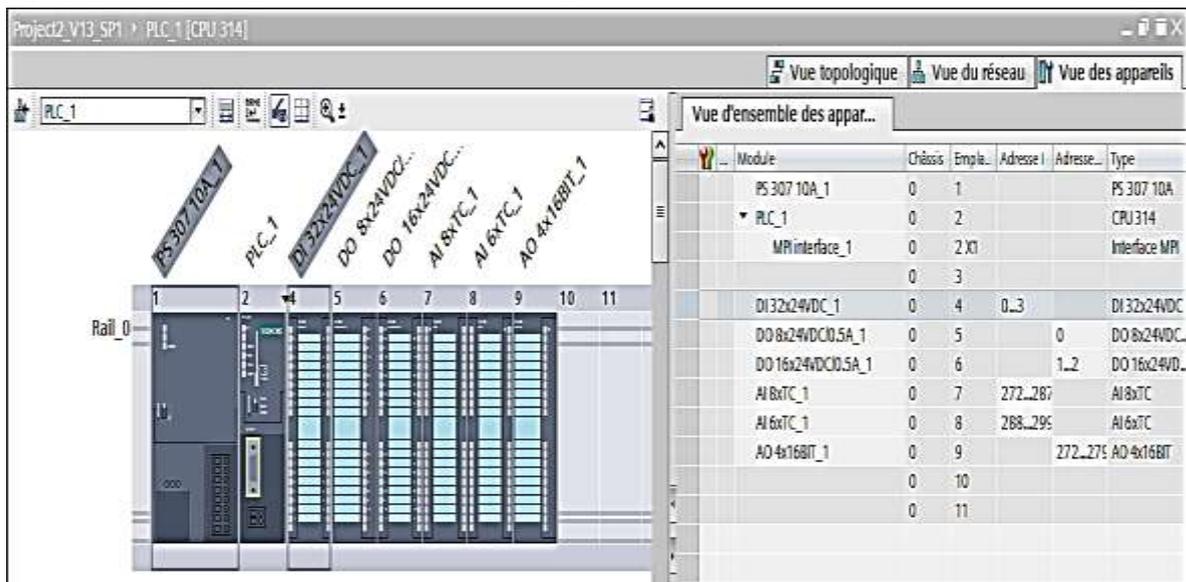


Figure 4-10 ; configuration matérielle

Il est préférable que le module des entrées analogiques a comme entrée le thermocouple de type J.

#### 4.5.3 La hiérarchie du programme :

On a subdivisé le système global en plusieurs sous-systèmes, Pour mieux gérer le système global, C'est-à-dire on a créé des blocs fonctionnels (FB) où chacun possède des données contenues dans le bloc de données (DB) associé. Ceci permet de mieux tester et déboguer les programmes.

Notre système se base sur 4 blocs fonctionnels essentiels (TETE « FB2 », TOR « FB3 », ZONE « FB1 » et vitesse « FB4 »).

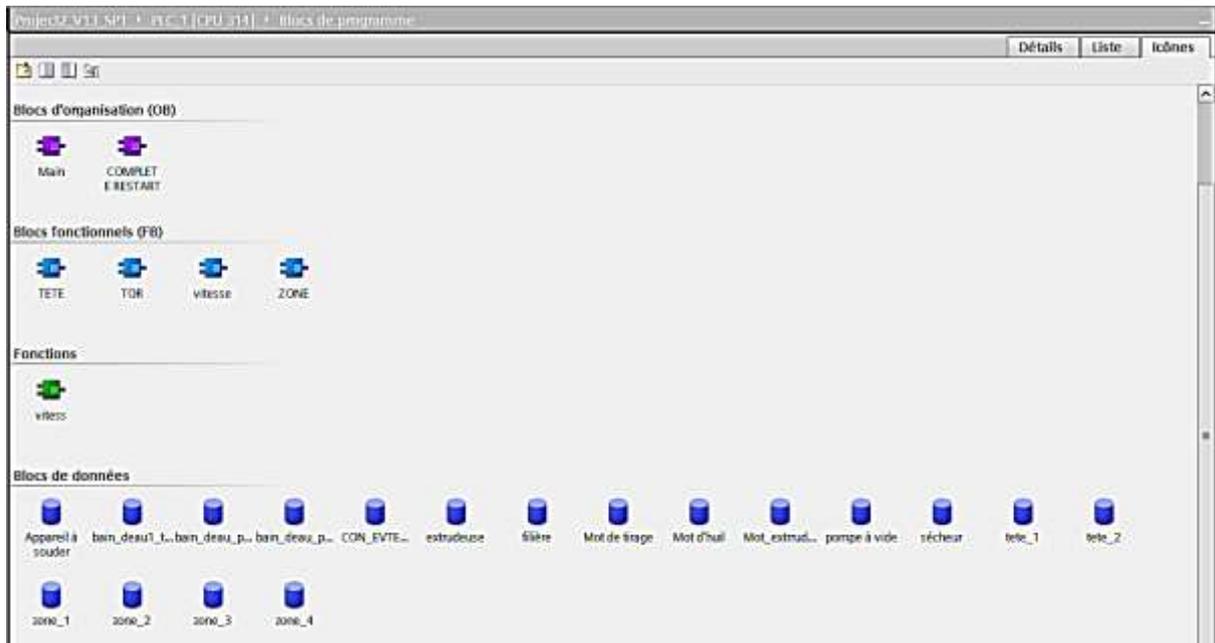


Figure 4-11 : La hiérarchie du programme

**4.5.4 Écriture du programme :**

D’après l’étude qu’on a faite sur la chaîne de gainage, on a conclu que ce système peut se diviser à des sous-systèmes, donc pour programmer ce système on va procéder la programmation structurelle.

