

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE**

ECOLE NATIONALE POLYTECHNIQUE



Département d'Automatique

Projet de fin d'études

Pour l'obtention du diplôme

D'Ingénieur d'Etat en Automatique

THÈME

**Commande et supervision de la station FESTO
PCS_COMPACT avec le TIA PORTAL**

Réalisé par :

CHERGUI Younes

Proposé et dirigé par :

D^r. R. ILLOUL

Juin 2012

Ecole Nationale Supérieure Polytechnique, 10, AV. Hassen Badi, El-Harrach, Algérie

REMERCIEMENTS

Je tiens tout d'abord à exprimer mes sincères remerciements à mon encadreur Dr. ILLOUL pour m'avoir encadré durant ce projet et m'avoir conseillés tout au long de mon travail.

Je remercie aussi les membres du jury pour l'honneur qu'ils m'ont fait en évaluant mon travail.

Je tiens à remercier particulièrement les enseignants de notre Ecole Nationale Supérieure Polytechnique pour tout ce qui nous ont transmis, leurs efforts et leur disponibilité.

Que tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail trouvent ici l'expression de notre sincère gratitude.

DEDICACES

*. Je dédie ce travail à ma mère, qui m'a soutenu et aidé tout au long de mon
parcourt et sans qui je n'en serais pas la .*

CHERGUI YOUNES

ملخص :

العمل المنجز في المذكرة يتمحور على إستعمال مسير صناعي ميرمج" سيمنس ". عندما اكتسبت دايرة الأوتوماتيك للمدرسة الوطنية العليا المتعددة التقنيات نموذج محطة تسيير وضبط الانظمة الصناعية مستوى الماء التصريف، الضغط، درجة الحرارة(، قمنا بإنجاز برنامج تسيير، تطابق وضبط مختلف الانظمة المحتواة في المحطة استنادا بالعمل تحديد وتحليل السيطرة،الذي قام به طلاب السابقون.

الكلمات المفتاحية:

مسير صناعي ميرمج" سيمنس" ، TIA PORTAL ، الاعددة و المحركات فسطو.

RESUME :

Le travail présenté dans ce mémoire est basé essentiellement sur l'utilisation des automates programmables SIEMENS. Le Département d'Automatique de l'Ecole Nationale Supérieure Polytechnique ayant acquis une station de contrôle des processus (débit, niveau, pression et température), FESTO, nous avons entrepris la réalisation d'un programme de gestion, et de supervision en se basant sur le travail d'identification et de synthèse de régulateur d'anciens étudiant.

Mots clés :

Automate programmable Siemens S7-313C, TIA PORTAL, capteurs et actionneurs FESTO.

ABSTRACT:

The work presented in this thesis is based mainly on the use of programmable SIEMENS. Since The Department of Automation of the national polytechnic school have acquired a station process control (flow, level, temperature and pressure), FESTO, we undertook a program of control and supervision, based on the work of identification and synthesis of control, that it was already made by former student

Key words:

Siemens PLC S7-313C, TIA PORTAL , sensors and actuators FESTO.

Introduction générale :

L'automate programmable industriel API (Programmable Logic Controller- PLC), est une machine électronique programmable destinée à piloter dans un milieu industriel et en temps réel, des procédés logiques séquentiels. Autrement dit, l'opérateur l'utilise pour le contrôle et essentiellement la commande d'un procédé industriel en assurant l'adaptation nécessaire entre les équipements de grande puissance du procédé industriel et ceux de faible puissance de la commande. Son objectif principal c'est "laisser-faire-seul" pour lequel le système contrôle ses sorties, décide et agit sur ses entrées afin de maintenir le fonctionnement tel que prévu par l'opérateur : C'est le principe de l'automatisme.

.L'automate programmable industriel A.P.I est aujourd'hui le constituant le plus répandu pour réaliser des automatismes. On le trouve pratiquement dans tous les secteurs de l'industrie car il répond à des besoins d'adaptation et de flexibilité pour un grand nombre d'opérations. Cette émergence est due en grande partie, à la puissance de son environnement de développement et aux larges possibilités d'interconnexions.

Dans le but de former ses élèves ingénieurs automaticiens à la maîtrise de cet outil, le Département Automatique de l'ENP, a acquis un système d'apprentissage didactique pour l'automatisation, FESTO, conçu pour répondre à un certain nombre d'exigences de formation professionnelle. En effet le matériel utilisé pour la fabrication de cette station de travail didactique est identique à celui disponible dans l'industrie.

Ce système didactique de contrôle des processus contient quatre sous-systèmes (niveau, débit, pression, température), qui grâce à l'automate programmable Siemens S7-313C, peuvent être utilisés réguler et contrôler.

Le but de notre travail est donc l'étude des différents composants (capteurs, actionneurs et convertisseurs...) de cette station afin de proposer une solution, à base d'automates programmable Siemens, permettant la gestion de la station d'un point de vue, régulation de niveau, débit, pression, et température. La communication entre la station et le pupitre de visualisation se fait via un réseau MPI. Il va sans dire que ce travail a nécessité l'étude préalable du logiciel de programmation des API Siemens STEP7 v11 et le logiciel de conception des interfaces homme-machine, WinCC advanced RT présent tous les deux sous un seul environnement commun et intuitive le TIA PORTAL.

Chapitre 1

1.1 Introduction sur les Automates programmables :

les automates programmables sont maintenant l'organe de contrôle le plus utilisé dans l'industrie, un automate programmable est un ordinateur industriel pouvant être programmé pour effectuer des tâches de contrôle, ils ont permis de pallier à la lourdeur de la conception par la logique à relais, en effet la programmation est beaucoup plus simple de même pour l'installation, aussi leurs rapidité, fiabilité, et la possibilité de communiquer sur plusieurs protocoles ont fait leurs succès dans le domaine de l'automatique industrielle.

les automates programmables sont conçus avec des entrées multiples et des sorties multiples supportant des températures très élevées, bruits électriques, et résistants à la vibration et au choc.

initialement les API ont été utilisés pour remplacer la logique à relais, mais grâce à l'évolution permanente de leurs fonctionnalités, on les trouve maintenant dans plusieurs autres applications beaucoup plus compliquées, et parce que la structure des API est basée sur le même principe que l'architecture d'un ordinateur, ils sont pas seulement capables de réaliser des tâches de commutation à relais mais aussi d'autres applications comme la temporisation, comptage, calcul, et traitement des signaux analogiques.

1.1.1 Les éléments de base d'un API :

Un API type peut être divisée en plusieurs parties comme le montre la figure suivante :

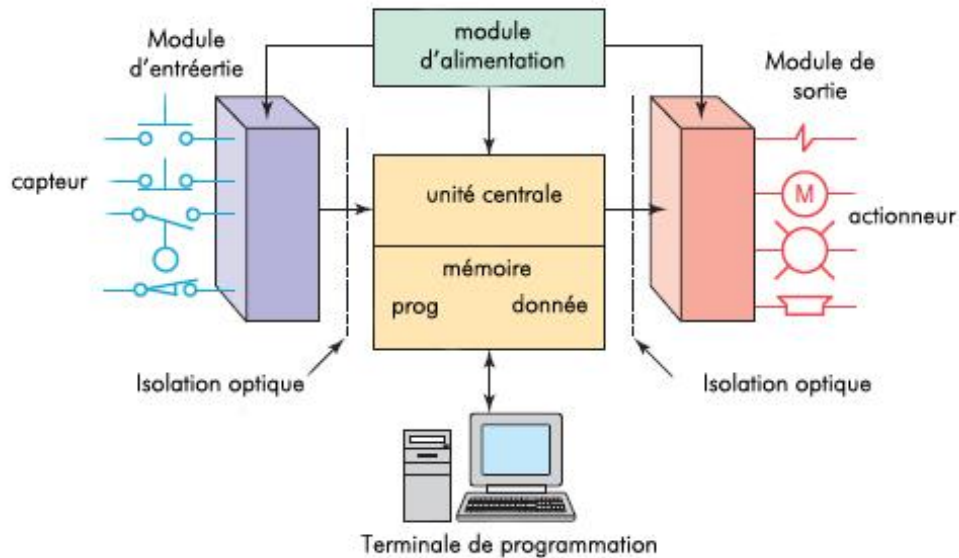


Figure 1.1: Architecture d'un API [2].

Unité central (CPU) peut être décrit comme le cerveau de l'api, elle contient un microprocesseur pour les traitement logique et le contrôle de la communication entre les modules, ce processeur requière une mémoire pour sauvegarder le résultat des opération logique, aussi cette mémoire (EPROM +RAM) est indispensable pour le programme ,l'unité central contrôle toutes les activité de l'api, le programme utilisateur est exécuté d'une manière cyclique ,chaque cycle comporte quatre étapes :

Chaque cycle commence par une lecture des entrées, puis le programme utilisateur est exécuté, une fois terminé, l'unité centrale fait une diagnostique interne et exécute les taches de communication, après le statut des sortie sont mise à jour, ce processus est répété continuellement tant que l'api est en mode Run.

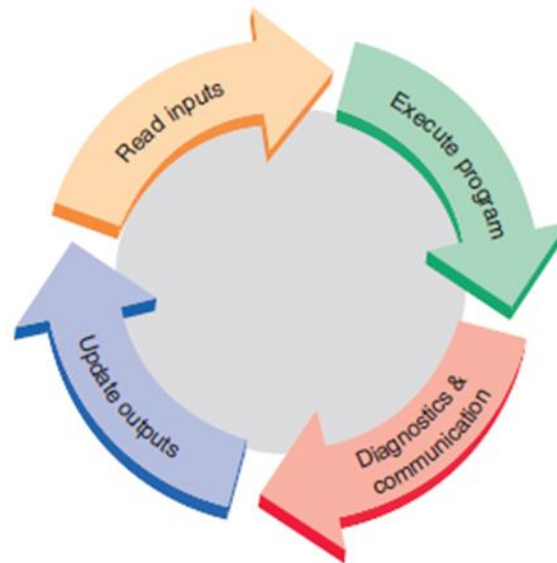


Figure 1.2: Cycle d'exécution d'un programme utilisateur [2].

Module d'alimentation Ce module permet l'alimentation en tension continue nécessaire au bon fonctionnement de l'automate programmable ainsi que le circuit de charge. Il convertit la tension du réseau (AC 220) en tension de service (DC 24V, 15V ou 5V). Ce module doit posséder de bonnes performances face aux microcoupures du réseau, ainsi qu'un transformateur d'isolement pour lutter contre les perturbations du même réseau.

Les Modules d'entrées/sorties (E/S) forment l'interface par laquelle les actionneur et les capteurs sont connecté à l'API, le rôle de cette interface est de conditionner les différent signaux qui entre ou sorte de l'api ,les capteurs comme les bouton poussoir ,fin de course et autre capteur sont câblé directement dans le module d'entrée ,les élément de sortie comme les solénoïde ,les indicateur sont directement câblés dans le module de sortie.
pour isoler électriquement les composent interne des terminaux d'entrée/sortie, les api généralement sont doté de Photocoupleur.

Modules d'E/S tout ou rien

(TOR) : Ces modules traitent une information qui ne peut prendre que deux états (faux, 0 ou 1), ils constituent l'interface entre l'API et les différents capteurs et pré-actionneurs présents.

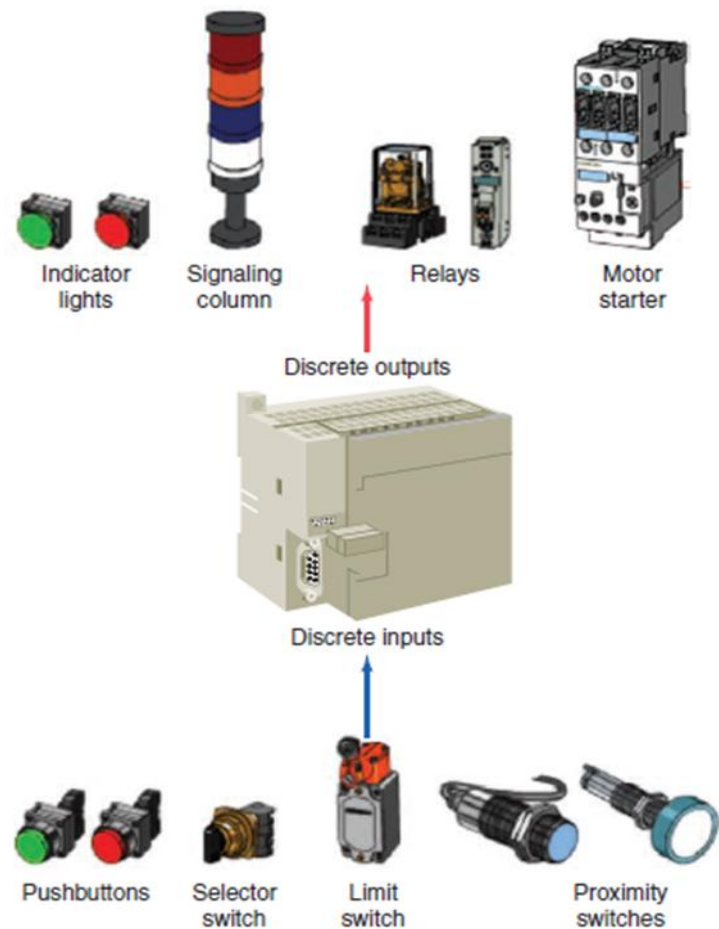


Figure 1.3: Modules d'E/S tout ou rien (TOR)[2].

Modules d'E/S analogique : Dans ce cas, le signal traité est analogique et prend des valeurs comprises dans une plage bien déterminée. Ces modules sont munis de convertisseur analogique/numérique pour les entrées et respectivement de convertisseur numérique/analogique.

Module spécialisés : l'information traitée est contenue dans des mots codés sous forme binaire ou bien hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent. Par exemple les modules de commande de mouvement et de position peuvent communiquer et piloter les servo-variateur comme le montre la figure suivante :

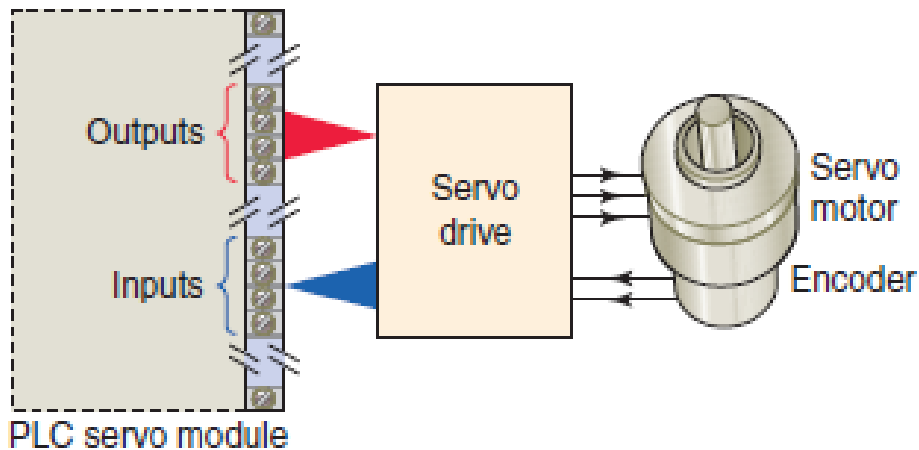


Figure 1.4: Module spécialisés [2].

Les modules de communication comprennent les consoles et les boîtiers tests.

Les consoles : Les consoles permettent la programmation, le paramétrage et les relevés d'informations, ils peuvent également afficher le résultat de l'autotest comprenant l'état des modules d'entrées et de sorties, l'état de la mémoire, de la batterie, etc. Ils sont équipés (pour la plupart) d'un écran à cristaux liquides.

Les boîtiers tests : Les boîtiers de tests quant à eux sont destinés aux personnels d'entretien ; ils permettent de visualiser le programme ou les valeurs des paramètres tel que l'affichage de la ligne de programme à contrôler, la visualisation de l'état des entrées et des sorties....

Auxiliaires :

Il s'agit principalement :

- D'un ventilateur : qui est en général indispensable dans les châssis comportant de nombreux modules ou dans le cas où la température ambiante est susceptible de devenir assez élevée (plus de 40 °C) ;
- Du support mécanique : Il peut s'agir d'un rack (structure métallique accueillant des cartes avec généralement un raccordement arrière), l'automate se présentant alors sous forme d'un ensemble de cartes, d'une armoire, d'une grille et des fixations correspondantes ;
- D'indicateurs d'état concernant la présence de tension, l'exécution du programme (mode RUN), la charge de la batterie, le bon fonctionnement des coupleurs...

1.2 Langages de programmation :**1.2.1 Introduction :**

La norme IEC 61131-3 (Commission Électrotechnique Internationale) définit cinq langages qui peuvent être utilisés pour la programmation des automates programmables industriels. Ces langages peuvent être divisés en deux catégories [3]:

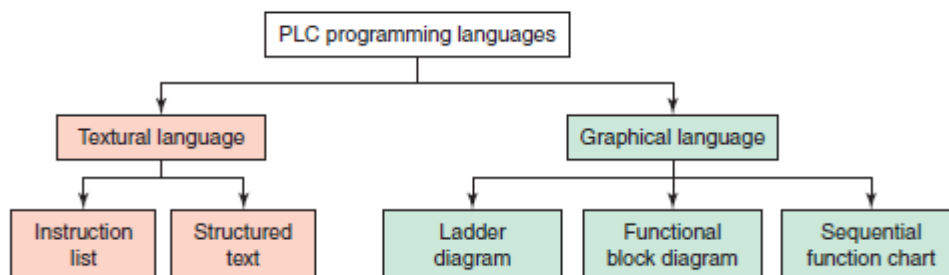


Figure 1.5: langages de programmation [2].

- **Langages graphiques :**
 - SFC « Sequential Funiculite Chart » ou GRAFCET ;
 - LD « Ladder Diagram » ou schéma à relais ;
 - FBD « Function Block Diagram » ou schéma par bloc.
- **Langages textuels :**
 - ST « structured text » ou texte structuré ;

- IL « Instruction List » ou liste d'instructions.

1.2.2 Les langages graphiques :

Le GRAFCET :

Le GRAFCET ou Graphe Fonctionnel de Commande Etape Transition est une méthode de représentation graphique permettant de décrire le cahier de charge d'un automatisme. Il est adapté aux systèmes à évolution séquentielle ; il est défini par un ensemble d'éléments graphiques de base traduisant le comportement de la partie commande vis-à-vis de ses entrées et ses sorties.

Un programme GRAFCET décrit un procédé comme une réceptivité .Celle-ci est une condition logique qui doit être vraie pour franchir la transition et passer à l'étape suivante. Des actions sont associées aux étapes du programme.

Le format graphique d'un programme GRAFCET est le suivant :

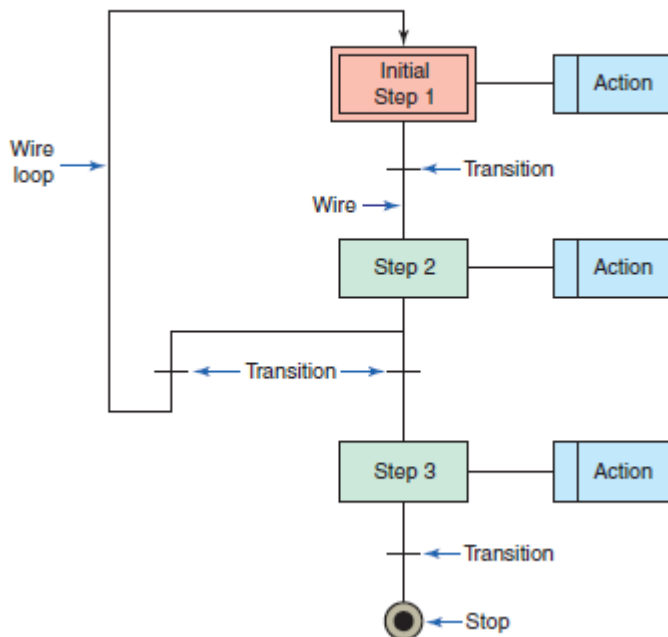


Figure 1.7 : Le GRAFCET.

Une étape représentée par un carré qui a un numéro identificateur et les actions associées sont indiquées dans un rectangle relié à la partie droite du carré (l'étape initiale est représentée par un carré double).

Une liaison orientée représentée par une ligne, parcourue par défaut de haut en bas ou de gauche à droite.

Une transition entre deux étapes et à laquelle est associée une réceptivité inscrite à sa droite, est représentée par une barre perpendiculaire aux liaisons orientées qui relient ces étapes.

Ladder Diagram :

Le LD est une représentation graphique qui traduit directement des équations booléennes en un circuit électrique et ce en combinant des contacts et des relais à l'aide de connexions horizontales et verticales ; les contacts représentent les entrées (contact normalement ouverts, contacts normalement fermés, ...) et les relais les sorties (relais directs, relais inversés,...). Les diagrammes LD sont limités sur la gauche par une barre d'alimentation et par la masse sur la droite.

Par exemple la fonction logique : $s = a \cdot (c + \bar{d} \cdot b)$, est réalisée par le diagramme suivant:

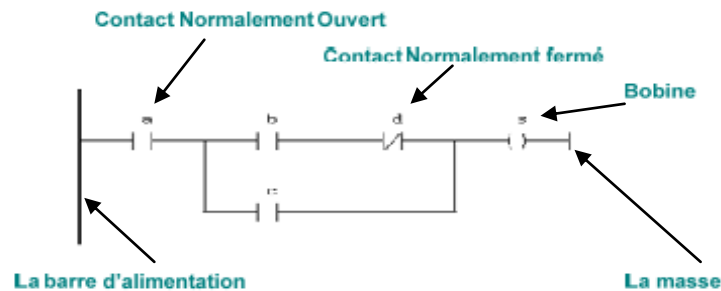


Figure 1.8 : Exemple 1 d'un programme en Ladder.