Avant de pouvoir commencer l’écriture du programme, nous devons définir au préalable tous les mnémoniques.

Dans la partie programmation on utilise le langage de programmation graphique « CONT ». Ce dernier permet de suivre le trajet du courant entre les barres d’alimentation, sauf la fonction « vitesse FC1 » qui est écrite en SCL (langage de programmation structuré) pour gérer la relation de synchronisation entre le moteur d’extrudeuse et le moteur de tirage.

**4.5.5 Exemples de programmation :**

**Programmer le grafcet du sécheur :** chaque étape est représentée par une case mémoire « memento », une entrée logique 1 à la bobine S mise le memento à 1et une autre entrée logique 1 à la bobine R va mettre ce memento à 0 (SET et RESET).

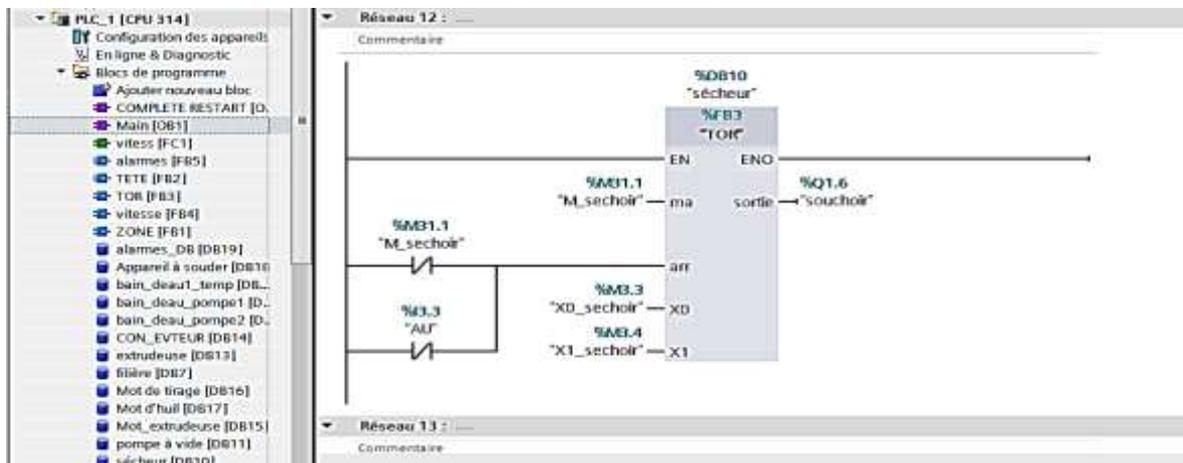


Figure 4-12 : appelle du bloc fonctionnel par le bloc de données sécheur (DB10)

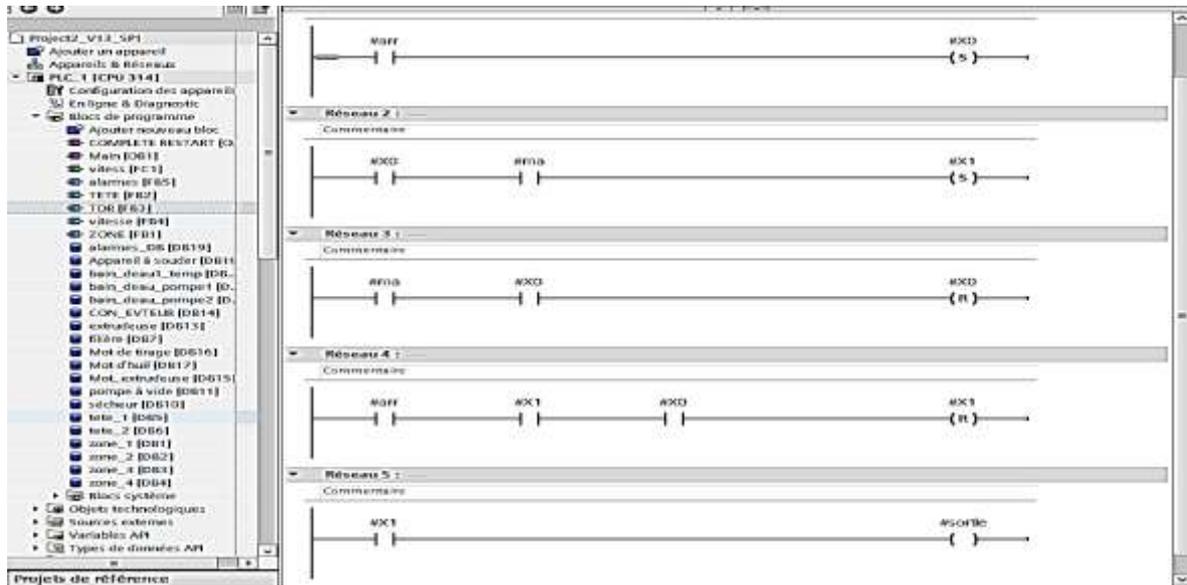


Figure 4-13 : le bloc fonctionnel TOR en langage CONT (Ladder)

4.5.6 Objet HMI :

D’abord nous avons défini le type du pupitre sur lequel les informations seront transmises, pour notre application nous avons utilisé un pupitre OP (poste opérateur) de type multi panels MP 377 15 tactile, lié à l’automate par une liaison MPI.