Le langage LD propose d'autres types de fonction tel que les fonctions de comptages et de temporisations, les fonctions arithmétiques et logiques, les fonctions de comparaison et de transfert. Par exemple pour réaliser la fonction : $c = a > b$, on utilise directement la fonction déjà disponible.

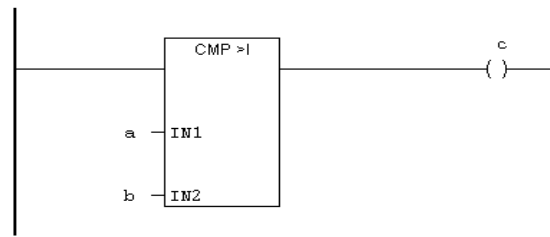


Figure 1.9 : Exemple 2 d'un programme en Ladder.

Bloc de fonction :

C'est un langage graphique qui permet la construction d'équations complexes à partir des opérateurs standards, ou de blocs fonctionnels ; il se compose de réseaux de fonctions préprogrammées ou non, représentées par des rectangles connectés entre eux par des lignes.

La programmation avec le FBD est très souple et facile à apprendre, la plupart des fonctions nécessaires (les fonctions arithmétique et logique, les fonctions de temporisation, des blocs fonctionnels PID...) sont déjà disponibles dans la bibliothèque. il suffit juste de les connecter et de bien paramétrer les entrées et les sorties, c'est-à-dire respecter le type des variables lors de la connexion.

Par exemple, pour réaliser la fonction arithmétique suivante : $w = 20 \cdot \frac{(x+y)}{z}$ on aura besoin de deux blocs : un pour l'addition, un pour la multiplication et un autre pour la division.

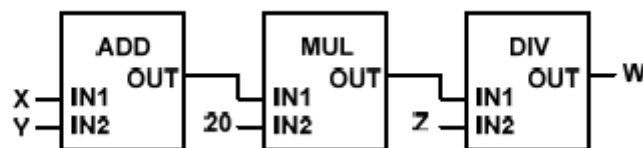


Figure 1.10 : Exemple d'un programme en Fonction Bloc.

1.2.3 Les langages textuels :

Texte Structuré :

Le langage ST (Structured Text) est un langage de programmation textuel de haut niveau dédié aux applications d'automatisation ; il est utilisé principalement pour décrire les procédures complexes et difficilement modélisables avec les langages graphiques. Il peut aussi être utilisé en tant que sous programme avec d'autres langages de programmation.

Il utilise les mêmes énoncés que les langages de programmation de haut niveau (Pascal, C, C++...) comme: les assignations, les appels de fonction, les énoncés de contrôle (IF, THEN, ELSE, CASE) ou d'itération (FOR, WHILE, REPEAT), en plus des opérations arithmétiques et logiques.

Par exemple pour le calcul de la distance entre deux points dans un plan à deux dimensions :

```
FUNCTION Ecart: REAL
VAR_INPIIT
X1, X2, Y1, Y2 : REAL;
END_VAR
BEGIN
RESULT := SQRT((X1-X2)^2 + (Y1-Y2)^2);
END_FUNCTION
```

Figure 1.11 : Exemple d'un programme en Langage Structuré.

Liste d'Instructions :

Le langage IL est un langage textuel de bas niveau (proche du langage machine), qui utilise un jeu d'instructions simples. Il trouve sa puissance dans les applications de petites tailles, et dans la création de sous programme ou procédure, car il permet un contrôle totale et une optimisation parfaite du code ; par contre pour les grandes applications il est très difficile de programmer avec le IL ; les programmes dans ce langage peuvent être traduit ou déduit des autres langages.

Le IL a la même structure que l'assembleur ; il utilise un ou plusieurs registres de travail. Les valeurs intermédiaires nécessaires pour l'exécution d'une instruction donnée seront mémorisées dans ces registres le temps de leur utilisation et il possède un jeu d'instructions assez riche pour décrire toutes les opérations arithmétiques et logiques, les opérations de comptage et temporisation, la comparaison et le transfert...

Pour l'exemple précédent : $w = 20 \cdot \frac{(x+y)}{z}$, on utilisera le code suivant :

```
LD X
LD Y
+R
LD +20
*R
LD Z
/R
ST W
```

Figure 1.12 : Exemple d'un programme en Liste d'Instructions.

1.3 Instrumentation utilisé dans la station FESTO

1.3.1 Relais électromagnétiques de control :

Le but principal d'un api était de remplacer les relais électromagnétiques par des commutation à semi-conducteurs qui pourrai être programmé, malgré ça les relais électromagnétiques sont toujours utiliser comme des diapositives auxiliaire pour commuter les diapositive E/S de terrain ,les API sont conçus pour remplacer les relai de petit taille qui était utilisé dans la logique a relai ,mais pas pour gérer les courant fort ou les haut voltage

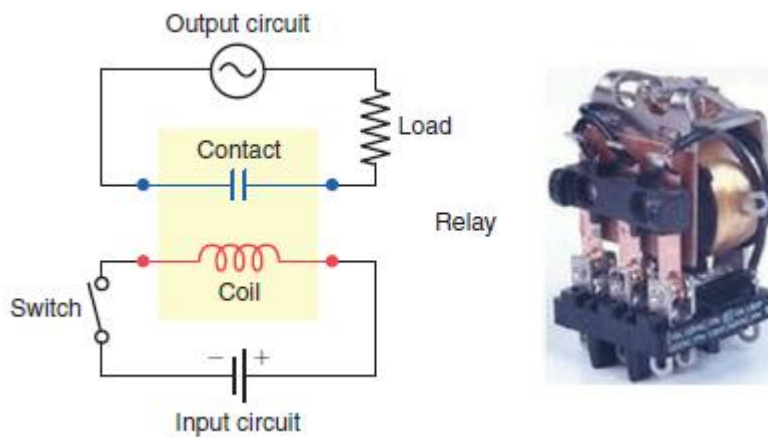


Figure 1.13: Relais électromagnétiques [2].

En plus, la compréhension du fonctionnement des relais électromagnétique et leurs terminologies est important pour convertir correctement les schémas a relai en programme Ladder .

Les symboles utilisés pour représenter les relais de control :

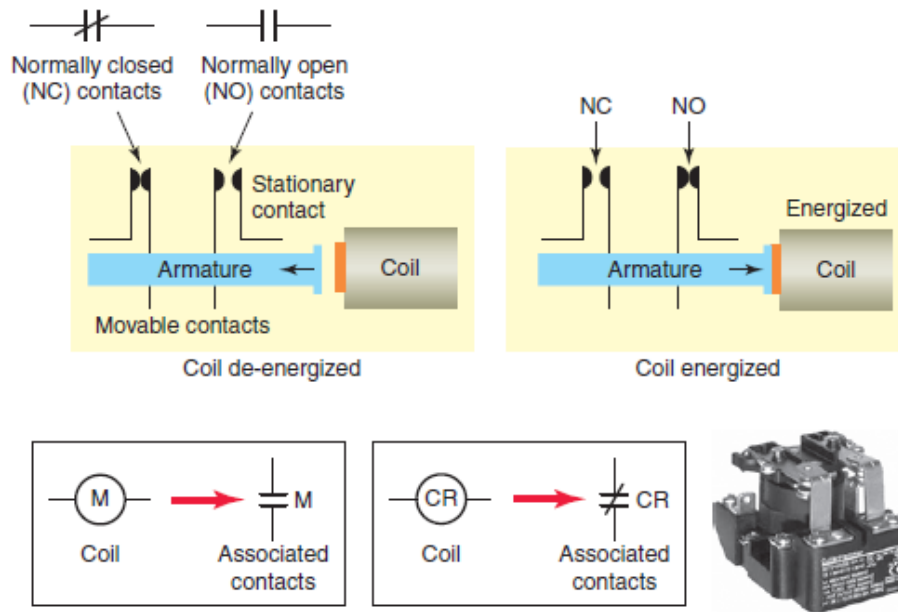
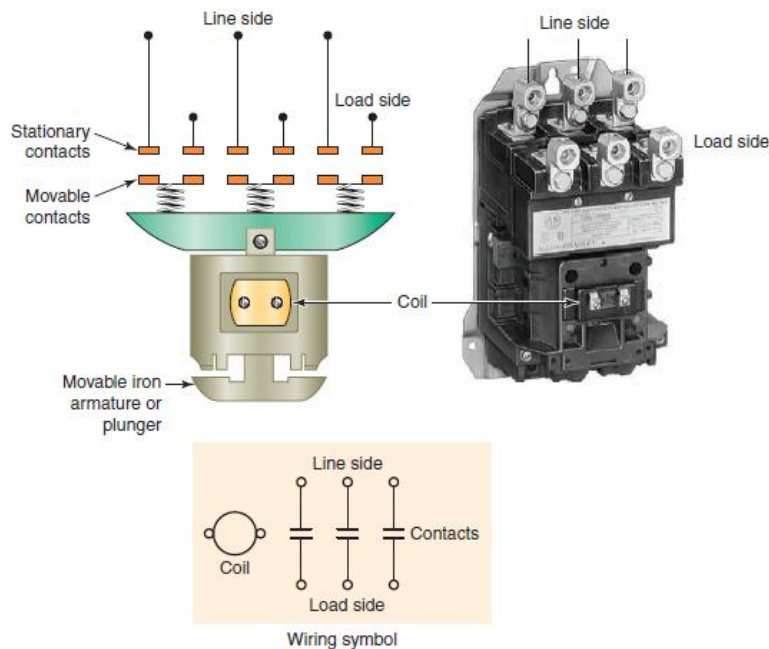


Figure 1.14: Symbole Relais électromagnétiques [2].

1.3.2 Contacteur :

Le contacteur est un relai spécial conçu pour commuter des grandes charges qui sont au-delà des capacités des relais de contrôle, la figure montre un contacteur triphasé



Les contacteurs sont conçu pour active et désactivé des circuits de courant fort sont être endommagé, comme les transformateur, les batteries de compensation et les moteur électrique.

1.3.3 Interrupteurs Actionnés manuellement :

Les interrupteurs manuelle sont contrôler manuellement Il s'agit notamment des boutons poussoir, des commutateurs, des interrupteurs à bascule

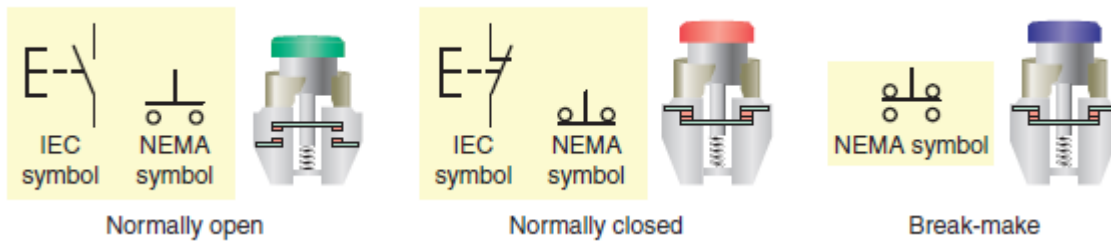


Figure 1.15: Interrupteurs Actionnés manuellement [2].

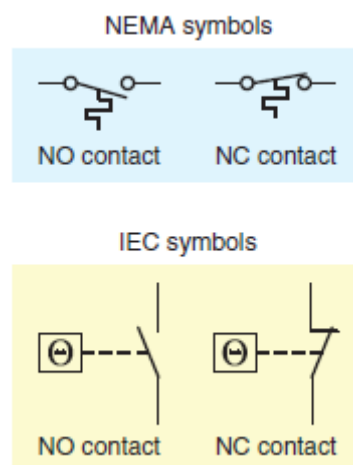
1.3.4 Interrupteurs Actionnés mécaniquement :

Ils sont contrôlés automatiquement par soi de la pression, position ou température, le fin de course est l'exemple le plus répandu



Figure 1.16: Interrupteurs Actionnés mécaniquement [2].

Autre interrupteur très commun aussi est le thermostat malgré qu'il y a plusieurs type ils sont tous activé par un changement de température et ils sont symbolisé comme le montre la figure ci-contre



1.3.5 Capteur :

Capteur de proximité :

Les capteurs de proximité comme le montre la figure sont des composants qui détectent la présence d'un objet sans contact physique



Figure 1.17: capteur inductif [2].

Les capteurs de proximité opèrent avec plusieurs principes, et cela dépend de la nature de la cible a détecté, quand l'objet a détecté est métallique on utilise les capteurs de type inductif

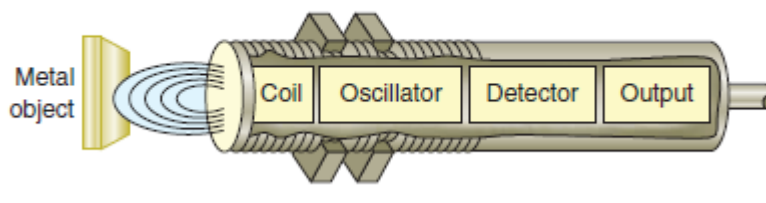


Figure 1.18: principe de fonctionnement capteur inductif [2].

Un autre type de capteur de proximité est le capteur de type capacitif qui peut détecter tout type d'objet métallique ou non.

Il existe deux types de câblage de ces capteurs :

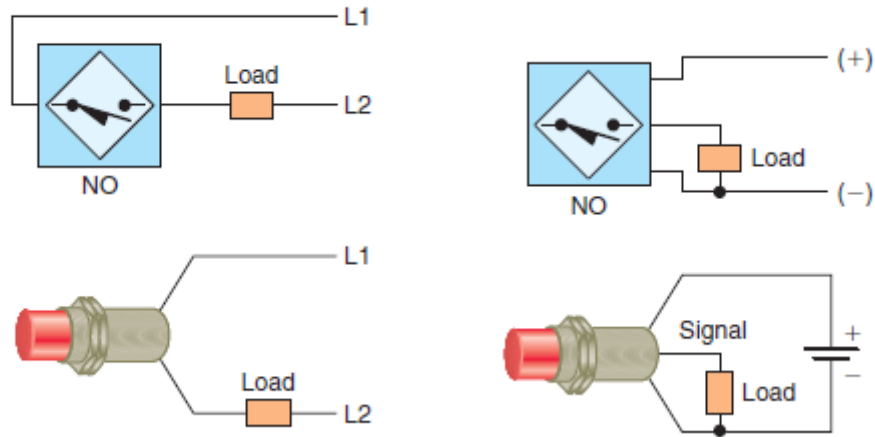
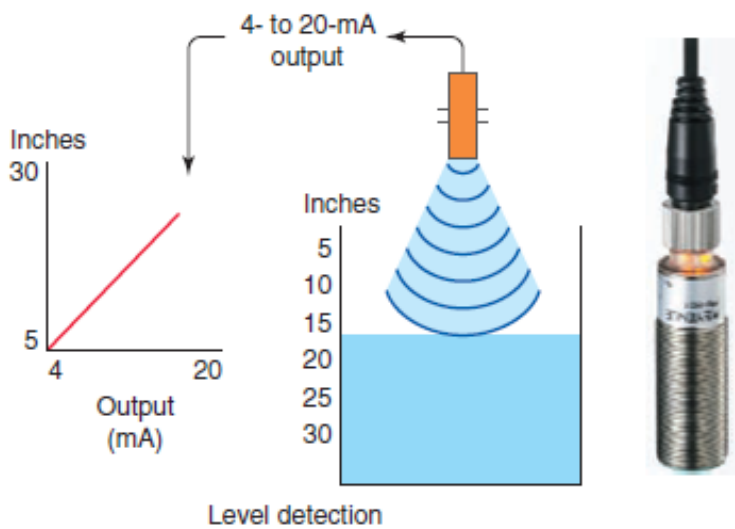


Figure 1.19: câblage capteur capacitifs [2].

Capteurs ultrason :

Ils opèrent par l'envoi d'un son haute fréquence en direction de la cible et mesure le temps que l'onde prend pour revenir, le temps pris par l'écho pour retourner au capteur est directement proportionnelle à la distance qui sépare le capteur de l'objet. La figure suivant illustre un exemple pratique de l'application de ce type de capteur pour la mesure de niveau, le retour de l'écho est converti électroniquement en courant de 4 à 20-mA.



Capteur de température :

Les thermocouples est le plus utilisé dans les capteurs de température,

Il se base sur le principe que si deux métaux de type différent sont joint, une f.e.m proportionnelle à la différence de température entre la jonction chaud et froid apparait.

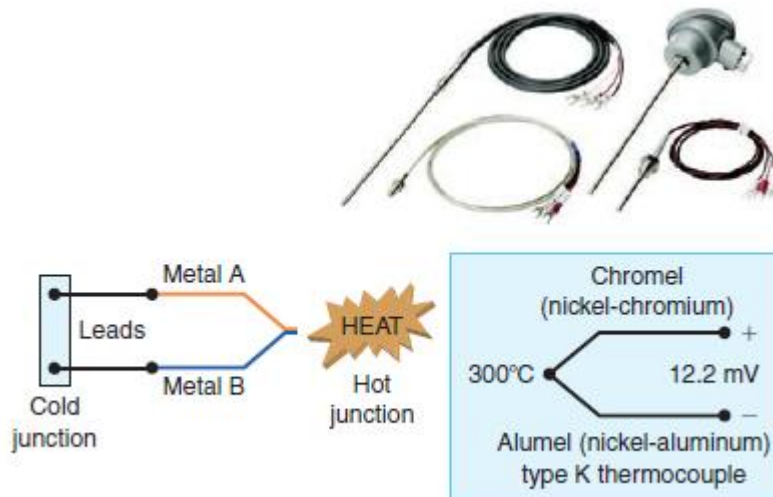


Figure 1.20: capteur de température [2].

1.4 Structure des systèmes de contrôle :

La plupart des systèmes sont contrôlé en boucle fermée, ils utilisent le feedback dans lequel la sortie d'un processus affecte l'entrée de signal de commande. Un système à circuit fermé mesure la production réelle du processus et la compare à la consigne. Des ajustements sont effectués en permanence par le système de contrôle jusqu'à ce que la différence entre la valeur souhaitée et sortie réelle tombe dans une tolérance prédéterminée. Figure 1.21 illustre un exemple d'une boucle fermée.

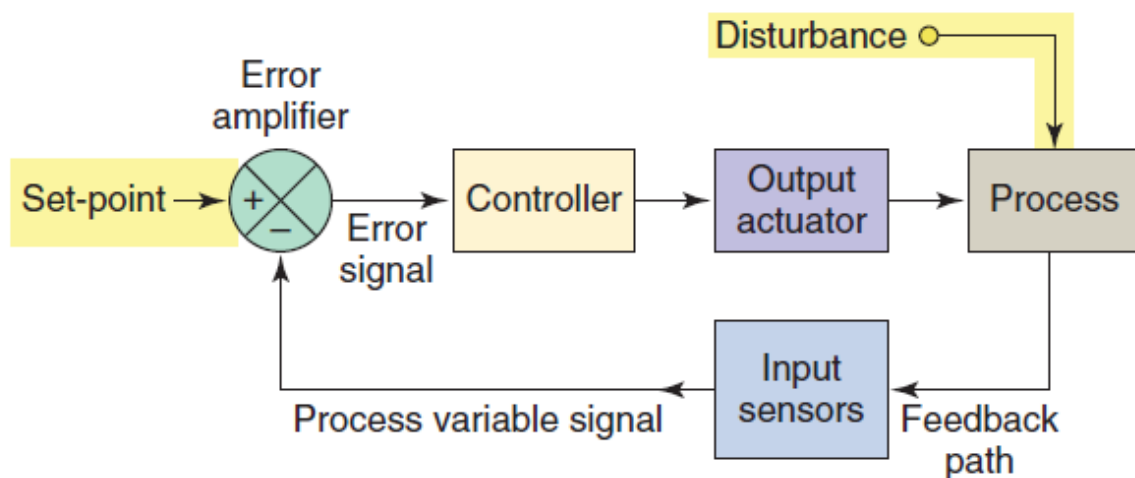


Figure 1.21: boucle de régulation en boucle fermée [2].

Consigne : L'entrée qui détermine le point de fonctionnement souhaité pour le processus.

Mesure : le signal du feedback qui contient des informations sur l'état du processus en cours.

Compensateur : calcule le signal d'erreur.

Régulateur : Produit la sortie corrective appropriée sur la base de l'entrée de signal d'erreur.

Actionneur : composant qui affecte directement un changement de processus. Les exemples sont les moteurs, les appareils de chauffage, les ventilateurs et les solénoïdes.

1.4.1 REGULATEUR TOUT OU RIEN :

Avec les régulateur tout ou rien, l'actionneur est soit à l'arrêt ou en marche, si la mesure est inférieur à la consigne l'actionneur est mis en marche, dans le cas contraire il sera arrêté.

La régulation sera achevé par les cycles de marche arrêt de l'actionneur.

La figure suivante montre un système comportant une régulation tout ou rien, dans lequel un liquide est chauffé avec de la vapeur.

Le processus de régulation se résume en trois étapes :

- si la température descend au-dessous de la consigne, la soupape ouvre de vapeur
- Lorsque la température du liquide passe au-dessus du point de consigne, la soupape de vapeur se ferme et la vapeur d'eau est fermée.
- Le cycle marche / arrêt se poursuivra aussi longtemps que le système fonctionne.

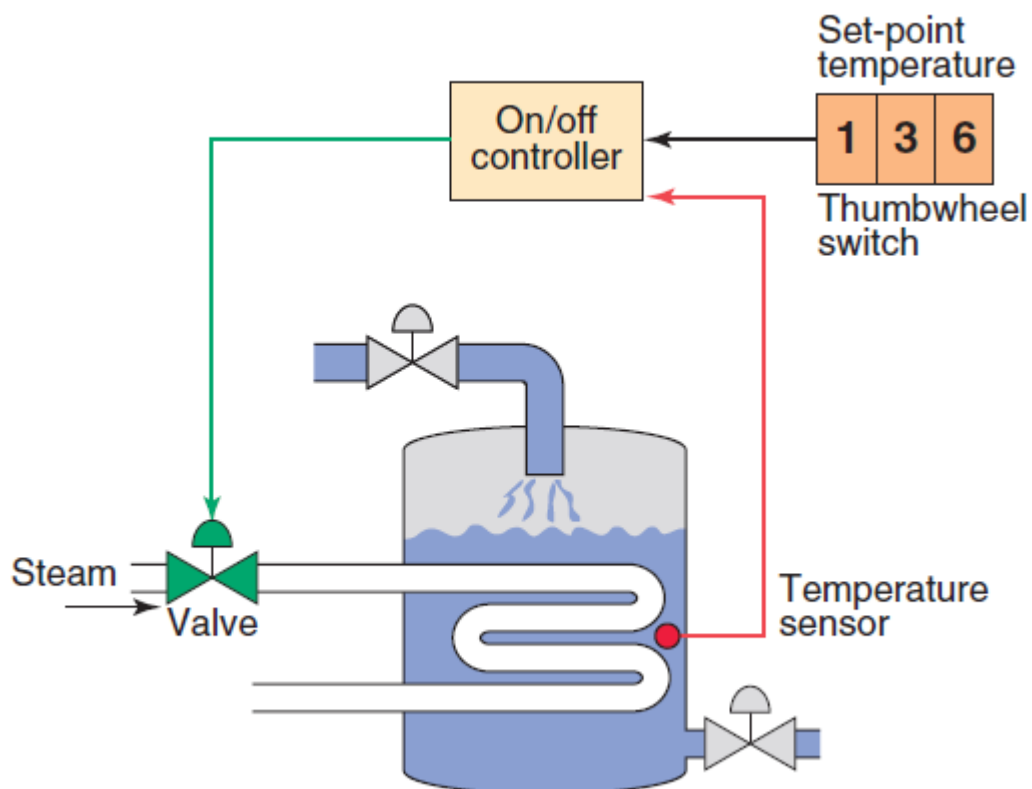


Figure 1.22: régulation Tout ou Rien [2].

Figure suivante illustre la réponse d'un régulateur de température tout ou rien :

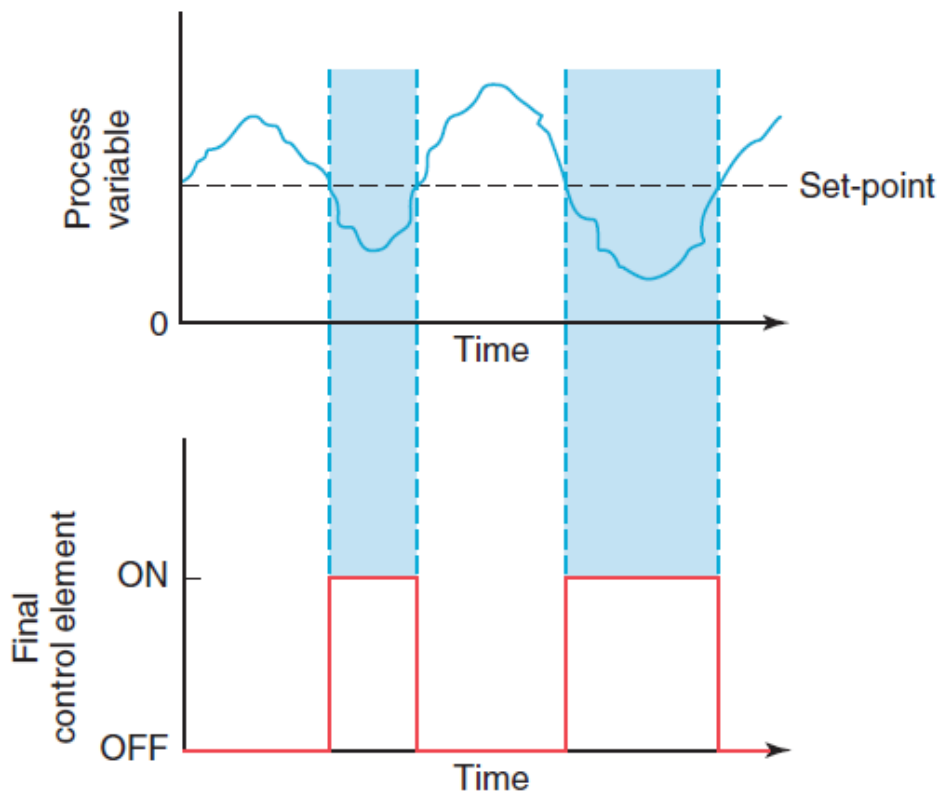


Figure 1.23: la réponse d'un régulateur de température tout ou rien [2].

- La sortie est activée lorsque la température descend au-dessous du point de la consigne et s'éteint lorsque la température atteint le point de consigne.
- le Contrôle est simple, mais le dépassement et le cycle répéter de la commutation peut être désavantageux dans certains procédés.
- La variable mesurée va osciller autour de la consigne à une amplitude et une fréquence dépendent du temps de réaction du processus.

Oscillations • Il peut être réduit en amplitude en augmentant la sensibilité du régulateur. Cette augmentation provoquera l'activation et désactivation de l'actionneur plus souvent, ce qui pourra nuire à ce dernier.

1.4.2 Régulateur PID :

Ils permettent à l'élément de commande final de prendre des positions intermédiaires entre marche et arrêt. Permet une commande analogique de l'organe de réglage pour faire varier la quantité d'énergie envoyé au processus, en fonction de l'écart qui existe entre la mesure et la consigne.

Un régulateur proportionnel permet un contrôle plus strict, parce que sa sortie peut prendre n'importe quelle valeur entre pleinement ouvert et entièrement fermé. En fonction de l'amplitude du signal d'erreur

En théorie, un régulateur proportionnel devrait être suffisant pour contrôler un processus, Tout changement dans la sortie du système est corrigé par une modification appropriée de la sortie du régulateur. Malheureusement l'emploi d'un régulateur proportionnel conduit à une erreur statique en régime permanent. Comme illustre la Figure 1.24. Selon l'application, ce décalage peut être ou ne pas être acceptable. La Commande proportionnelle est souvent utilisé en conjonction avec une action intégral et / ou une action dérivé

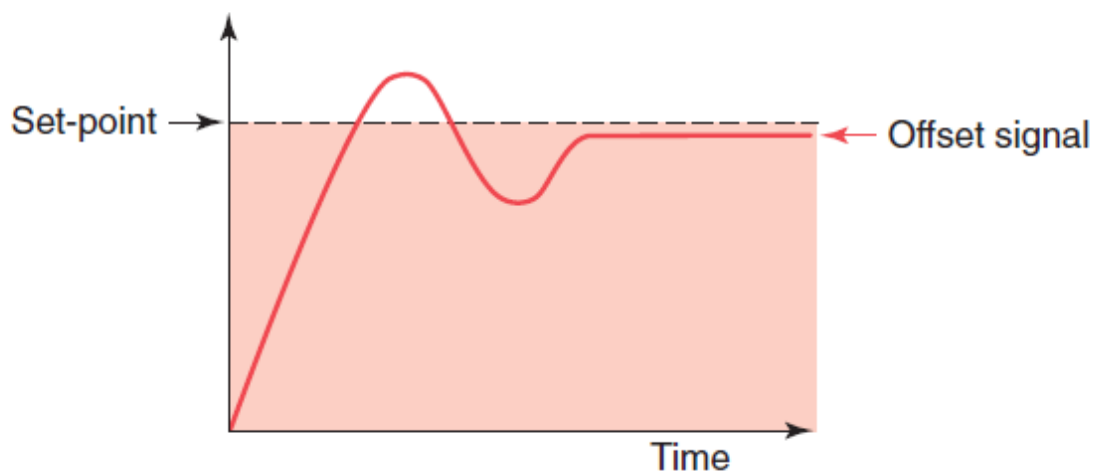


Figure 1.24: la réponse d'un régulateur proportionnel [2].

L'action intégrale, répond à la durée et la taille de l'erreur. Un signal d'erreur existe lorsqu'il y a une différence entre la variable de processus et la consigne, donc l'action intégrale provoque un changement à la sortie et continuer à le faire jusqu'à ce que l'erreur n'existe plus.

L'action dérivé répond à la vitesse à laquelle le signal d'erreur est changeant, plus le changement d'erreur est grand, plus la correction de sortie est grande. L'action dérivée est mesurée en unité de temps.

L'action dérivée agit sur l'erreur de la même manière que l'action intégrale, sauf que l'action dérivée agit sur le taux de changement plutôt que sur l'amplitude.

Chapitre 2

2.1 Introduction au portail TIA

Introduction :

Le portail Totally Integrated Automation, ci-après appelé portail TIA, offre la fonctionnalité complète pour réaliser des tâches d'automatisation, regroupée dans une plateforme logicielle globale.

Le portail TIA permet pour la première fois de disposer, au sein d'un cadre, d'un environnement de travail commun pour une ingénierie transparente avec différents systèmes SIMATIC. Le portail TIA nous permet donc également pour la première fois de travailler de manière sécurisée et confortable dans le système global.

Tous les logiciels requis, de la configuration matérielle à la visualisation du processus en passant par la programmation, sont intégrés dans un cadre complet d'ingénierie.



Figure 2.1: TIA PORTAL [1].

Les avantages de travailler avec le portail TIA

En travaillant avec le portail TIA, on bénéficie d'un soutien efficace lors de la réalisation de notre solution d'automatisation grâce aux fonctions suivantes :

- Ingénierie transparente basée sur un concept unitaire de commande
L'automatisation de processus et la visualisation de processus vont "de paire".
- Gestion centrale cohérente des données à l'aide d'éditeurs performants et d'une symbolique transparente

Une fois créées, les données sont disponibles dans tous les éditeurs. Les modifications et les corrections sont reprises et mises à jour automatiquement dans l'ensemble du projet.

- Concept global de bibliothèque
Utilisez les instructions prédéfinies et réutilisez des parties de projets déjà existantes.
- Plusieurs langages de programmation

Cinq langages de programmation différents sont à notre disposition pour effectuer votre tâche de programmation.

2.2 Vues du portail TIA

Deux vues différentes sont à notre disposition pour une initiation spécifique au portail TIA : La vue du portail et la vue du projet.

Les fonctions de la vue du portail et de la vue du projet sont expliquées dans ce qui suit.

La vue du portail :

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

Les différents portails ("Démarrage", "Appareils et réseaux", "Programmation API", "Visualisation", " En ligne et diagnostic", etc.) montrent de manière claire et ordonnée l'ensemble des étapes de travail nécessaires à l'exécution d'une tâche d'automatisation. Vous pouvez alors décider rapidement de ce que vous souhaitez faire et appeler l'outil dont vous avez besoin.

La figure suivante montre la structure de la vue de portail :

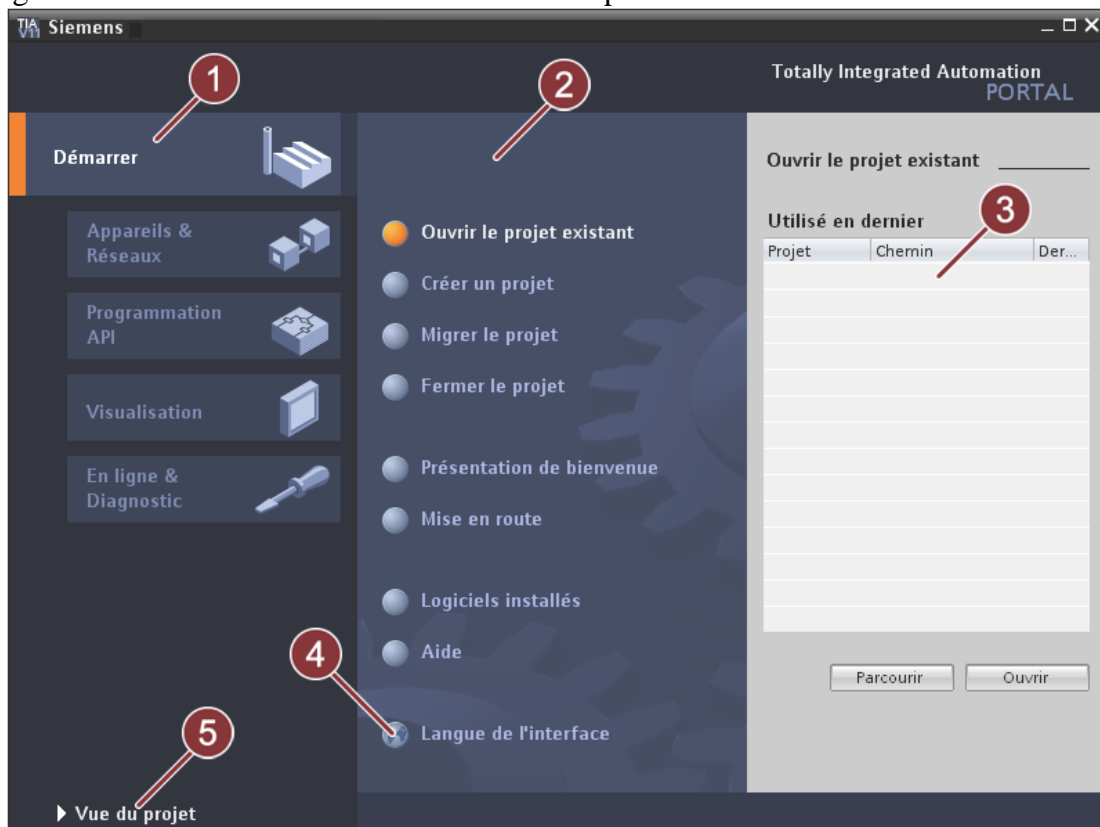


Figure 2.2: VUE TIA PORTAL .

①	Portails pour les différentes tâches : Les portails mettent à disposition les fonctions élémentaires requises par chaque type de tâche. Les portails qui vous sont proposés dans la vue de portail dépendent des produits installés.
②	Actions correspondant au portail sélectionné : En fonction du portail sélectionné, les actions que vous pouvez exécuter dans ce portail vous sont proposées ici. L'appel d'une aide contextuelle vous est proposé dans chaque portail.
③	Fenêtre de sélection correspondant à l'action sélectionnée : La fenêtre de sélection est disponible dans chaque portail. Son contenu s'adapte à la sélection en cours.
④	Sélectionner la langue d'interface.
⑤	Passer à la vue de projet.

La vue du projet

La vue du projet correspond à une vue structurée hiérarchisée de l'ensemble des composants d'un projet. La vue du projet permet un accès rapide intuitif à tous les objets du projet, aux zones de travail correspondantes et aux éditeurs. Les éditeurs existants permettent de créer et d'éditer tous les objets nécessaires au projet. Toutes les données correspondantes relatives aux objets sélectionnés s'affichent dans les différentes fenêtres de travail.

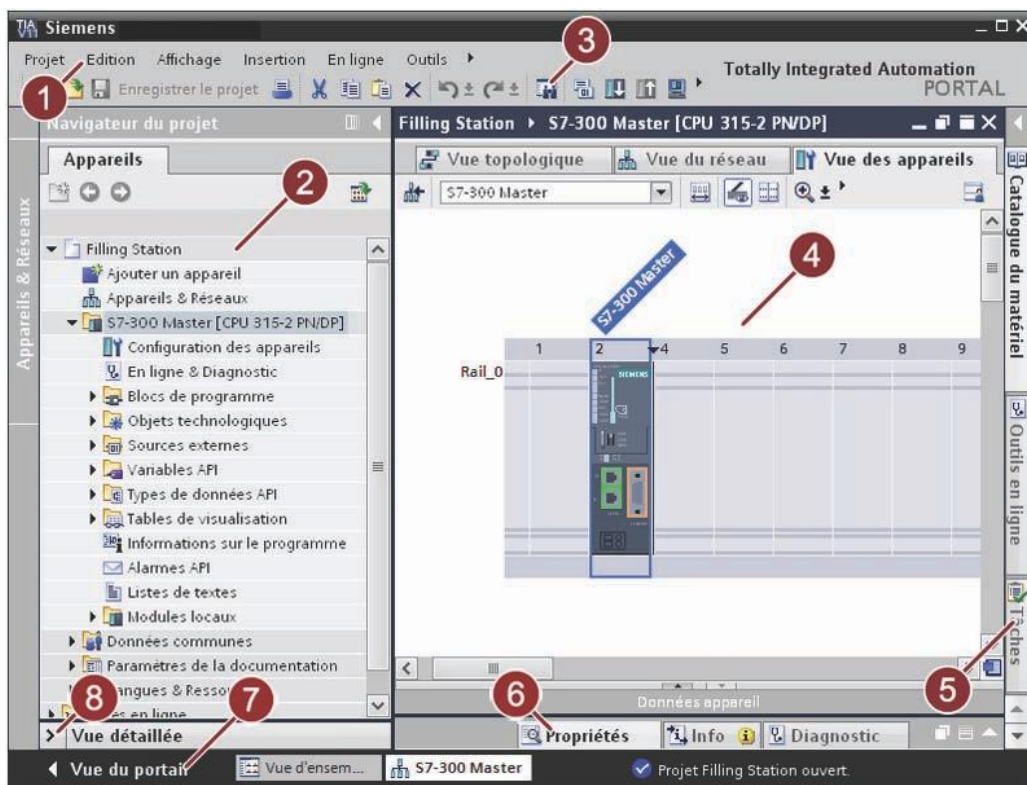


Figure 2.1: VUE PROJET.

①	Barre des menus : la barre des menus contient toutes les commandes indispensables pour réaliser votre tâche.
②	Navigateur de projet : le navigateur de projet vous permet d'accéder à tous les composants et données de projet.
③	Barre d'outils : la barre d'outils met à votre disposition des boutons vous permettant d'exécuter les commandes les plus fréquemment utilisées. Vous pouvez ainsi accéder à ces commandes plus vite que par les menus dans la barre des menus.
④	Zone de travail : la zone de travail affiche les objets que vous ouvrez afin de les éditer.
⑤	Task Cards : vous disposez de Task Cards en fonction de l'objet édité ou sélectionné. Les Task Cards disponibles figurent dans une barre au bord droit de l'écran. Vous pouvez à tout moment ouvrir ou fermer cette barre.
⑥	Fenêtre d'inspection : la fenêtre d'inspection affiche des informations supplémentaires sur un objet sélectionné ou sur des actions exécutées.
⑦	Vue du portail : basculer à la vue du portail
⑧	Vue de détail : la vue de détail affiche certains contenus d'un objet sélectionné. Les contenus possibles sont par ex. des listes de textes ou des variables.

2.3 Introduction au projet

Dans notre projet on a abordé une approche méthodique permettant de comprendre comment une tâche d'automatisation complète peut être exécutée pas à pas avec le portail TIA V11.0 Professional.

Le plus simplement du monde, en apprenant à utiliser le portail TIA avec cette approche méthodique, toutes les étapes de commande peuvent être transposées à d'autre projet d'automatisation.

Condition :

- Matériel :

Comme le module utilisé et le pupitre IHM permettant de tester le projet ,sont simulés côté logiciel, aucun matériel supplémentaire n'est nécessaire hormis un ordinateur en état de marche.

- Logiciels :

Les logiciels suivants doivent être installés sur l'ordinateur et en état de fonctionner :

- "STEP 7 Professional V11"
- "WinCC Advanced V11"
- les logiciels de simulation "S7-PLCSIM" et "WinCC Runtime Advanced Simulator"

Le projet :

Le projet "Station Festo" est réalisé en tant qu'installation industrielle de régulations de processus composé de quatre boucles de régulations, comme le montre la figure ci-après :



Figure 2.3: STATION FESTO [3].

Plus d'information sur la station de régulation festo PCS Compact et les boucles de régulation qu'elle comporte vont être expliqués dans le chapitre suivant .

2.4 Description de l'application :

Nous avons vu, plus haut dans ce chapitre, les différents types de régulation disponibles dans le système d'apprentissage didactique pour l'automatisation FESTO. Notre application consiste donc à élaborer un programme d'automatisation qui permet de gérer d'une manière optimale ces boucles de régulations

L'application est divisée en deux parties :

- La partie STEP7 qui contient les programmes principaux de régulation
- La partie WINCC qui contient les programmes de simulation, d'interfaçage et d'affichage.

Description des programmes STEP7 :

En premier lieu, nous nous proposons de présenter, ci-dessous l'organigramme décrivant le choix des différentes boucles de régulation.