4.5.6.1 La description des vues :

Notre projet de supervision est structuré de manière à pouvoir visualiser l’ensemble de ligne de gainage.

**La vue d’accueil :** C’est une vue de présentation de projet, elle permet l’accès à la navigation entre les différentes vues développées dans cette solution de supervision et cela grâce à un ensemble de boutons configurés sur celle-ci. En cliquant sur chaque bouton, on aura accès à la vue correspondante. Cette vue est représentée dans la figure suivante :



Figure 4-14 : vue d'accueil

**La vue principale :** c'est la vue la plus utilisée. Elle affiche l'état de fonctionnement de tous les éléments de la ligne de gainage (Marche/Arrêt), contient des boutons, des leviers pour l'enclenchement ou le déclenchement de l'appareil à souder, la pompe à vide, le con\_evateur, la pompe et la résistance du bac d'eau 1, la pompe du bac d'eau 2, le sécheur, le moteur d'extrudeuse et de tirage et des champs d'E/S pour varier la vitesse de production ou de crée le rapport de synchronisation entre les deux moteurs.

Au sein de cette vue, on a le droit d'entrer les consignes des vitesses et voir les vitesses réelles. On peut aussi entrer la consigne de température du réservoir d'eau 1 et voir sa valeur réelle. De plus, il y a une vue d'alarme pour contrôler les erreurs et les warning de fonctionnement du système d'automatisation.

De cette vue on peut passer à la vue précédente ou bien à la vue suivante ou on clique sur le graphique de l'extrudeuse pour passer à la vue « les zones de chauffage ».



Figure 4-15 : vue principale

**La vue des zones de chauffage :** on utilise cette vue pour entrer les consignes de température des 4 zones et des 3 partitions de la tête et de voir ses valeurs réels.

On a utilisé des graphiques pour voir l'état de fonctionnement des actionneurs de chaque zone et chaque partie de la tête (résistances de chauffage, ventilateurs de refroidissement).

On utilise les boutons (chauffage, refroidissement, Arrêt) pour enclencher et déclencher les actionneurs des 4 zones de chauffage et les boutons (chauffage\_tete, Arrêt\_tete) pour enclencher et déclencher les 3 parties de la tête.

Il y a aussi une vue d'alarme pour contrôler les erreurs et les warning de fonctionnement du système d'automatisation. De cette vue on peut passer à la vue précédente (la vue principale) ou bien à la vue suivante (la vue de l'archive des alarmes et de la recette).

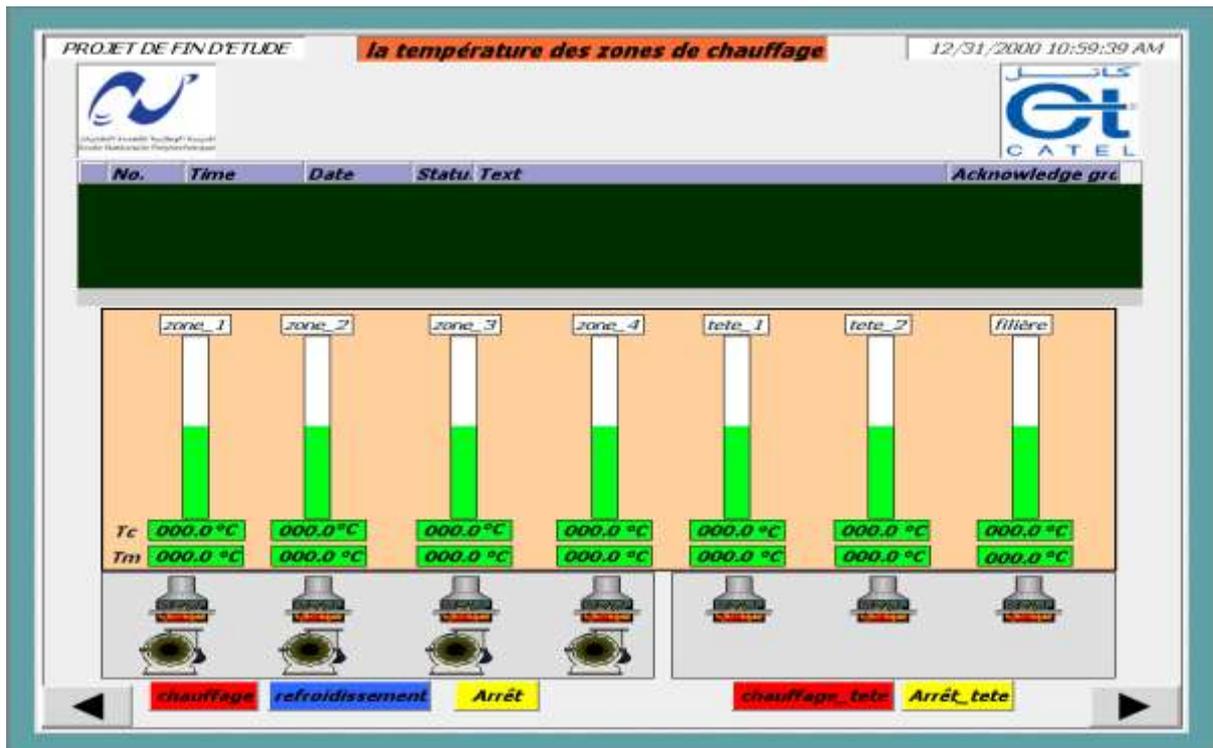


Figure 4-16 : vue zones de chauffage

**La vue de l'archive des alarmes et la recette :** on utilise dans cette vue deux fenêtres, la fenêtre « vue de la recette » qui nous permet d'entrer quelques paramètres du câble à gainer (largeur de la bonde ALU/PE, l'épaisseur de gainage, diamètre du câble gainer, les températures de chauffage de la matière, température max et min de chauffage de la matière, la vitesse max de ligne, ... etc.). La fenêtre « vue d'alarmes » qui nous permet de voir l'archive des alarmes (alarmes système et alarmes utilisateur).