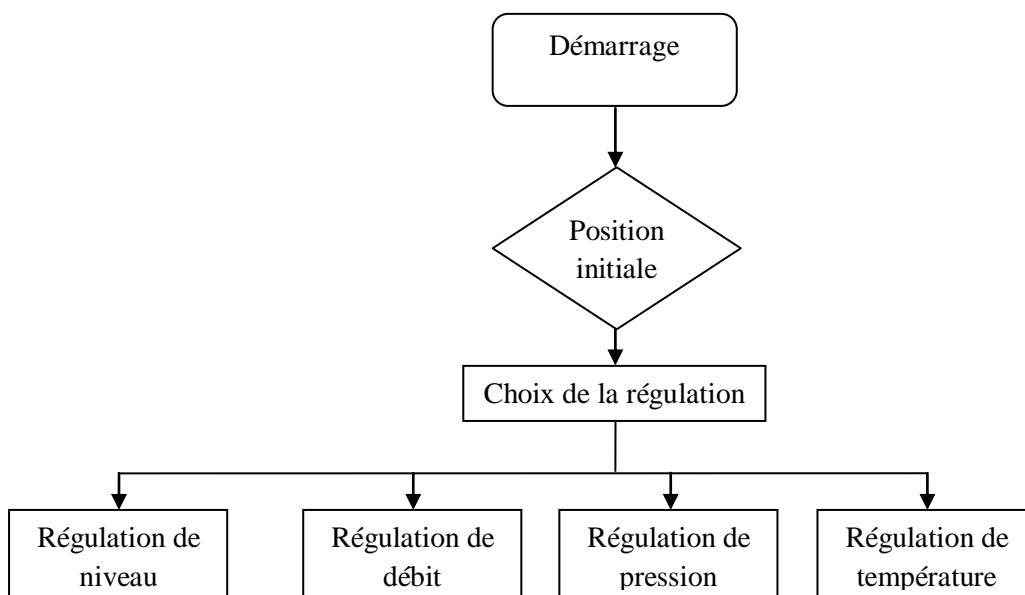


Figure 2.4 : Organigramme du choix des boucles de régulation.

Suivant le choix de la régulation plusieurs configurations s'offrent à nous. Si le choix s'est arrêté à la régulation de niveau, l'organigramme de calcul de la commande est alors représenté sur comme suit :

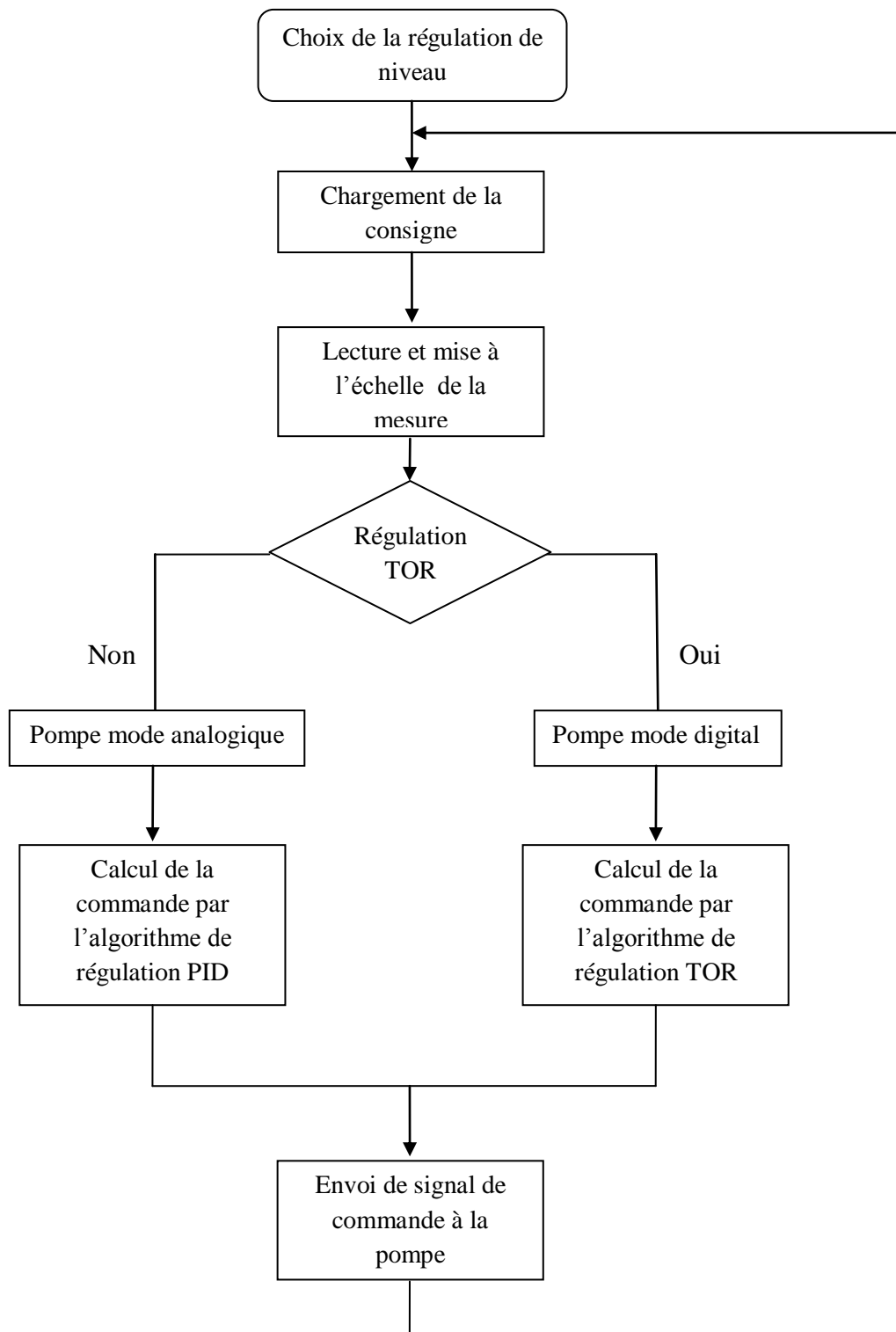


Figure 2.5 : Organigramme de la régulation de niveau.

Si maintenant le choix s'est arrêté à la régulation de débit ou bien sur la régulation de pression, l'organigramme de calcul de la commande est représenté sur comme suit :

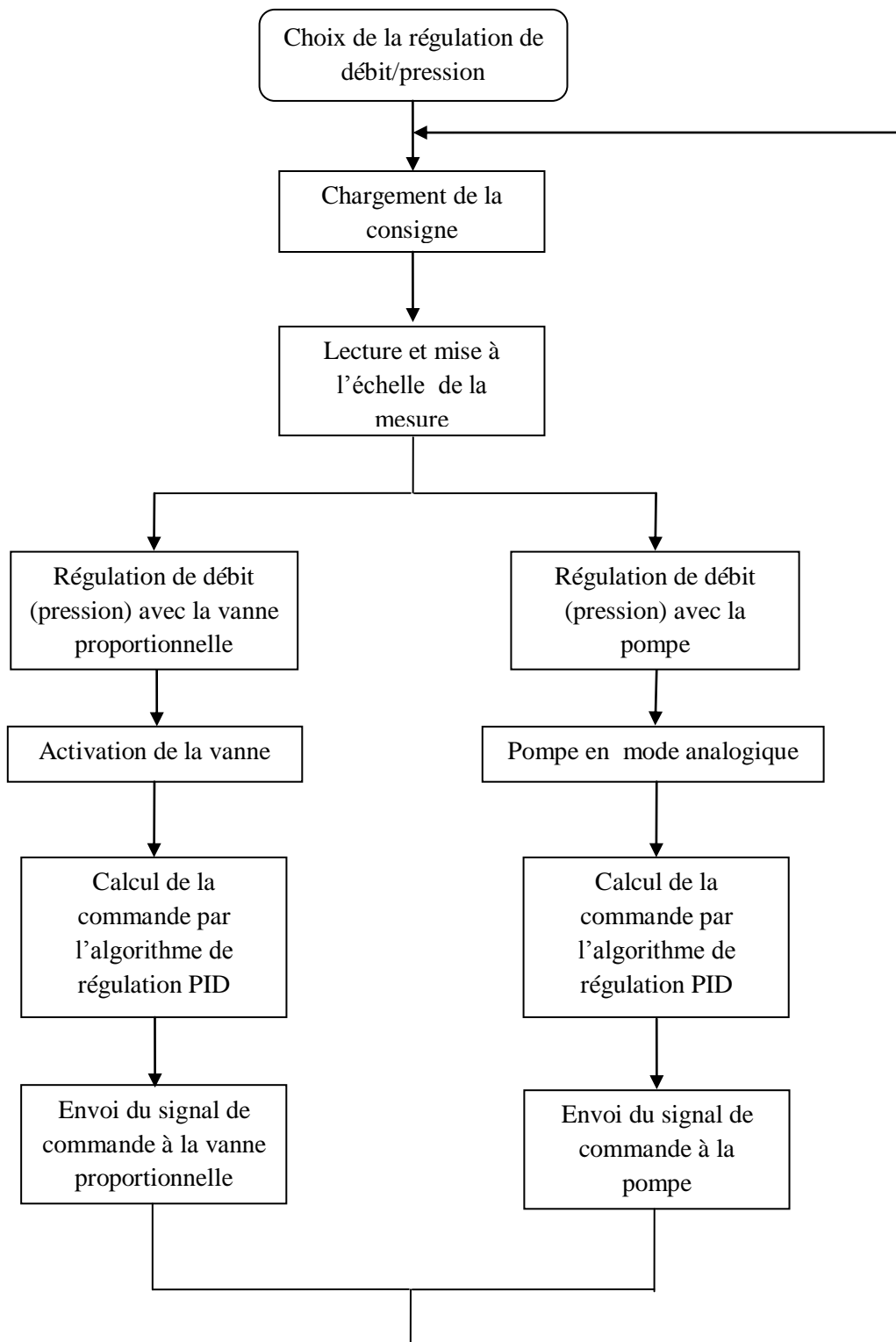


Figure 2.6 : Organigramme de la régulation de débit/pression.

Si le choix s'est à présent arrêté à la régulation de température, l'organigramme de calcul de la commande est représenté sur comme suit :

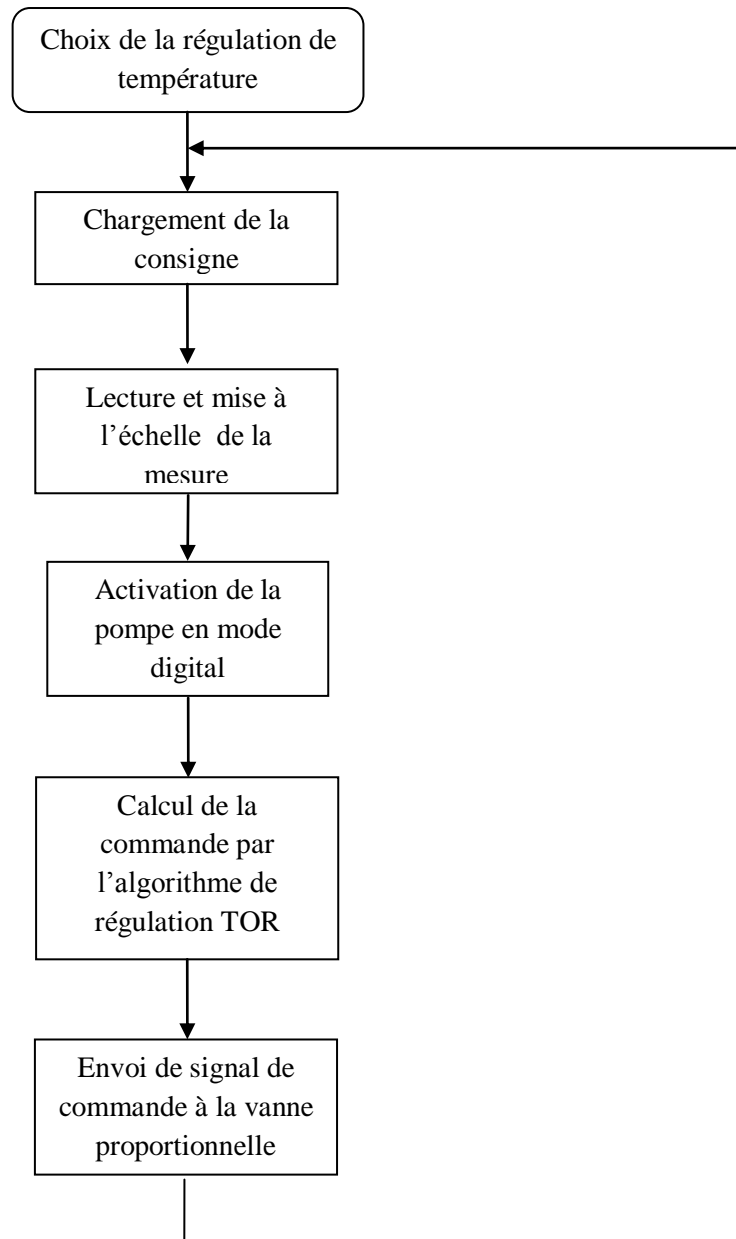


Figure 2.7 : Organigramme de la régulation de température.

2.5 Structure de la Tâche d'automatisation

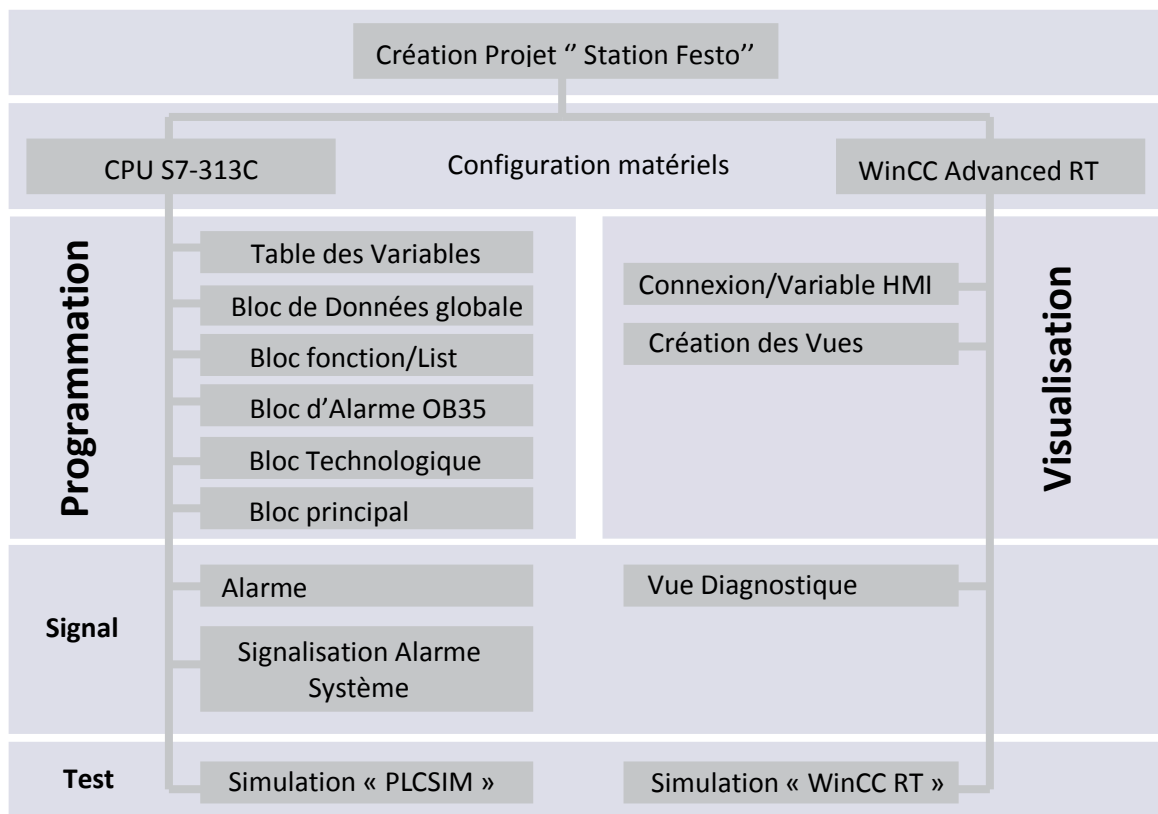
Dans ce qui suit, un aperçu des différentes étapes de la réalisation du projet et des objets que nous avons créé dans le cadre du projet "Festo Station" au sein du portail TIA.

Structure du projet "Station Festo"

Le projet en gros comprend les étapes de configuration suivantes :

- Création du projet "Station Festo"
- Insertion et configuration des matériels
- Programmer API
- Visualisation du processus
- Configuration des alarmes
- Tester en ligne l'exemple de projet

Le graphique ci-après montre ces étapes de configuration de projet et les objets à créer :



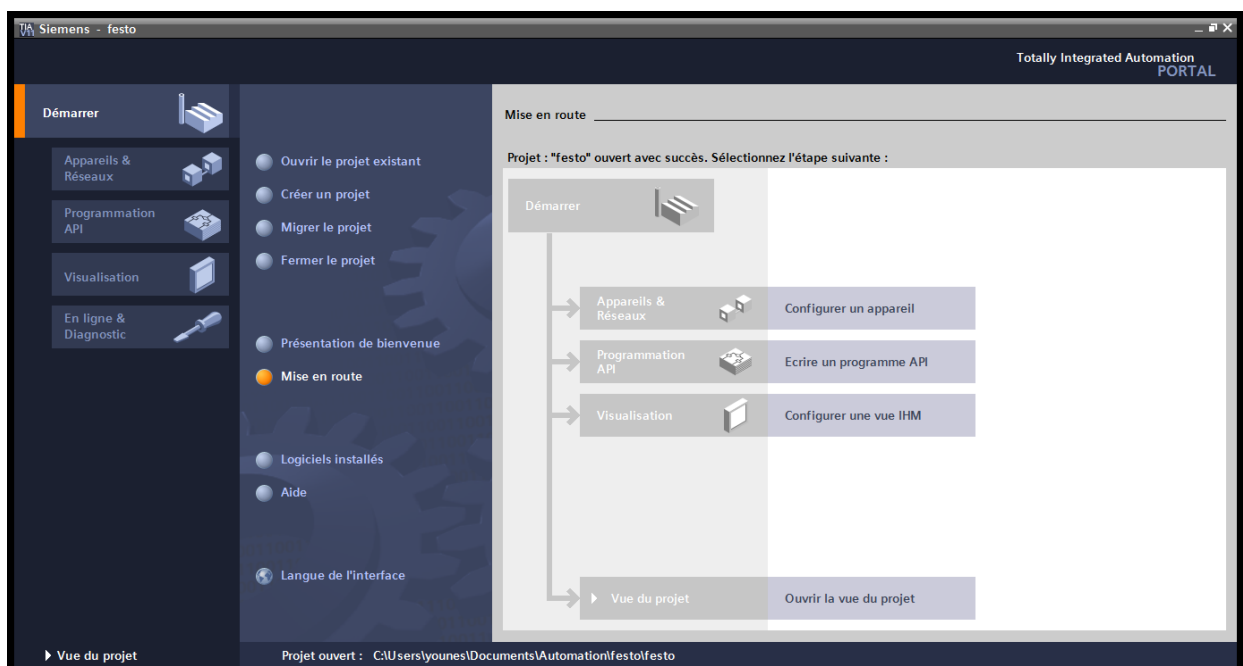
Etape	Contenu de la tâche	Réalisation
1	Création du projet "Festo station "	<ul style="list-style-type: none"> • Démarrer le portail TIA • Créer un nouveau projet
2	Insertion et configuration d'un matériel	<ul style="list-style-type: none"> • Insérer la CPU • Affichage de la CPU dans la vue des appareils • Configurer l'interface de la CPU • Insérer l'alimentation électrique et les modules de signaux • Adressages
3	Programmer API	<ul style="list-style-type: none"> • Créer des tables des variables API • Créer un bloc global de données • Créer les différents blocs fonctionnels • Appeler les blocs de programme dans le bloc principal [OB1]
4	Visualisation du processus	<ul style="list-style-type: none"> • Configurer le HMI Advanced Runtime • Créer la vue racine "choix de régulation " • Créer les vues "boucle de régulation et courbes"
5	Configuration des alarmes	<ul style="list-style-type: none"> • Alarmes • Signalisation d'erreurs système
6	Tester en ligne l'exemple de projet	<ul style="list-style-type: none"> • Test du programme • Tester la visualisation du processus

2.6 Créer le projet "Festo Station" :

2.6.1 Démarrer le portail TIA

Pour démarrer le portail TIA, on procède comme suit :

1. Cliquez sur Démarrer > Programmes > Siemens Automation > TIA-Portal V11.



Résultat :

Nous avons démarré le portail TIA et la vue de portail s'ouvre

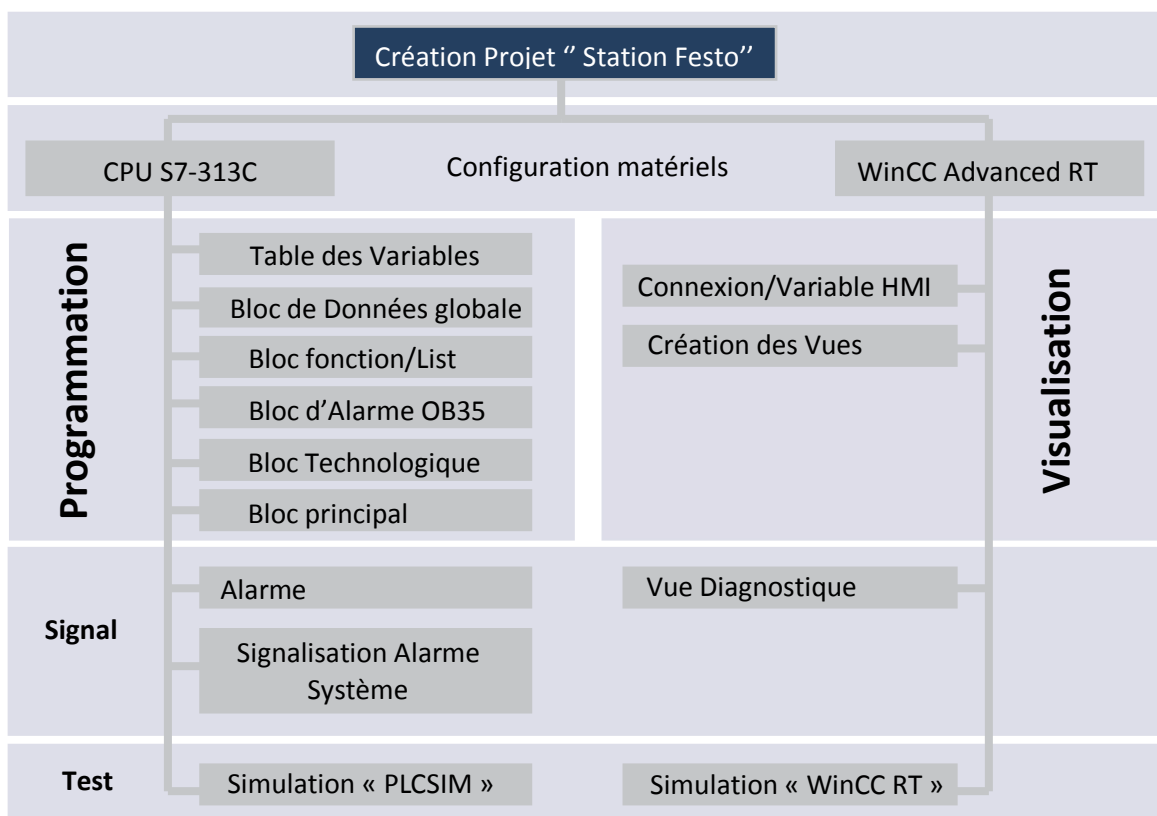
Figure 2.8: VUE PORTAL .

2.6.2 Créer un nouveau projet

Dans ce qui suit, nous créons un nouveau projet. Toutes les tâches d'automatisation à effectuer, comme p. ex. la configuration matérielle et la programmation API, sont exécutées au sein d'un projet.

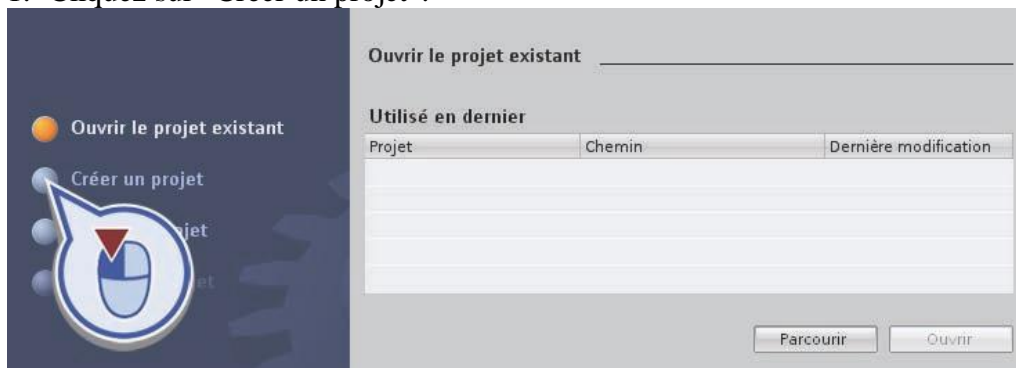
Avancement du projet

Le graphique suivant nous montre l'étape de configuration de projet à exécuter ensuite :

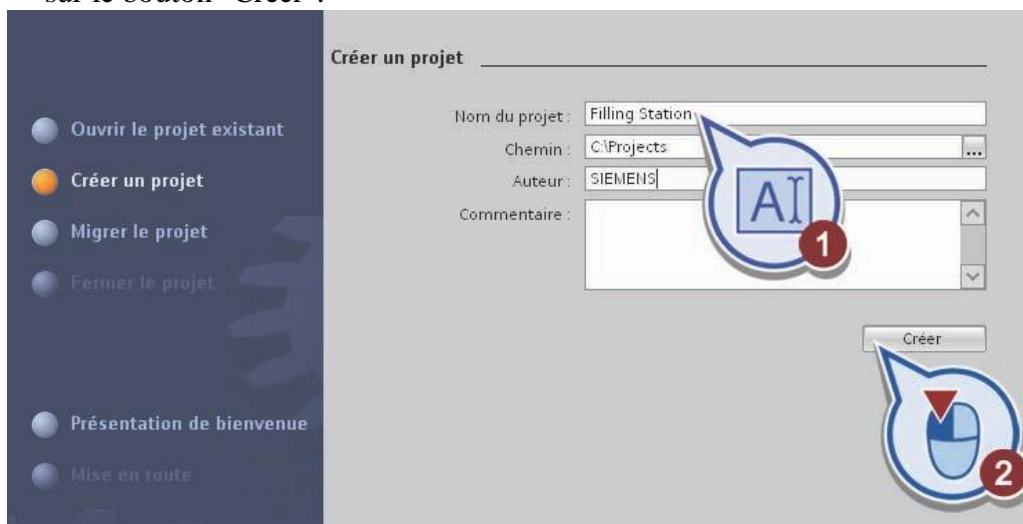


Pour créer l'exemple de projet "Festo station ", on procède comme suit :

1. Cliquez sur "Créer un projet".



2. Entrez dans le champ de texte "Nom du projet" la désignation "Festo Station" et cliquez sur le bouton "Créer".



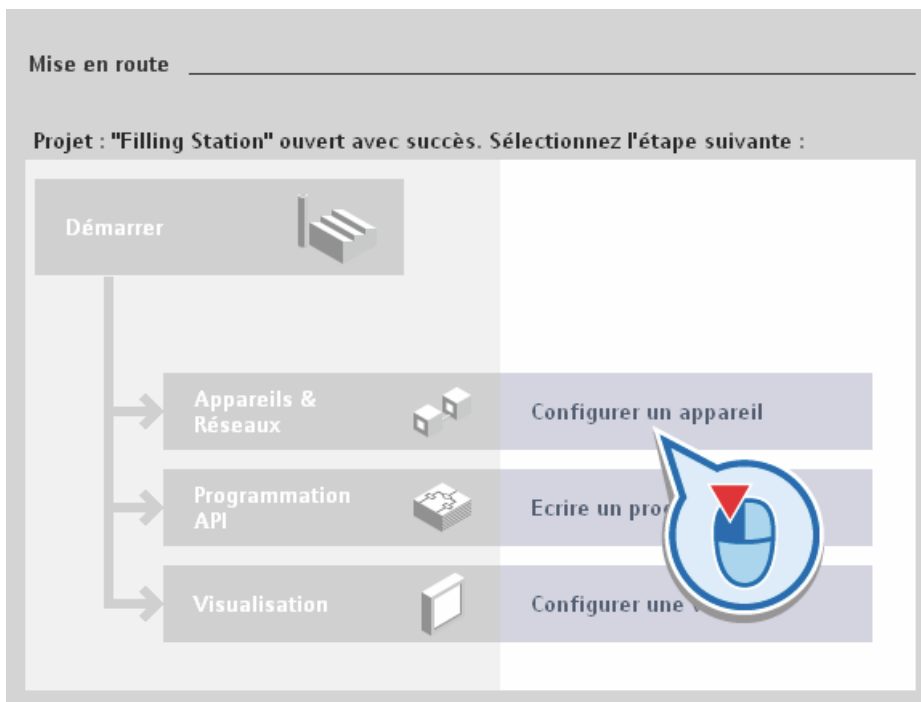
2.7 Insertion et configuration des matériels

2.7.1 Insertion d'une CPU

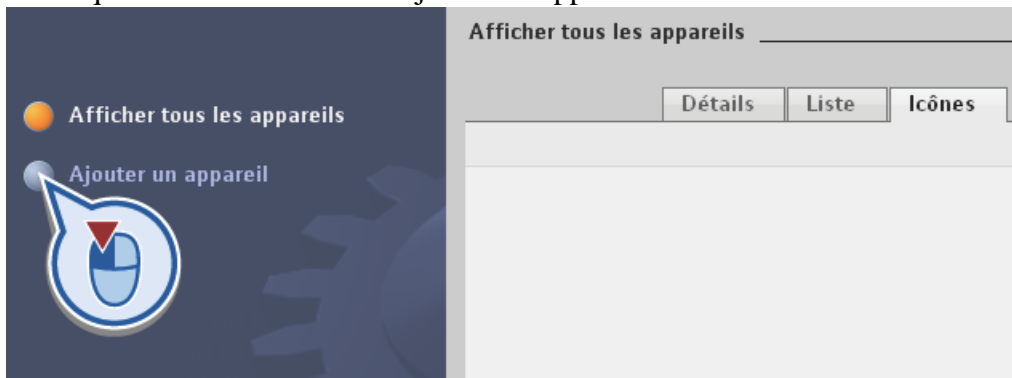
Nous insérons ci-après dans le projet "Festo Station" la CPU "S7-313C".

Pour insérer la CPU, on procède comme suit :

1. Cliquez sur "Configurer un appareil".

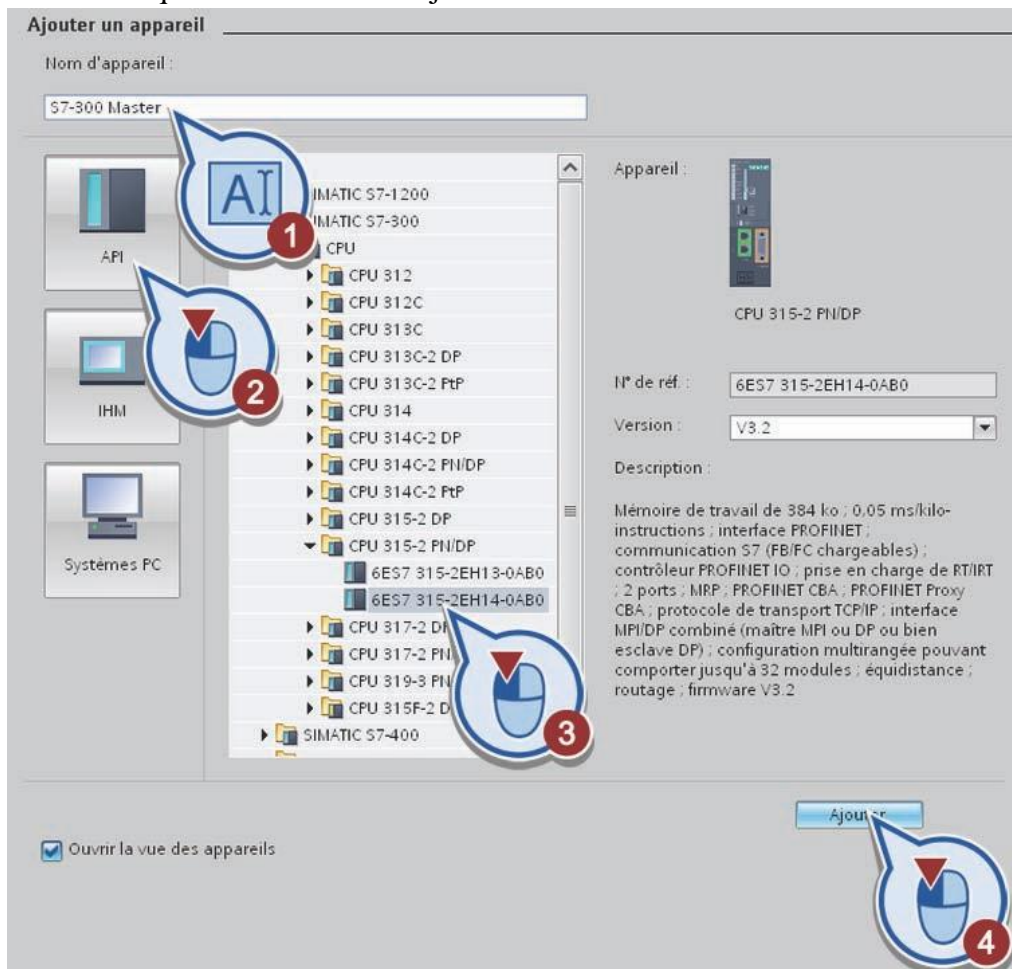


2. Cliquez sur la commande "Ajouter un appareil".



Pour créer la CPU :

- Dans la zone de texte "Nom d'appareil", entrez la désignation "S7-300".
- Sélectionnez la CPU "313C" : à cet effet, cliquez sur le bouton "API" et développez les dossiers "API" > "SIMATIC S7-300" > "CPU" > "CPU 313C" puis sélectionnez la version .
- Cliquez sur le bouton "Ajouter".



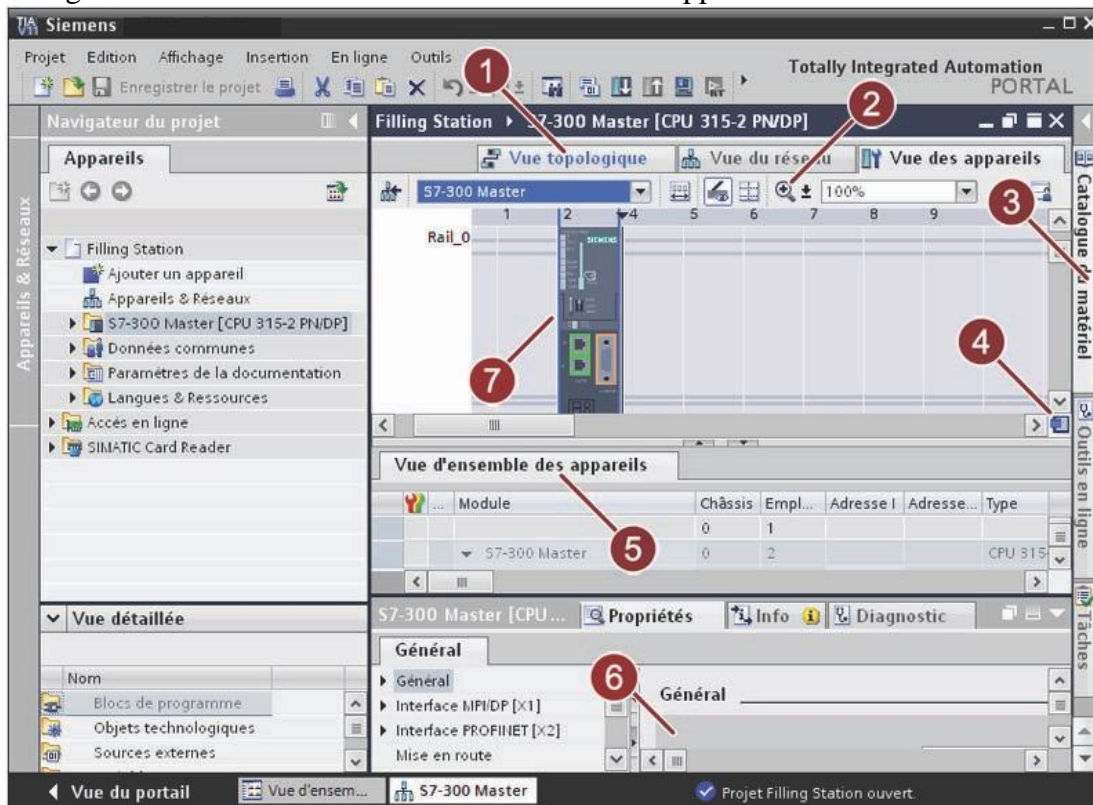
2.7.2 Affichage de la CPU dans la vue des appareils

La CPU que nous avons insérée dans le de projet "Festo Station", s'affiche dans la vue des appareils de l'éditeur de matériel et de réseaux.

L'éditeur de matériel et de réseaux dans le portail TIA :

La vue des appareils est l'une des trois zones de travail de l'éditeur de matériel et de réseaux dans lequel vous configurez et paramétrez les appareils et les modules.

La figure suivante montre la structure de la vue des appareils :

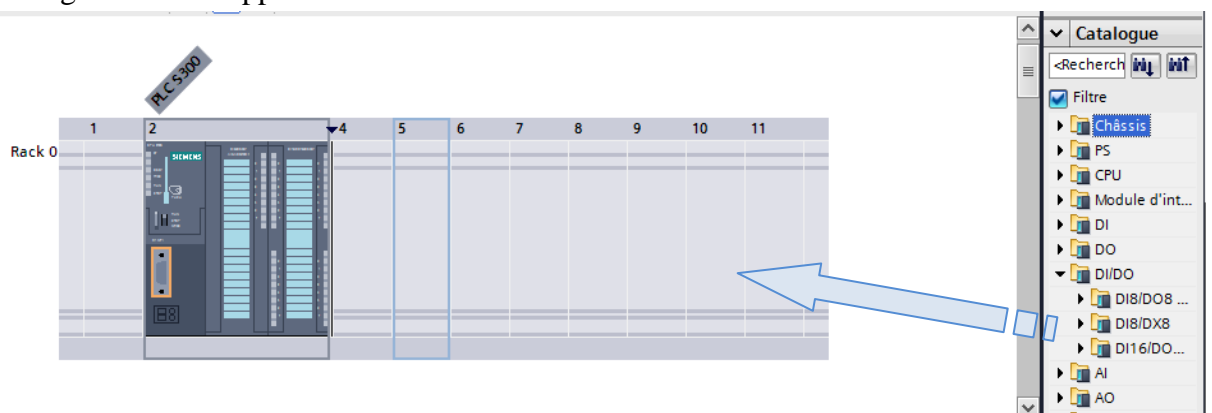


①	Onglets pour commuter entre la vue topologique, la vue du réseau et la vue des appareils
②	Barre d'outils de la vue des appareils : La barre d'outils permet de commuter entre les différents appareils et d'afficher ou de masquer certaines informations. La fonction zoom vous permet de modifier la représentation dans la zone graphique.
③	Task Card "Catalogue du matériel" : Le catalogue du matériel permet d'accéder rapidement aux différents composants matériels. A partir du catalogue du matériel, vous pouvez faire glisser les appareils et modules requis pour votre tâche d'automatisation dans la zone graphique de la vue des appareils.

④	Navigation générale : Cliquez avec la souris dans la navigation générale pour obtenir une vue générale des objets créés dans la zone graphique. En maintenant la touche de la souris enfoncée dans la navigation générale, vous pouvez naviguer rapidement vers les objets souhaités et les afficher dans la zone graphique.
⑤	Zone tabellaire de la vue des appareils : La zone tabellaire de la vue des appareils donne une vue d'ensemble du matériel utilisé avec les composants et données techniques les plus importants.

2.7.3 Insertion des modules de signaux

On insère ci-après le module d'entrée/de sortie numérique "DI24/DO16 x DC24V / 0,5A" et le module d'entrée/de sortie analogique AI5/AO2 x DC24/0.5A dans la configuration de l'appareille



Adressage

Les emplacements 2 et 3 de l'automate "Festo Station" sont occupés par un modules d'entrée/sortie numérique "DI24/DO16 x DC24V / 0,5A" et un module d'entrée/de sortie analogique AI5/AO2 x DC24/0.5A. Le premier modules disposent de 24 entrées et 16 sortie numériques et requièrent donc une plage d'adresses respective de 3 octets et 2 octets. Le second modules disposent de 5 entrées et 2 sortie Analogique et requièrent donc une plage d'adresses respective de 10 octets et 4 octets.