De cette vue, on peut revenir à la vue des zone de température ou à la vue d'accueil.

Eléments		Enregistrements								
	Nom	Nom d'affichage	Variable	Type de données	Longueur d...	Valeur par défaut	Valeur minimale	Valeur maximale	Décimales	Texte d'aide
	Element_recette_1	largeur bonde ALU/PE	largeur_ALUPE	Int	2	0	-32768	32767	0	
	Element_recette_2	Epaisseur	Epaisseur	Int	2	0	-32768	32767	0	
	Element_recette_3	Diametre sur gain	Diametre_max	Int	2	0	-32768	32767	0	
	Element_recette_5	Tmin	T_min	Real	4	0	-3,402823466E+38	3,402823466E+38	0	la température min de la matiere
	Element_recette_6	Tmax	T_max	Real	4	0	-3,402823466E+38	3,402823466E+38	0	la température max de la matiere
	Element_recette_7	TZ1	Tc_1	Real	4	0	-3,402823466E+38	3,402823466E+38	0	la température de la zone 1
	Element_recette_8	TZ2	Tc_2	Real	4	0	-3,402823466E+38	3,402823466E+38	0	la température de la zone 2
	Element_recette_9	TZ3	Tc_3	Real	4	0	-3,402823466E+38	3,402823466E+38	0	la température de la zone 3
	Element_recette_10	TZ4	Tc_4	Real	4	0	-3,402823466E+38	3,402823466E+38	0	la température de la zone 4
	Element_recette_11	TT1	TC_t_1	Real	4	0	-3,402823466E+38	3,402823466E+38	0	la température de la tete 1
	Element_recette_12	TT2	TC_t_2	Real	4	0	-3,402823466E+38	3,402823466E+38	0	la température de la tete 2
	Element_recette_13	TT3	TC_t_3	Real	4	0	-3,402823466E+38	3,402823466E+38	0	la température de la tete 3
	Element_recette_14	Vmax	V_master_max	Real	4	0	-3,402823466E+38	3,402823466E+38	0	vitesse max du master
	Element_recette_15	TBE1	Tc_bain_deau1	Real	4	0	-3,402823466E+38	3,402823466E+38	0	température du bac 1
	Element_recette_16	V_firage/V_ligne	K_prod	Real	4	0	-3,402823466E+38	3,402823466E+38	0	rapport vitesse de tirage à la vitesse de proc.
	Element_recette_4	cable d'acier 110	9B_99	Bool	1	0			0	l'existence du cable d'acier
	<ajouter>									