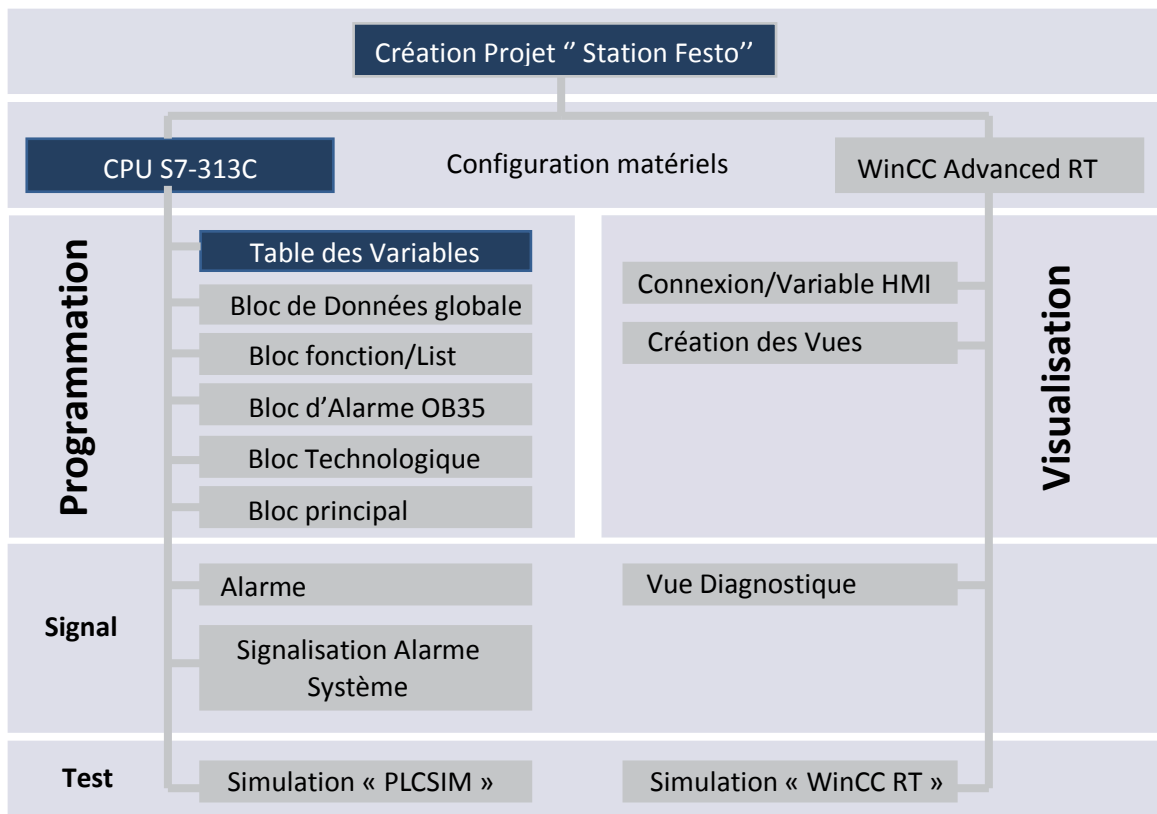
Module	Plage d'adresses Entrées	Plage d'adresses Sortie
DI24/DO16 sur l'emplacement 2	depuis l'octet 0 à l'octet 2 Plage d'adresses : I0.0 à I2.0	depuis l'octet 0 à l'octet 1 Plage d'adresses : Q0.0 à Q1.0
AI5/AO2 sur l'emplacement 3	depuis l'octet 3 à l'octet 13 Plage d'adresses : IW3 à IW12	depuis l'octet 2 à l'octet 6 Plage d'adresses : QW2 à QW5

2.8 Programmation de l'API

2.8.1 Création des tables de variables API

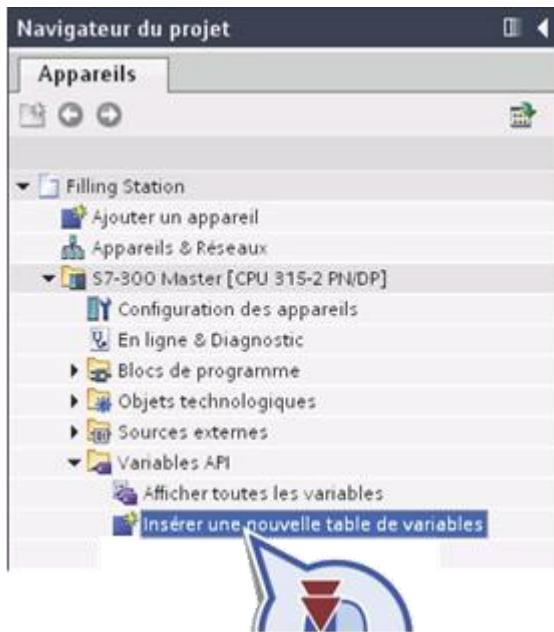
Avancement du projet

Le graphique suivant nous indique la prochaine étape de configuration que nous exécutons :



Pour créer une table de variables, on procède comme suit :

1. Dans la navigation du projet, ouvrez le dossier "Variables API" qui se trouve sous la CPU. "S7-300".
2. Effectuez un double-clic sur l'entrée "Insérer une nouvelle table de variables"



Résultat :

Nous avons créé correctement la table de variables API. Les variables qui seront définies au cours du projet sont créées dans cette table de variables API.

Remarque

Tables des variables API

Dans un projet, l'éditeur dans lequel vous modifiez les propriétés des variables n'a pas d'importance. Toutes les modifications sont appliquées automatiquement à tous les points d'utilisation correspondants.

2.8.2 Créer un bloc de données global

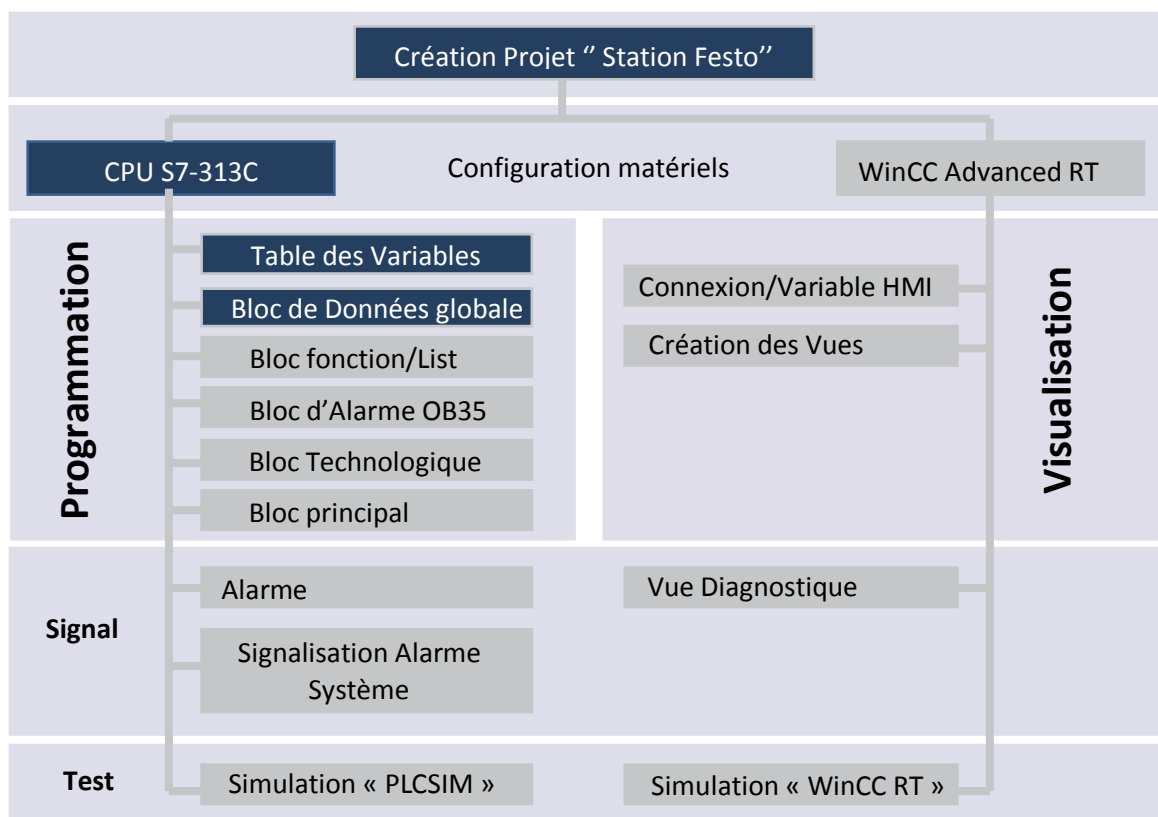
Ce bloc de données nous permet de gérer toutes les données de programme du projet "Festo Station" à un emplacement central.

Définition : Bloc de données

Les blocs de données servent à mémoriser les données de programme. Vous pouvez accéder en lecture ou en écriture à un bloc de données global à partir de tous les blocs, qu'il s'agisse d'un bloc fonctionnel, d'une fonction ou d'un bloc d'organisation. Les données de programme sont conservées dans le bloc de données global jusqu'à ce qu'elles soient écrasées.

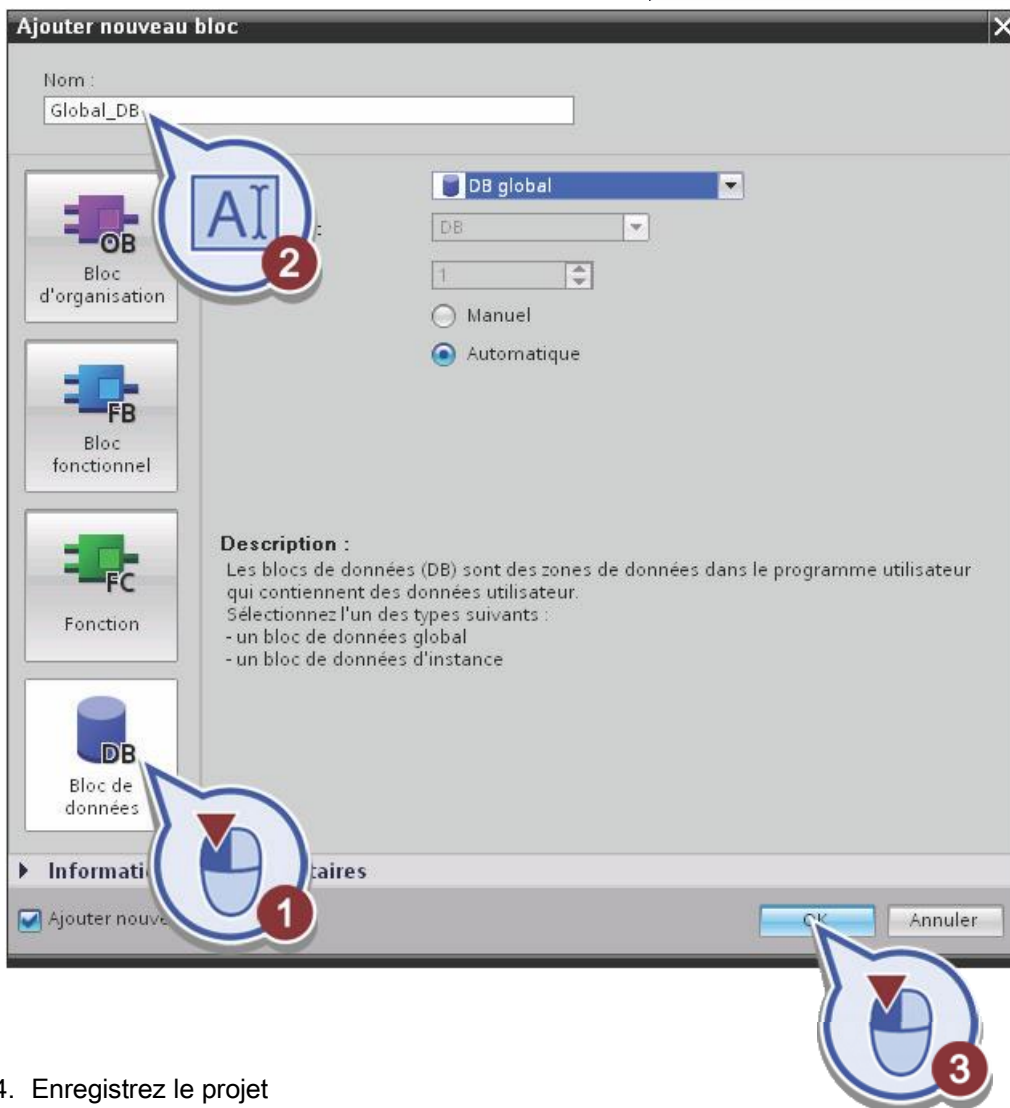
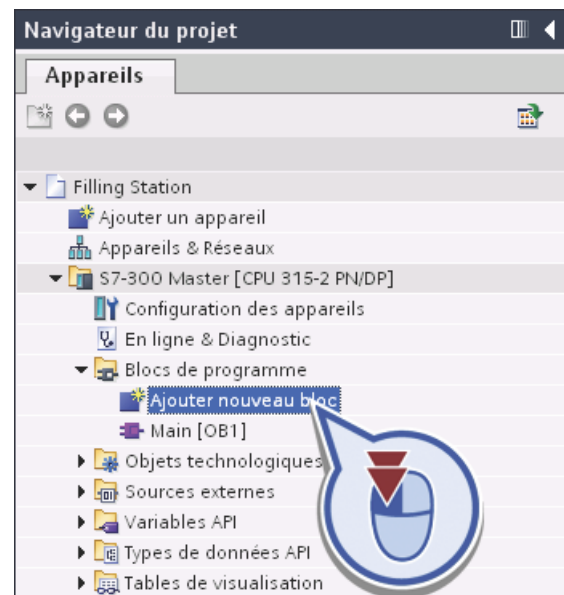
Avancement du projet

Le graphique suivant nous indique la prochaine étape de configuration que nous exécutons :



Pour créer un bloc de données global, on procède comme suit :

1. Ouvrez le dossier "Blocs de programme".
2. Double-cliquez sur "Ajouter nouveau bloc".
3. Pour ajouter un nouveau bloc de données :
 - Cliquez sur le bouton "Bloc de données".
 - Attribuez le nom de bloc "Global_DB".
 - Sélectionnez le type "Global_DB".
 - Cliquez sur "OK".



4. Enregistrez le projet

Résultat :

Nous avons créé correctement le bloc de données global "Global_DB" dans lequel nous gérons ultérieurement les données du projet.

2.8.3 Bloc fonction/List

Bloc mise à l'échelle (en langage LIST)

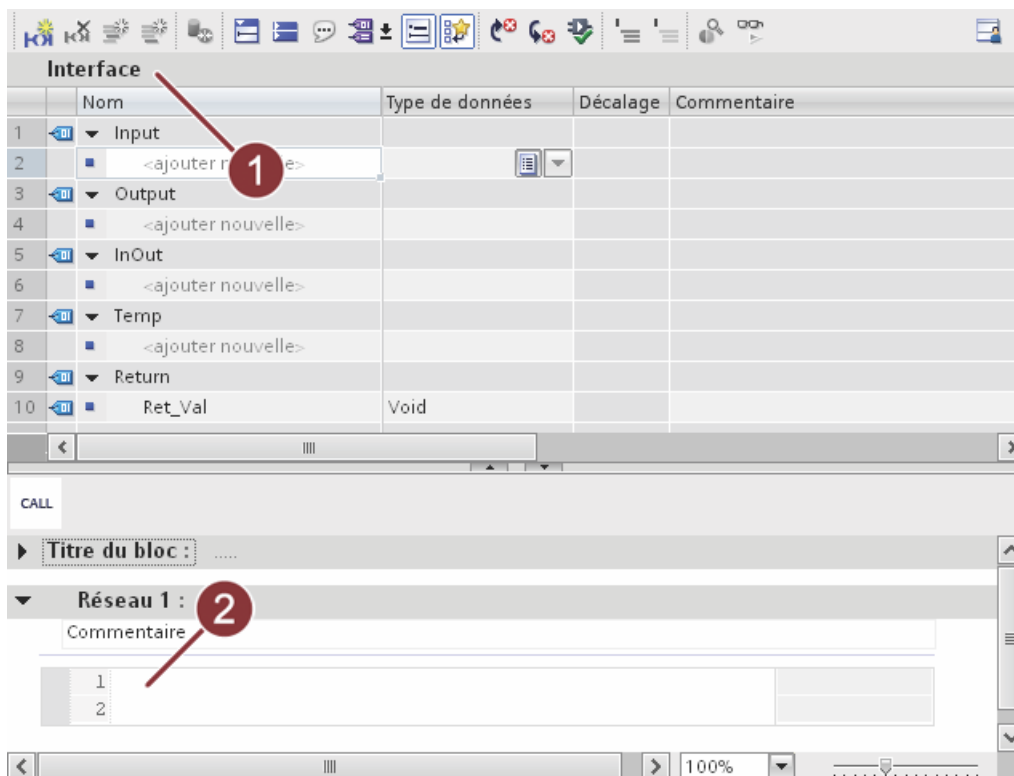
Dans le de projet "Festo Station", nous établissons une fonction en langage LIST pour mettre à l'échelle les grandeurs physique

Définition : Fonction

Une fonction (FC) est un bloc de code sans mémoire. Elle permet de transmettre des paramètres via les interfaces de la fonction. Les fonctions ne peuvent enregistrer les données de manière permanente. Lorsqu'une fonction nécessite certaines données de manière permanente, elle doit être remplacée par un bloc fonctionnel.

Structure de l'éditeur LIST

La figure suivante montre la structure de l'éditeur LIST :



①	Interfaces Dans cette rubrique de l'éditeur, vous définissez les paramètres d'entrée et de sortie avec lesquels la fonction LIST sera commutée.
②	Programmation C'est dans cette zone de l'éditeur que se déroule la programmation à proprement parler de la fonction. La barre des favoris vous permet de sauvegarder des instructions fréquemment utilisées pour la programmation.

Création de la fonction LIST

Pour créer la fonction LIST "mise à l'échelle", on procède comme suit :

1. Dans la navigation du projet, ouvrez le dossier " Blocs de programme".
2. Double-cliquez sur "Ajouter nouveau bloc".
3. Pour ajouter une fonction :
 - Cliquez sur "Fonction".
 - Attribuez le nom de bloc "Mise à l'échelle".



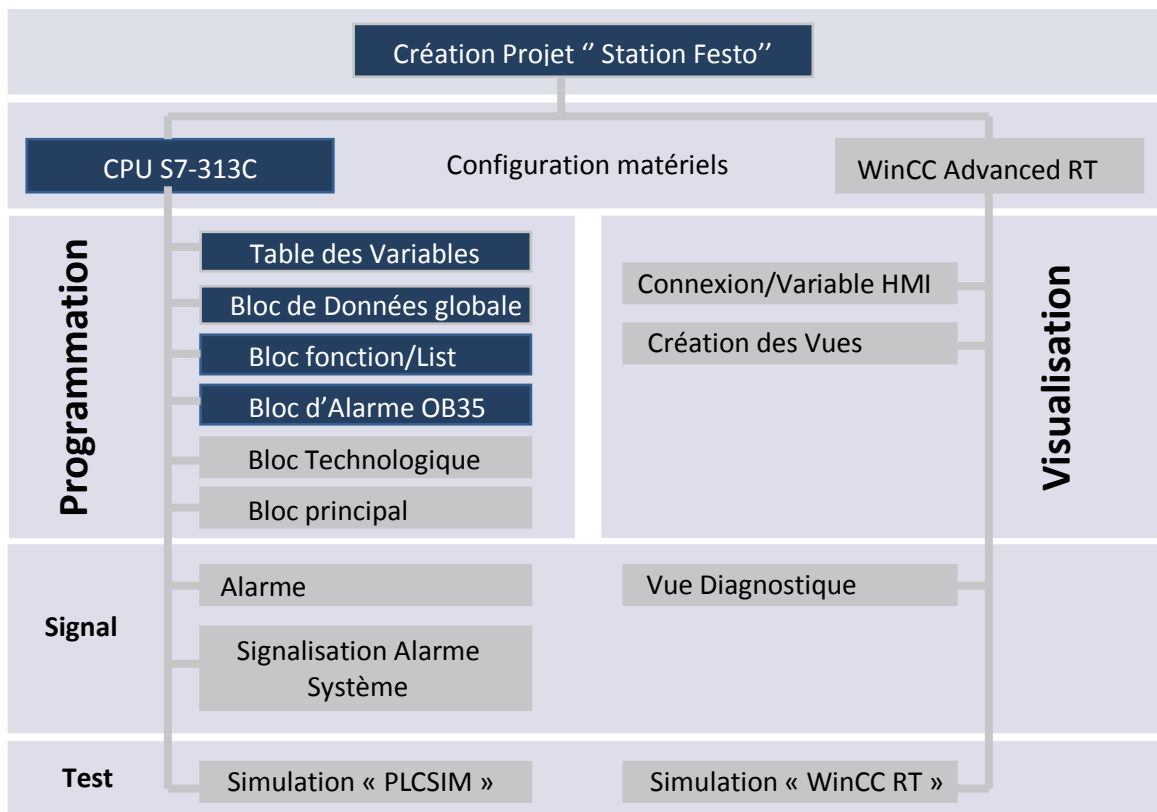
4. Enregistrez le projet

2.8.4 Bloc D'alarme OB35

Ce bloc d'organisation est utilisé pour faire appel au bloc de régulation . En effet un seul bloc PID a été utilisé, par souci d'optimisation, pour toutes les boucles de régulation. Le conflit est évité par un choix au préalable de la régulation (OB35) et du type de cette dernière (FC11).

Avancement du projet

Le graphique suivant nous indique la prochaine étape de configuration que nous exécutons :



La création du bloc OB35 se fait de la même manière que précédemment sauf que le langage de programmation utilisé sera le LADDER.

2.8.5 Bloc technologique PID

Principe de la régulation :

La grandeur à réguler est mesurée de façon continue et est ensuite comparée avec une valeur consigne donnée en entrée du système. La régulation devra, en fonction du résultat de la comparaison, rapprocher la valeur à réguler de la consigne.

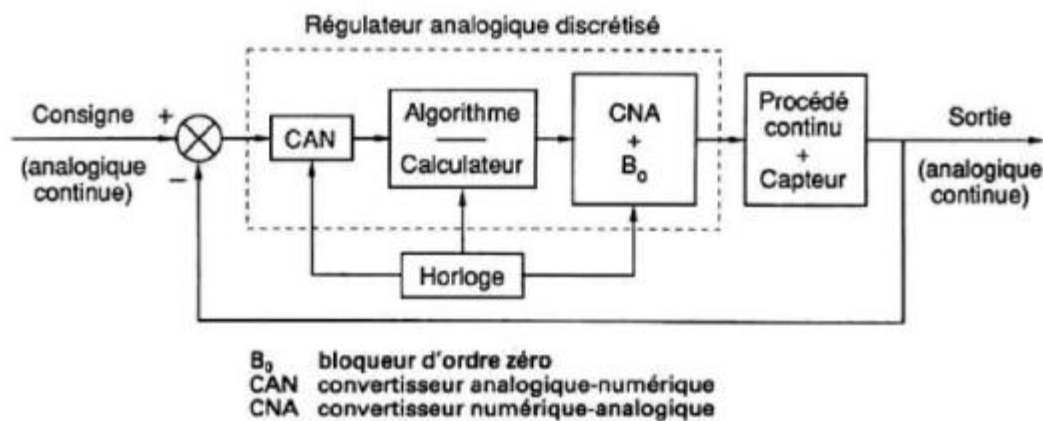


Figure 2.9 : Schéma de principe de la régulation via un API.