Figure 4-17 : les éléments de la recette

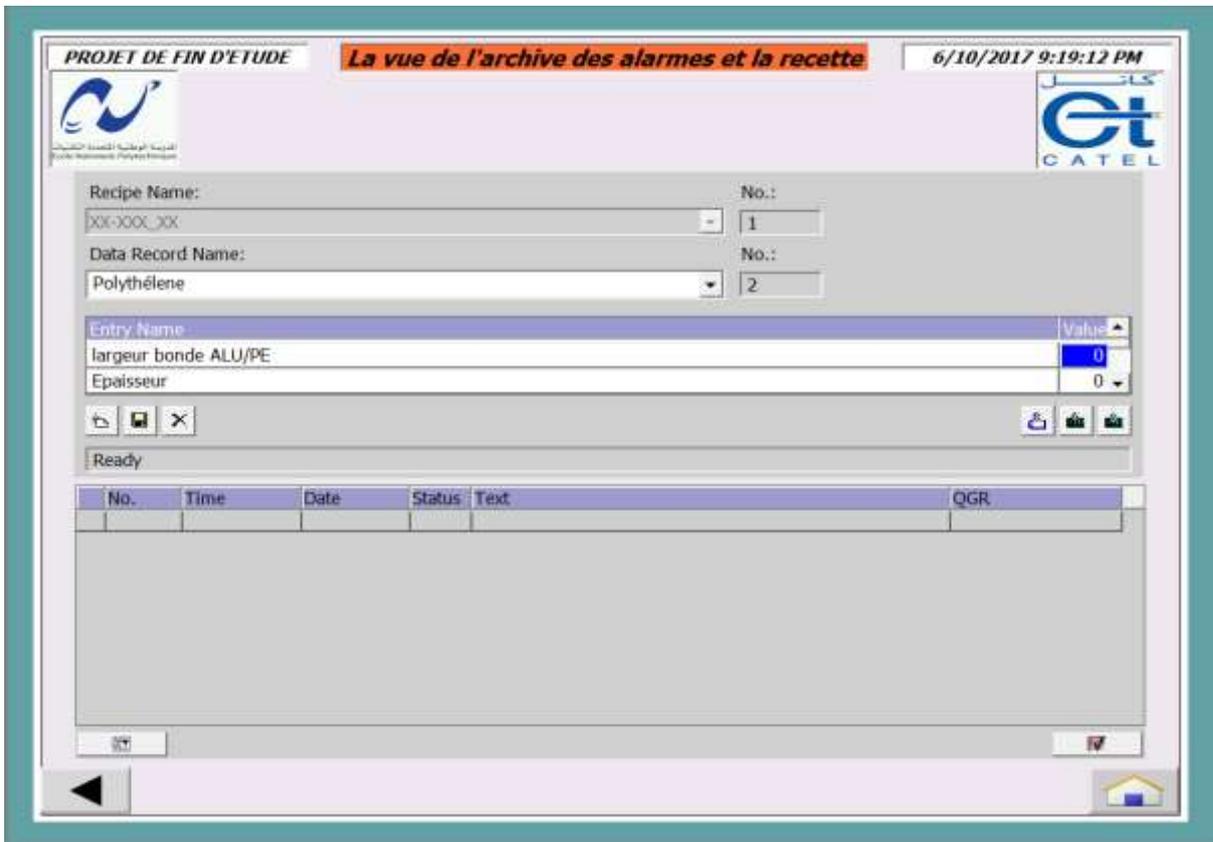


Figure 4-18 : vue de l'archive des alarmes et la recette

#### 4.6 Conclusion :

Dans ce chapitre, nous avons élaboré les graficets de l'installation et les programmé sous TIA portal, nous avons fait la supervision du procédé à travers un écran, et nous avons créé les différentes vues qui permettent de suivre l'évolution du fonctionnement de l'installation. Tout cela nous a offert une grande flexibilité de contrôle.

## **Conclusion générale**

Notre travail a pour objectif d'automatiser la ligne de gainage BMN120, qui fonctionne à base de la logique câblée où il y aura un manque de sécurité des éléments de l'installation et du personnel. Le travail élaboré permet l'opérateur de visualiser l'état de fonctionnement ou de commander tous les équipements de l'installation (les moteurs, les résistances de chauffage, les ventilateurs de refroidissement, ...etc.).

Nous avons fait appel à la technologie programmée (Automate programmable Industriel), dans notre cas, on a utilisé un API SIEMENS et le logiciel de programmation et de supervision TIA portal de SIEMENS.

On pourrait dire que remplacer les dispositifs classiques de contrôle par les Automates Programmables Industriels se révèle comme étant la solution qui subviendrait aux besoins du monde industriel, qui ne cesse de trop exigé pour ce qui est de qualité et quantité des produits.

On a présenté aussi des améliorations dans le cahier des charges pour assurer la sécurité de l'installation et aider l'opérateur à visualiser toute l'installation.

Cette réalisation nous a poussé à faire appel à toutes nos connaissances et aptitudes acquises durant notre formation et nous a permis d'appréhender les difficultés qu'on peut rencontrer tous les jours dans le monde professionnel et d'avoir une expérience pratique.

Dans des travaux futurs on pourra envisager :

- Remplacer les relais dynamiques sur les zones de températures qui fonctionnent en TOR par des gradateurs de puissance et implémenter des fonctions de régulation continues afin d'avoir un fonctionnement plus fluide.
- Proposer une communication moderne entre les variateurs de vitesses et l'automate afin de pouvoir visualiser les différentes performances du moteur (courant, vitesse, tension, couple ...etc.) directement sur le pupitre.
- Remettre en service le capteur de diamètre et faire une régulation sur cette base.
- Automatisation de l'étape de démarrage
- Optimisation du cycle de changement des bobines pour avoir un gain en temps et en production.
- Elargir le programme pour contrôler toute la chaîne de production en centralisant la commande sur un seul pupitre afin de permettre à l'opérateur de connaître l'état d'avancement du processus en temps réel et d'intervenir directement sur le pupitre de commande.

## Bibliographie

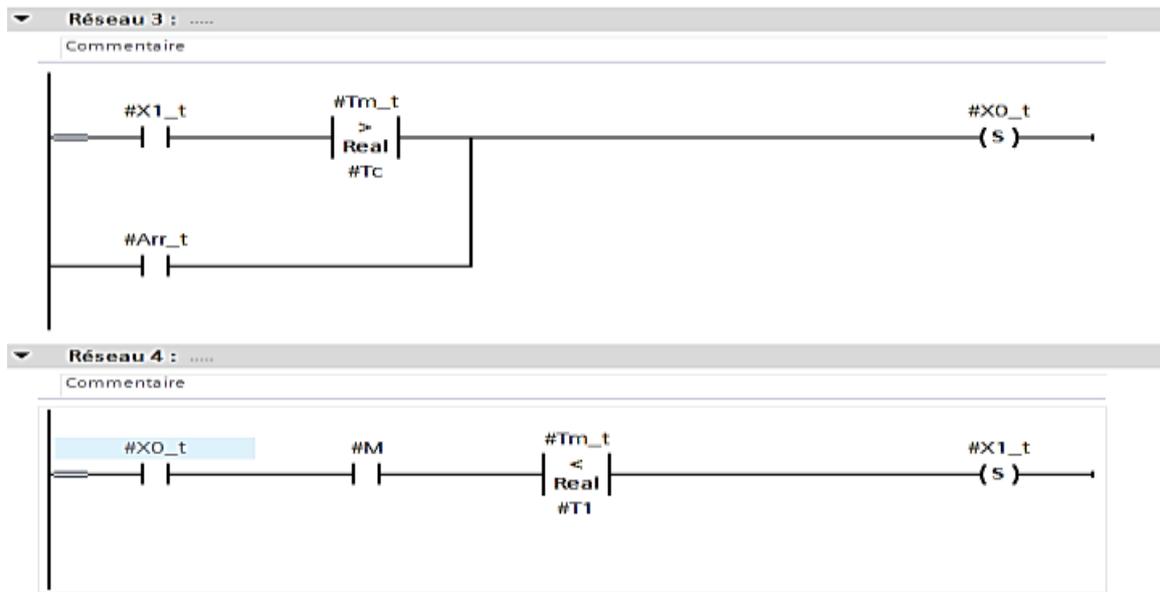
- [1] M. BERTRAND, «Automates programmables industriels,» *Technique de l'ingénieur*, vol. S 8 015.
- [2] G. MICHEL, les A.P.I Architecture et application des automates programmables industriels, Edition DUNOD, 1987.
- [3] P. JARGOT, «Langages de programmation par API, Norme IEC 1131-3,» *Technique de l'ingénieur*, vol. S 8 030.
- [4] W. BOLTON, Les automates programmables industriels, 5ème Edition, 2009.
- [5] SIEMENS, STEP 7 Professionnel V13.1, Réf. 6ES7822-1AA03-2YA6 SIMATIC, 2014.
- [6] SIEMENS, Système d'automatisation S7-400 caractéristique des CPU, Réf 6ES7498-8AA04-8CA0 SIMATIC, 2006.
- [7] «CATEL-DZ,» [En ligne]. Available: <http://www.catel-dz.com>.
- [8] D. D.DUPONT, «réalisation technologique du GRAFCET,» *Technique de l'ingénieur*, vol. S 8 032.
- [9] H.NUSSBAUMER, Informatique industrielle 4, Presses polytechniques romandes Suisse, 1987.
- [10] [En ligne]. Available: [www.siemens.com](http://www.siemens.com).

**Annexe A : les langages de programmation des APIs**

### A.1 Le langage LD :

Le langage Ladder Diagram (LD) est une représentation graphique d'équations booléennes combinant des contacts (arguments d'entrée) avec des bobines (résultats de sortie). Il permet la description de tests et la modification de données booléennes à l'aide des symboles graphiques placés dans un diagramme.

Les symboles graphiques LD sont organisés comme les éléments d'un schéma électrique à contacts. Et sont connectés à gauche et à droite aux barres d'alimentation verticales par des segments de liens. Chaque segment de lien peut prendre l'état booléen FALSE ou TRUE. Cet état est le même pour tous les segments directement connectés ensemble.



*Exemple de programmation sous le langage Ladder (LD)*

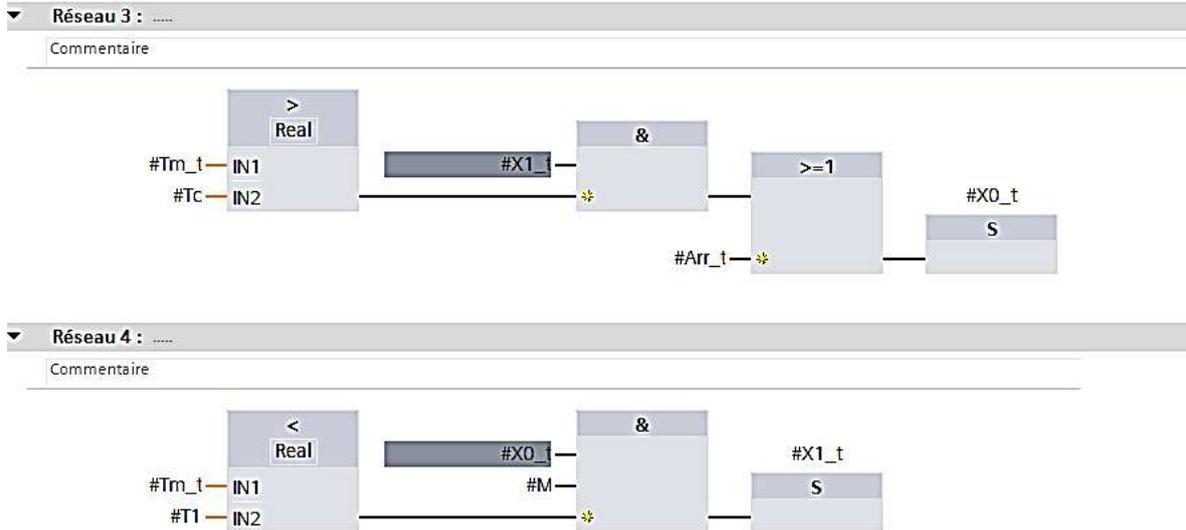
### A.2 Le Langage FBD :

Le diagramme FBD décrit une fonction entre des variables d'entrée et des variables de sortie. Une fonction est décrite comme un réseau de fonctions élémentaires. Les variables d'entrée et de sortie sont connectées aux boîtes fonctions par des arcs de liaison. Une sortie d'une boîte peut être connectée sur une entrée d'une autre boîte.

Les variables d'entrée du diagramme FBD doivent être connectées aux entrées des boîtes fonctions. Le type de chaque variable doit être cohérent avec le type de l'entrée de la boîte correspondante. Une entrée peut être une expression constante, une variable interne ou externe.

Les variables de sortie du diagramme FBD doivent être connectées aux sorties des boîtes fonctions. Le type de chaque variable doit être cohérent avec le type de la sortie de la boîte correspondante.