Le régulateur est l'élément central d'un asservissement. Il évalue l'erreur calculée par le comparateur, c'est à dire l'écart entre la sortie et la consigne. Il en déduit une valeur de correction à transmettre au procédé afin de corriger la sortie.

Concept de la régulation PID:

Les blocs fonctionnels (FB) de régulation PID comprennent les blocs pour régulation continue (CONT_C) et pour régulation pas à pas (CONT_S) ainsi que le bloc pour la modulation de largeur d'impulsions (PULSEGEN).

Les FB de régulation proposent une régulation purement logicielle, c'est-à-dire qu'un bloc contient toutes les fonctions du régulateur. Les données nécessaires au calcul cyclique sont stockées dans des blocs de données associés, les blocs de données d'instance, ce qui permet aux FB de les appeler plusieurs fois.

Régulation continue avec le bloc FB41 « CONT_C » :

Le bloc FB 41 « CONT_C » sert à régler des processus industriels à grandeurs d'entrée et de sortie continues sur les automates programmables SIMATIC S7. Le paramétrage permet d'activer ou de désactiver des fonctions partielles du régulateur PID et donc d'adapter ce dernier au système réglé.

Le bloc peut être utilisé comme régulateur PID de maintien autonome mais aussi comme régulateur en cascade, de mélange ou de rapport dans des régulations à plusieurs boucles. Sa méthode de travail se base sur l'algorithme PID du régulateur à échantillonnage à sortie analogique, complété le cas échéant par un étage conformateur d'impulsions assurant la formation des signaux de sortie à modulation de largeur d'impulsions pour régulations à deux ou trois échelons, avec organes de réglage proportionnels.

Le schéma fonctionnel de ce bloc est représenté dans la figure ci-dessous :

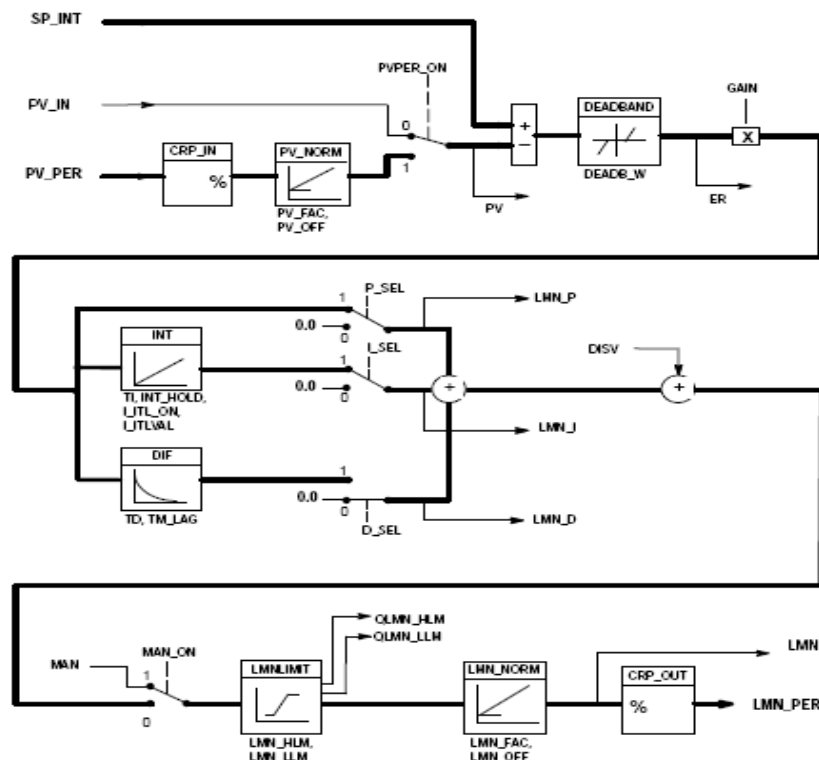


Figure 2.10 : Schéma de principe du bloc PID.

Les tableaux ci-dessous représentent les paramètres d'entrées et de sorties du bloc PID :

- Paramètres d'entrées :

Paramètres d'entrée	Types de données	Valeurs admises	Par défaut	Description
COM_RST	BOOL		FALSE	COMPLETE RESTART / Démarrage Le bloc renferme un sous-programme de démarrage qui est exécuté quand cette entrée est à 1.
MAN_ON	BOOL		TRUE	MANUAL VALUE ON / Activation du mode manuel Quand cette entrée est à 1, la boucle de régulation est interrompue. La valeur de réglage manuelle est sortie comme grandeur de réglage.
PVPER_ON	BOOL		FALSE	PROCESS VARIABLE PERIPHERY ON / Activation de la mesure de périphérie Pour que la mesure soit lue en périphérie, il faut relier l'entrée PV_PER à la périphérie et mettre à 1 l'entrée PVPER_ON.
P_SEL	BOOL		TRUE	PROPORTIONAL ACTION ON / Activation de l'action proportionnelle Dans l'algorithme PID, il est possible d'activer et de désactiver séparément chacune des actions. L'action P est active quand cette entrée est à 1.
I_SEL	BOOL		TRUE	INTEGRAL ACTION ON / Activation de l'action par intégration Dans l'algorithme PID, il est possible d'activer et de désactiver séparément chacune des actions. L'action I est active quand cette entrée est à 1.

INT_HOLD	BOOL		FALSE	INTEGRAL ACTION HOLD / Gel de l'action par intégration La sortie de l'intégrateur peut être gelée. Pour cela, il faut mettre à 1 cette entrée.
I_ITL_ON	BOOL		FALSE	INITIALIZATION OF THE INTEGRAL ACTION / Initialisation de l'action par intégration La sortie de l'intégrateur peut être positionnée sur la valeur initiale I_ITL_VAL. Pour cela, il faut mettre à 1 cette entrée.
D_SEL	BOOL		FALSE	DERIVATIVE ACTION ON / Activation de l'action par dérivation Dans l'algorithme PID, il est possible d'activer et de désactiver séparément chacune des actions. L'action D est active quand cette entrée est à 1.
CYCLE	TIME	≥ 1 ms	T#1 s	SAMPLE TIME / Période d'échantillonnage Le temps s'écoulant entre les appels de bloc doit être constant. Il est indiqué au niveau de cette entrée.
SP_INT	REAL	-100 à 100%	0,0	INTERNAL SETPOINT / Consigne interne Cette entrée sert à introduire une valeur de consigne.
PV_IN	REAL	-100 à 100%	0,0	PROCESS VARIABLE IN / Mesure d'entrée Cette entrée permet de paramétrer une valeur de mise en service ou d'appliquer une mesure externe en virgule flottante.

PV_PER	WORD			<p>PROCESS VARIABLE PERIPHERIE / Mesure de périphérie</p> <p>La mesure en format de périphérie est appliquée au régulateur par cette entrée.</p>
MAN	REAL	-100 à 100%		<p>MANUAL VALUE / Valeur de réglage manuelle</p> <p>Cette entrée sert à introduire une valeur de réglage manuelle moyennant des fonctions de contrôle-commande</p>
GAIN	REAL			<p>PROPORTIONAL GAIN / Coefficient d'action proportionnelle</p> <p>Cette entrée indique le gain du régulateur.</p>
TI	TIME	>= cycle		<p>RESET TIME / Temps d'intégration</p> <p>Cette entrée détermine la réponse temporelle de l'intégrateur</p>
TD	TIME	>= cycle		<p>DERIVATIVE TIME / Temps de dérivation</p> <p>Cette entrée détermine la réponse temporelle de l'unité de dérivation.</p>
TM_LAG	TIME	>= cycle/2		<p>TIME LAG OF THE DERIVATE ACTION / Retard de l'action par dérivation</p> <p>L'algorithme de l'action D contient un retard qui peut être paramétré à cette entrée</p>

DEADB_W	REAL	$\geq 0,0\%$		DEAD BAND WIDTH / Largeur de zone morte Le signal d'erreur traverse une zone morte. Cette entrée détermine la taille de la zone morte
LMN_HLM	REAL	LMN_LLM à 100%		MANIPULATED VALUE HIGH LIMIT / Limite supérieure de la valeur de réglage La valeur de réglage est toujours bornée à une limite supérieure et une limite inférieure. Cette entrée indique la limite supérieure.
LMN_LLM	REAL	-100% à LMN_HLM		MANIPULATED VALUE LOW LIMIT / Limite inférieure de la valeur de réglage La valeur de réglage est toujours bornée à une limite supérieure et une limite inférieure. Cette entrée indique la limite inférieure.
PV_FAC	REAL			PROCESS VARIABLE FACTOR / Facteur de mesure Cette entrée est multipliée par la mesure. Elle sert à adapter l'étendue de valeur de mesure.
PV_OFF	REAL			PROCESS VARIABLE OFFSET / Décalage de mesure Cette entrée est ajoutée à la mesure. Elle sert à adapter l'étendue de valeur de mesure.

LMN_FAC	REAL			<p>MANIPULATED VALUE FACTOR / Facteur de valeur de réglage</p> <p>Cette entrée est multipliée par la valeur de réglage. Elle sert à adapter l'étendue de réglage.</p>
LMN_OFF	REAL			<p>MANIPULATED VALUE OFFSET / Décalage de valeur de réglage</p> <p>Cette entrée est ajoutée à la valeur de réglage. Elle sert à adapter l'étendue de réglage.</p>
I_ITLVAL	REAL	-100 à 100%		<p>INITIALIZATION VALUE OF THE INTEGRAL ACTION /</p> <p>Valeur d'initialisation pour l'action par intégration</p> <p>La sortie de l'intégrateur peut être forcée par l'entrée I_ITL_ON. La valeur d'initialisation est appliquée à l'entrée I_ITLVAL.</p>
DISV	REAL	-100 à 100%		<p>DISTURBANCE VARIABLE /</p> <p>Grandeur perturbatrice</p> <p>La grandeur perturbatrice est appliquée à cette entrée pour l'action anticipatrice.</p>

Tableau 2.1 : Entrées du bloc PID.

- Paramètres de sortie :

Paramètre	Type de données	Valeurs admises	Par défaut	Description
LMN	REAL		0.0	MANIPULATED VALUE / Valeur de réglage Cette sortie donne en virgule flottante la valeur de réglage agissant réellement
LMN_PER	WORD		W#16#0000	MANIPULATED VALUE PERIPHERY / Valeur de réglage de périphérie Cette sortie fournit la valeur de réglage en format de périphérie
QLMN_HLM	BOOL		FALSE	HIGH LIMIT OF MANIPULATED VALUE REACHED / Grandeur de réglage à la limite supérieure La valeur de réglage est toujours bornée à une limite supérieure et à une limite inférieure. Cette sortie signale le dépassement de la limite supérieure
QLMN_LLM	BOOL		FALSE	LOW LIMIT OF MANIPULATED VALUE REACHED / Grandeur de réglage à la limite inférieure La valeur de réglage est toujours bornée à une limite supérieure et à une limite inférieure. Cette sortie signale le dépassement de la limite inférieure

LMN_P	REAL		0.0	PROPORTIONALITY COMPONENT / Composante P Cette sortie contient la composante proportionnelle de la grandeur de réglage.
LMN_I	REAL		0.0	INTEGRAL COMPONENT / Composante I Cette sortie contient la composante intégrale de la grandeur de réglage.
LMN_D	REAL		0.0	DERIVATIVE COMPONENT / Composante D Cette sortie contient la composante différentielle de la grandeur de réglage.
PV	REAL		0.0	PROCESS VARIABLE / Mesure Cette sortie donne la mesure effective
ER	REAL		0.0	ERROR SIGNAL / Signal d'erreur Donne le signal d'erreur effectif

Tableau 2.2 : Sorties du bloc PID.

Pour une CPU donnée, il faut trouver un compromis entre le nombre de régulateurs et la fréquence de traitement de chacun d'eux. Plus les boucles de régulation seront rapides, c'est à dire plus souvent les grandeurs à réguler sont à calculer par unité de temps, plus faible sera le nombre de régulateurs.

Il n'y a pas de restriction concernant le type de processus à mettre en œuvre. Les blocs fonctionnels s'appliquent aux systèmes inertiels (températures, niveaux de remplissages, etc.) comme aux systèmes très rapides (débit, vitesse de rotation, etc.).

Les courbes de la valeur souhaitée (consigne), la valeur réelle (mesure) et la grandeur d'ajustement (commande) peuvent être visualisées en temps réel grâce au graphique.

2.8.6 Appel des blocs de programme dans le bloc d'organisation "Main"

Vue d'ensemble de la structure d'appel

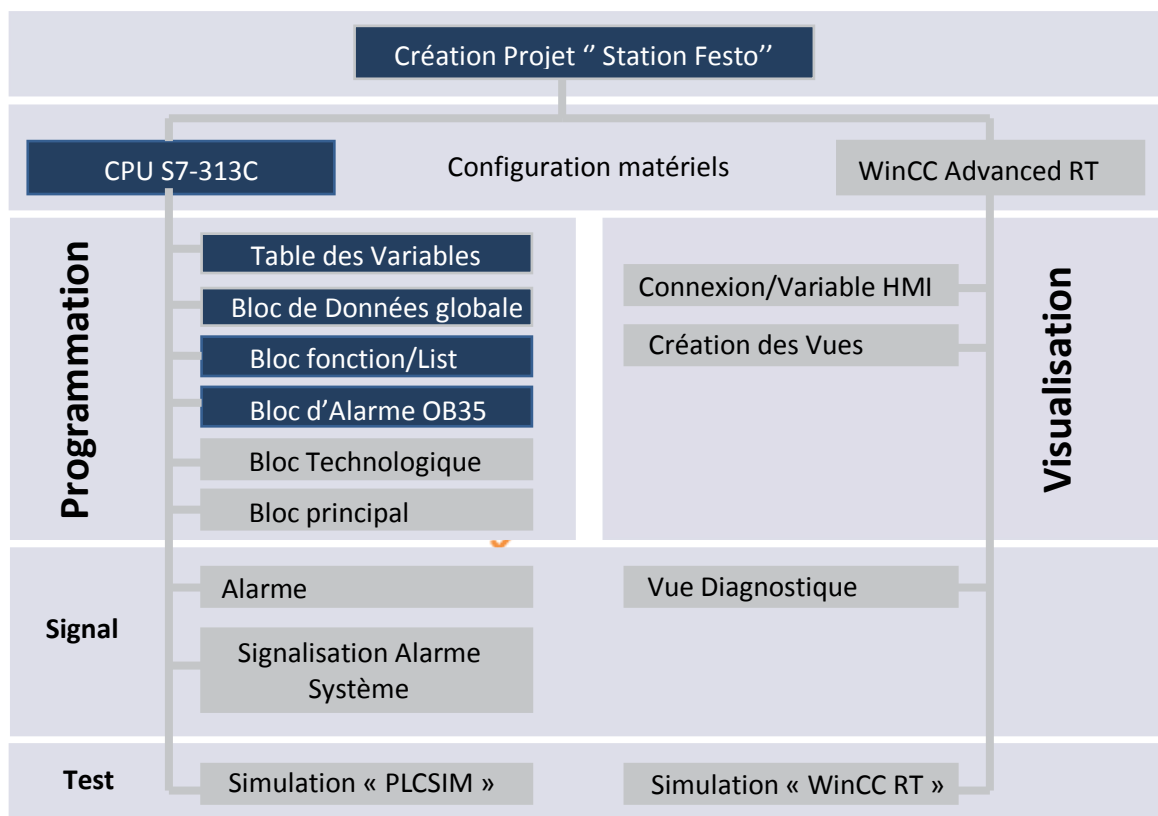
Pour que nos blocs de programme soient exécutés dans le programme utilisateur, ils doivent être appelés par un autre bloc. Dans l'exemple de projet "Festo Station", utilisez à cet effet le bloc d'organisation "Main". Si le bloc d'organisation "Main" appelle un bloc de programme, les instructions du bloc appelé sont traitées. Le bloc d'organisation "Main" est exécuté en parallèle de manière cyclique.

Définition du bloc d'organisation cyclique

Le bloc d'organisation "Main" est un bloc d'organisation pour le traitement cyclique de programmes et a été créé automatiquement lors de la création de la CPU dans la navigation de projet du dossier "Blocs de programme". Pour que le traitement du programme démarre, le projet doit posséder au moins un OB cyclique. Le système d'exploitation appelle cet OB et lance ainsi le traitement du programme utilisateur.

Avancement du projet

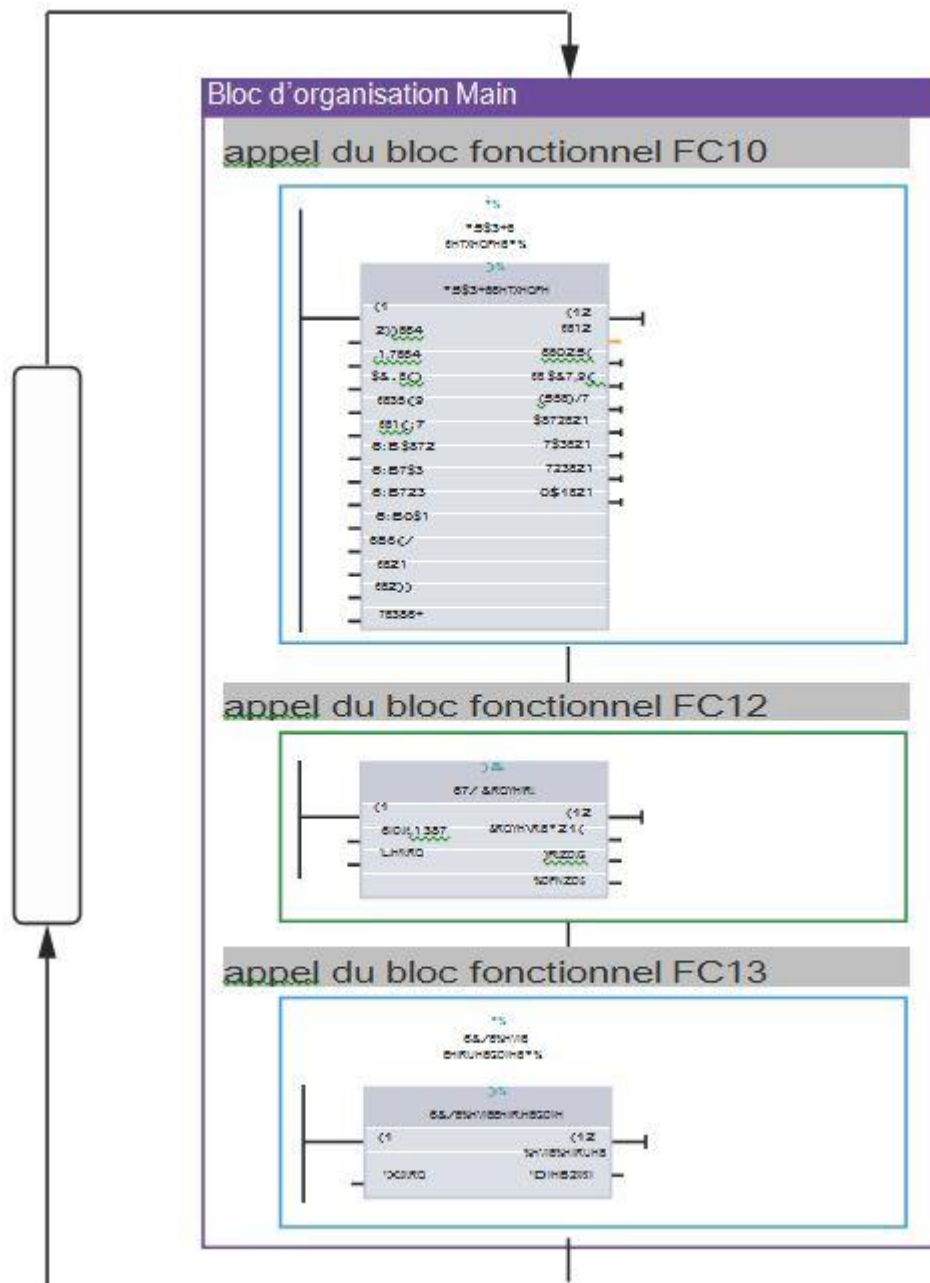
Le graphique suivant nous indique la prochaine étape de configuration que nous exécutons :



Les blocs de programme sont appelés au sein du bloc d'organisation "Main" dans l'ordre suivant :

1. Réseau 1 : FC10 "mise à l'échelle"
2. Réseau 2 : FC11 "choix de type de la régulation"
3. Réseau 3 : FC12 "arrêt/stop"
4. Réseau 4 : FC13 "Reset"

Le graphique suivant affiche le traitement cyclique du programme des blocs programmés :



2.9 Visualisation du processus :

Dans le projet "Festo Station", on crée plusieurs vues IHM. Ces vues nous permettent de visualiser le déroulement complet de la régulation que nous avons déjà programmée dans le chapitre précédent.

Des objets prédéfinis sont à notre disposition pour créer les vues nécessaires, objets qui nous permettent de reproduire le déroulement de la régulation. Les séquences de processus peuvent ainsi être affichées et les valeurs de processus saisies

Définition de l'interface homme-machine (IHM)

Le système d'interface homme-machine (IHM) constitue l'interface entre l'opérateur et le processus. Le déroulement du processus est commandé par la CPU. L'opérateur peut visualiser le processus ou intervenir dans le processus en cours par le biais d'un pupitre opérateur.