Une sortie peut être une variable interne ou de sortie, ou le nom du programme édité (pour les fonctions seulement). Quand le nom de la fonction éditée est utilisé comme sortie du diagramme, il représente une assignation de la valeur retournée par la fonction.

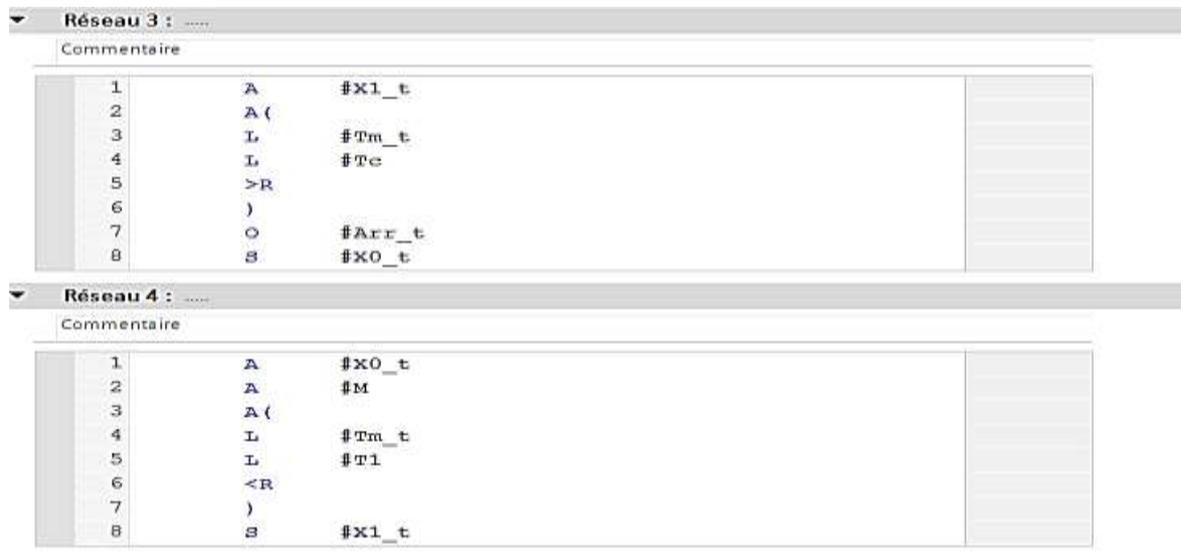


*Même exemple sous le langage (FBD)*

### A.3 Le langage IL :

Le langage IL, ou Instruction List, est un langage textuel de bas niveau. Les instructions travaillent 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 à son tour stocké dans le registre.

Un programme IL est une liste d'instructions. Chaque instruction doit commencer par une ligne nouvelle, et doit contenir un opérateur, complété éventuellement par des modificateurs et, si nécessaire pour l'opération, un ou plusieurs opérandes, séparés par des virgules (","). Une étiquette suivie de deux points (" :") peut précéder l'instruction.



*même exemple sous le langage (IL)*

### A.4 Le langage ST :

Le langage ST (Structured Text) est un langage textuel de haut niveau dédié aux applications d'automatisation. Ce langage est principalement utilisé pour implémenter des procédures complexes, difficilement modélisables avec les langages graphiques. Le langage ST peut être

utilisé pour la programmation des actions dans les étapes et les conditions associées aux transitions du SFC.

Un programme ST est une suite d'énoncés. Chaque énoncé se termine par un point-virgule (";"). Les noms utilisés par le code source (identificateurs de variables, constantes, mots-clés du langage, etc.) sont délimités par des séparateurs passifs (espace, fin de ligne ou tabulation) ou par des séparateurs actifs, qui ont une signification bien définie et qui jouent le rôle d'opérateurs (par exemple, le séparateur ">" indique une comparaison "Plus grand que").

```

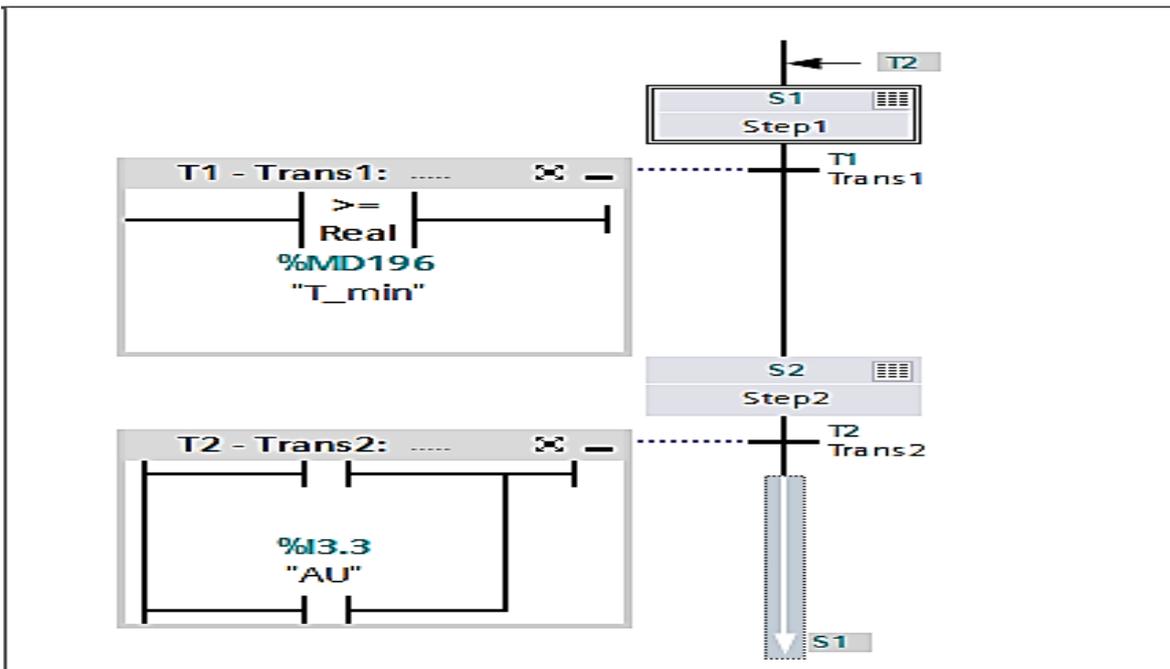
1 IF (#inc = true) THEN
2   #v_ext_p := #v_ext_in + #val_inc/déc;
3 ELSE
4   IF (#déc = true) THEN
5     #v_ext_p := #v_ext_in - #val_inc/déc;
6   ELSE
7     #v_ext_p := #v_ext_in;
8   END_IF;
9 END_IF;
10
11 IF ((#v_ext_in >= 0 AND #v_ext_in <= 100) AND (#v_tirage_in >= 0 AND #v_tirage_in <= 100) AND (#v_master_in >= 0 AND #v_master_in <= 100)) THEN
12   #v_ext_Out_reel := #v_ext_in * #v_master_in / 100;
13   #v_tirage_out_reel := #v_tirage_in * #v_master_in / 100;
14
15 END_IF;

```

*La fonction de synchronisation entre la vitesse de l'extrudeuse et celle de tirage*

#### A.5 Le GRAFCET (SFC) :

Le langage (sequential function chart), ou GRAFCET, 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.



*Exemple d'un grafcet simple*

**Annexe B : les alarmes système et les icônes de diagnostic**

**B.1 les alarmes système :**

Fonction système	Action
EditerAlarme	Déclenche l'événement "Loop-In-Alarm" pour toutes les alarmes sélectionnées.
EffacerTamponAlarmes	Cette fonction supprime les alarmes du tampon des alarmes du pupitre opérateur.
EffacerAlarmesTamponProtoolHeritage	Fonction comme "EffacerTamponAlarmes". Cette fonction système est présente pour des raisons de compatibilité et utilise la numérotation ProTool.
VueAlarmesEditerAlarme	Déclenche l'événement "Loop-In-Alarm" pour toutes les alarmes sélectionnées dans la vue des alarmes indiquée.
VueAlarmesAcquitterAlarme	Acquitte les alarmes sélectionnées dans la vue des alarmes indiquée.
VueAlarmesAfficherTexteAide	Affiche l'info-bulle configurée pour l'alarme qui est sélectionnée dans la vue des alarmes indiquée.
AcquitterAlarme	Acquitte toutes les alarmes sélectionnées.
AfficherFenêtreAlarmes	Cette fonction système affiche ou masque la fenêtre des alarmes sur le pupitre opérateur.

**B.2 Les icones de diagnostic :**

icône	Signification
	Erreur matérielle dans le composant de niveau inférieur : Au moins un composant matériel de niveau inférieur contient des versions en ligne et hors ligne différentes (dans la navigation de projet uniquement).
	Erreur logicielle dans le composant de niveau inférieur : Au moins un composant logiciel de niveau inférieur contient des versions en ligne et hors ligne différentes (dans la navigation de projet uniquement).
	Les versions en ligne et hors ligne de l'objet sont différentes
	Objet disponible en ligne uniquement
	Objet disponible hors ligne uniquement
	Les versions en ligne et hors ligne de l'objet sont identiques

Icône	Signification
	La liaison vers une CPU est en cours d'établissement.
	La CPU n'est pas accessible sous l'adresse donnée.
	La CPU configurée et la CPU réelle ont des types non compatibles.
	Pendant l'établissement de la liaison en ligne à une CPU protégée, la boîte de dialogue pour le mot de passe a été interrompue sans saisie du mot de passe correct.
	Aucune défaillance
	Maintenance nécessaire
	Maintenance requise
	Erreur
	Le module ou l'appareil est désactivé.
	Le module ou l'appareil n'est pas accessible à partir de la CPU (s'applique aux modules et aux appareils sous la CPU).
	Il n'y a pas de données d'entrée ou de sortie disponibles, car le module ou le sous-module a verrouillé ses voies d'entrée ou de sortie.
	Aucune donnée de diagnostic disponible, car les données de configuration en ligne actuelles sont différentes de celles de la configuration hors ligne.
	Le module ou l'appareil configuré est incompatible avec le module ou l'appareil réel (s'applique aux modules et aux appareils sous la CPU).
	Le module configuré ne prend pas en charge l'affichage de l'état de diagnostic (valide pour les modules en dessous d'une CPU).
	La liaison est établie, mais l'état du module n'est pas encore déterminé ou est inconnu.
	Le module configuré ne prend pas en charge l'affichage de l'état de diagnostic.
	Erreur matérielle dans le composant de niveau inférieur : au moins un composant matériel subordonné présente une erreur matérielle. (n'apparaît comme icône indépendante que dans la navigation de projet)
Icône	Etat de fonctionnement
	MARCHE
	ARRET
	MISE EN ROUTE
	ATTENTE
	DEFAULT
	Etat de fonctionnement inconnu
	Le module configuré ne prend pas en charge l'affichage de l'état de fonctionnement