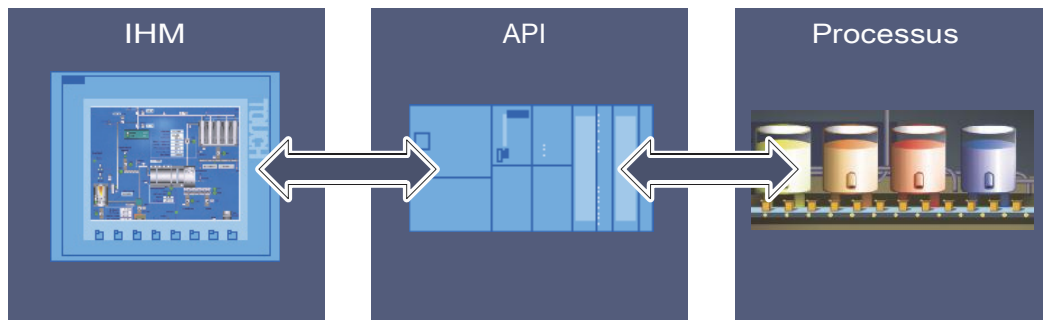


Figure 2.12: HMI TIA PORTAL [1].

Les possibilités suivantes vous sont en autres offertes pour le contrôle-commande des machines et installations :

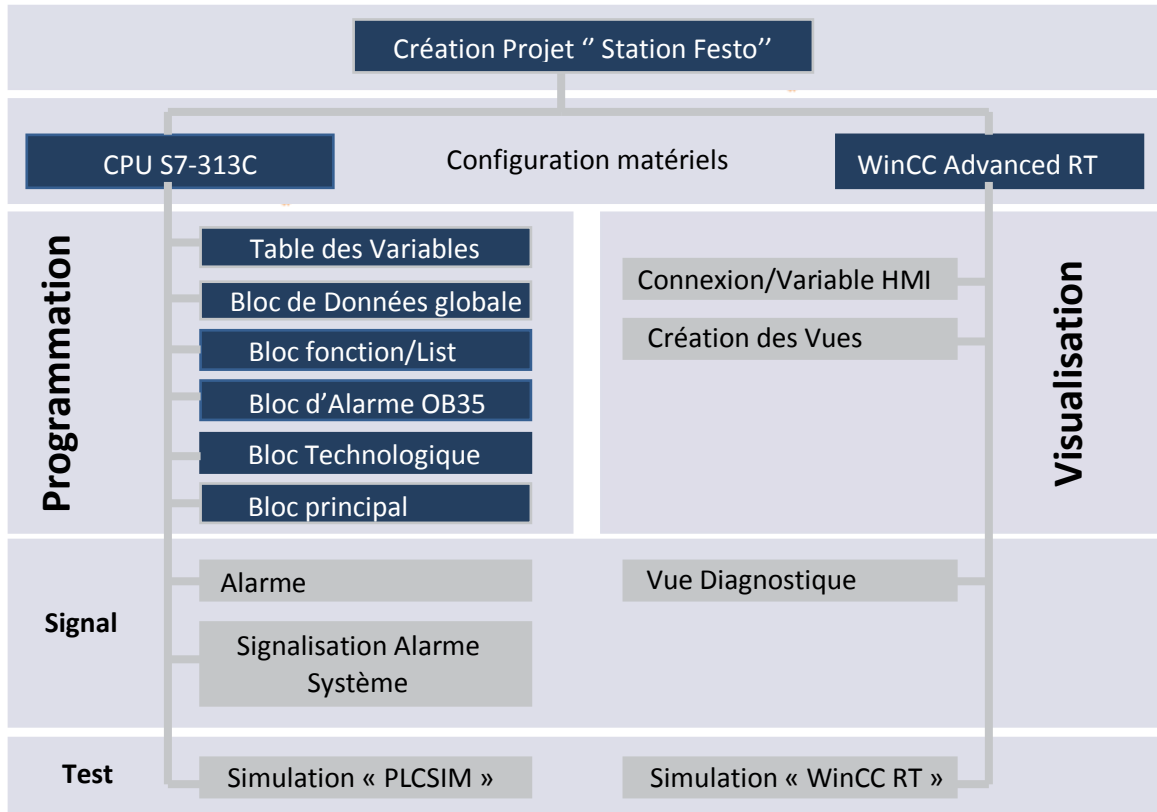
- Représenter les processus
- Commander les processus
- Emettre des alarmes
- Gérer les paramètres du processus et les recettes

2.9.1 Configuration du WinCC Advanced Runtime

Nous allons maintenant créer le HMI « SIMATIC PC station » et réaliser la configuration des vues. Ces vues IHM vous permettent de visualiser les différentes parties du processus :

Avancement du projet

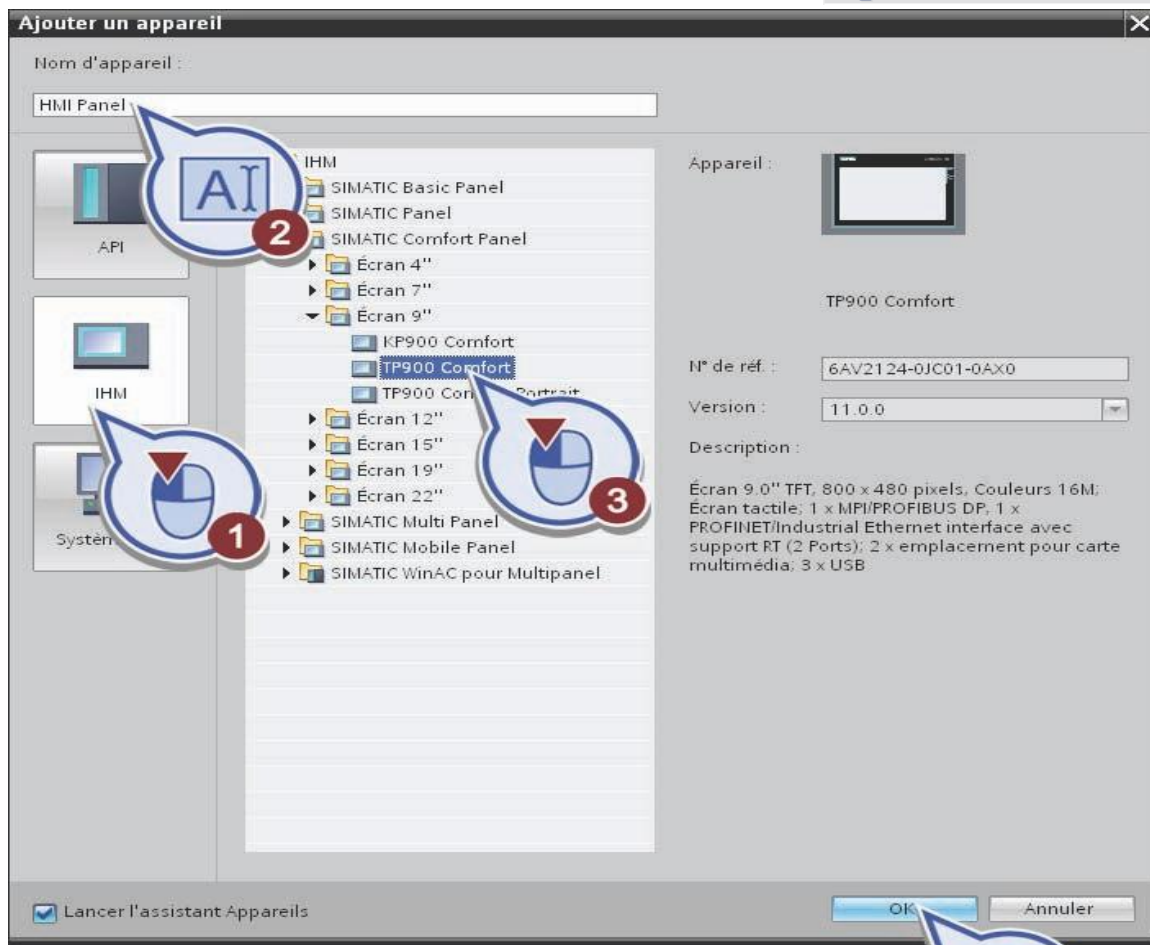
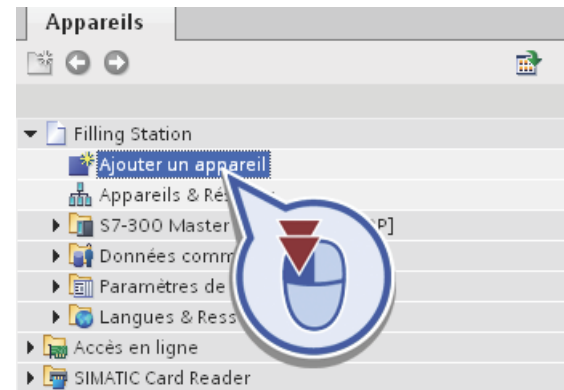
Le graphique suivant nous indique la prochaine étape de configuration que nous exécutons :



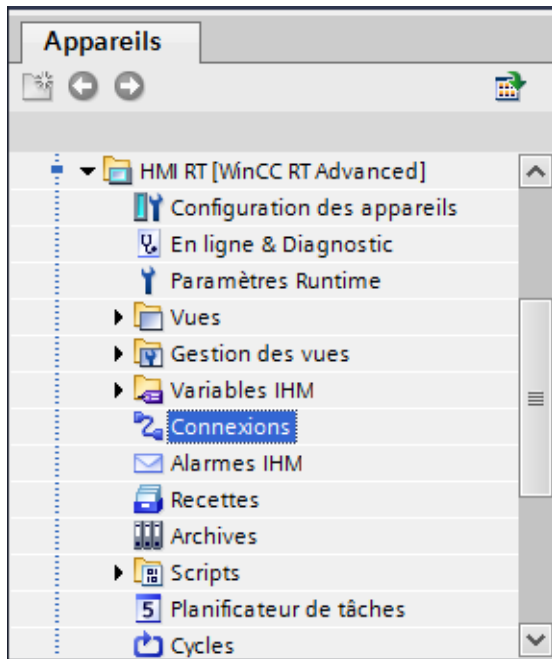
Marche à suivre

Créez le HMI PC station comme suit :

1. Double-cliquez dans la navigation du projet sur "Ajouter un appareil".
2. Effectuez dans la boîte de dialogue "Ajouter un appareil" les paramètres suivants :
 - Cliquez sur "Système PC".
 - Entrez "logiciel SIMATIC HMI Runtime" comme nom de l'appareil.
 - Sélectionnez le "WinCC RT advanced".



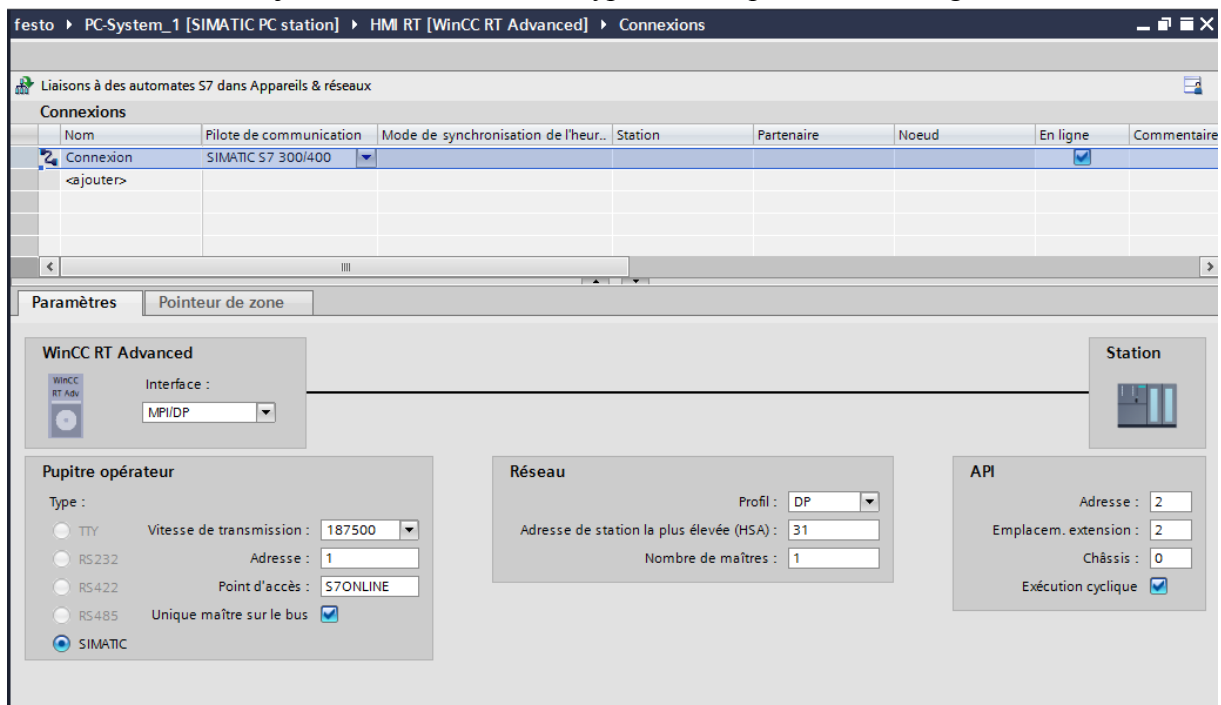
2.9.2 Configuration de la connexion entre API et WinCC Advanced RT :



Pour établir la connexion entre l'API et WinCC, on utilise une connexion de type MPI à l'aide de L'adaptateur USP/MPI

Ouvrir l'éditeur de configuration de connexions en appuyant sur « connexion »

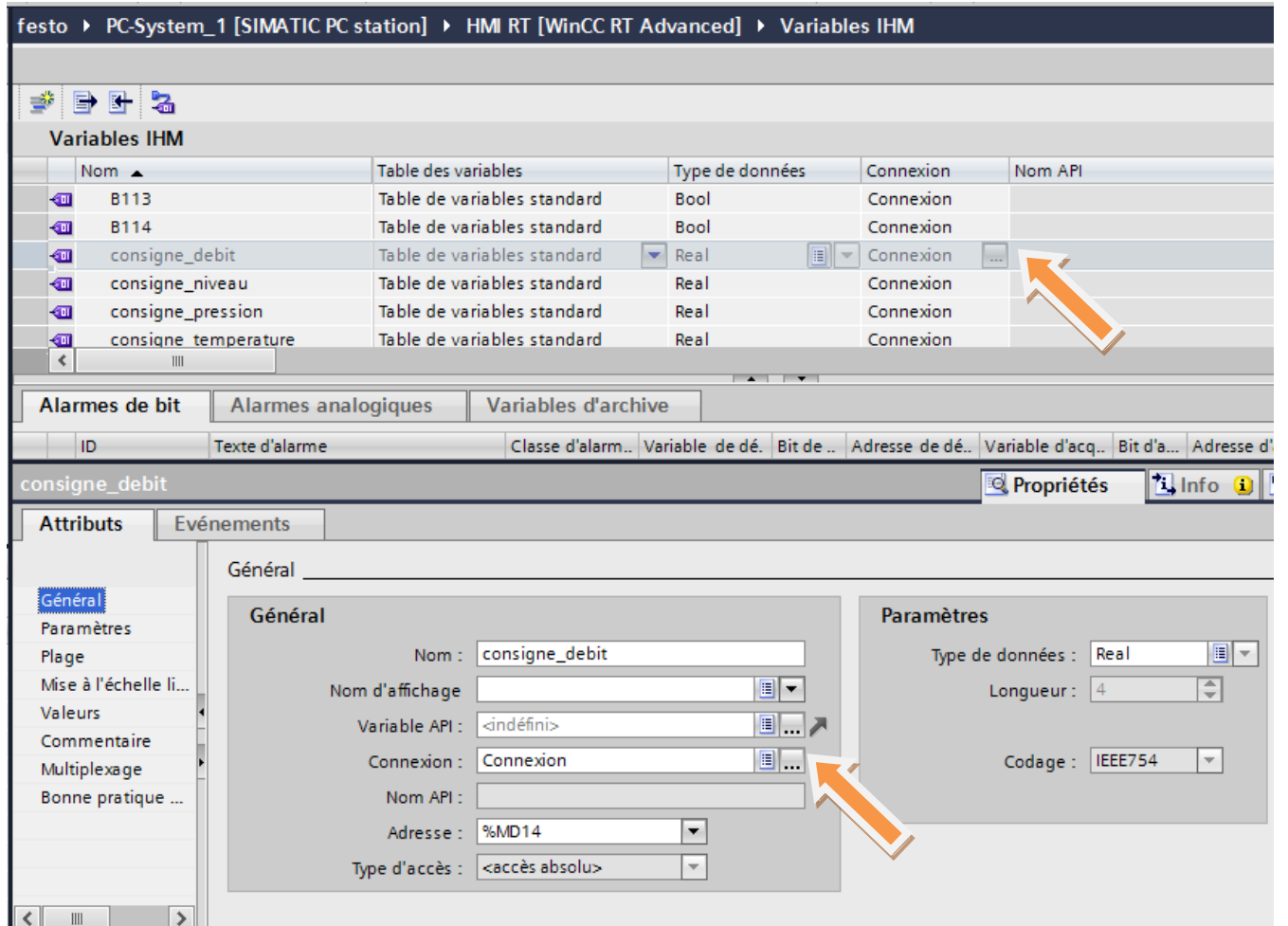
Dans l'éditeur, on ajoute une connexion de type MPI en prenant la configuration suivante :



1-on choisit MPI/DP pour l'interface

2-on spécifie l'adresse API et du Wincc RT Advanced qui doit toujours être '1'

Maintenant chaque fois qu'une variable API est utilisé dans WinCC on doit lui associe une variable équivalente connecté avec la connexion précédemment configuré



2.9.3 Création des Vues :

Pour la création des vue, on dispose dans l'éditeur des vue d'une large palette D'objet permettant de créer des visualisations du processus des courbes, et même l'archivage des variables.

2.10 Test en ligne du projet

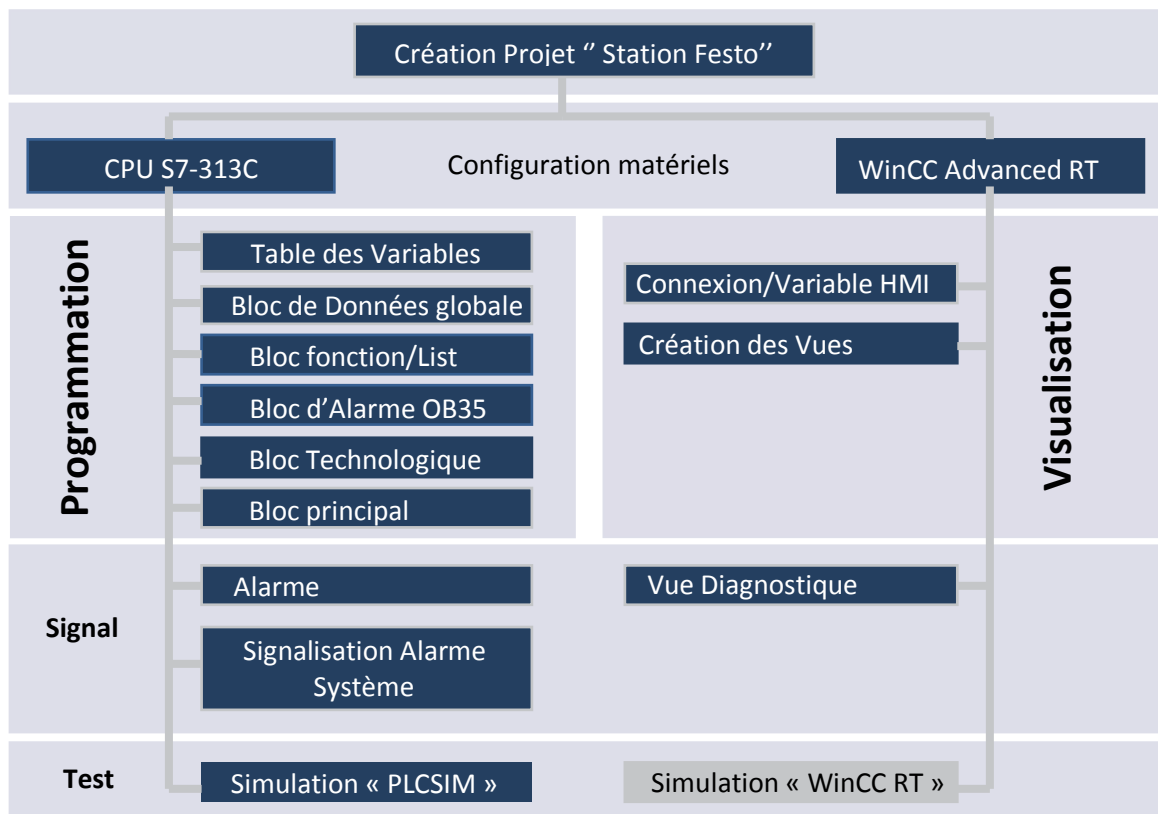
Test du programme

Démarrer la simulation dans PLCSIM

Nous allons maintenant tester la fonctionnalité du programme avec le logiciel de simulation PLCSIM. Cette simulation nous permet de tester le bon fonctionnement du programme avant de démarrer la production. Nous chargeons d'abord la configuration et le programme utilisateur dans le module simulé et nous insérons des sous-fenêtres pour visualiser et forcer les sorties et mémentos.

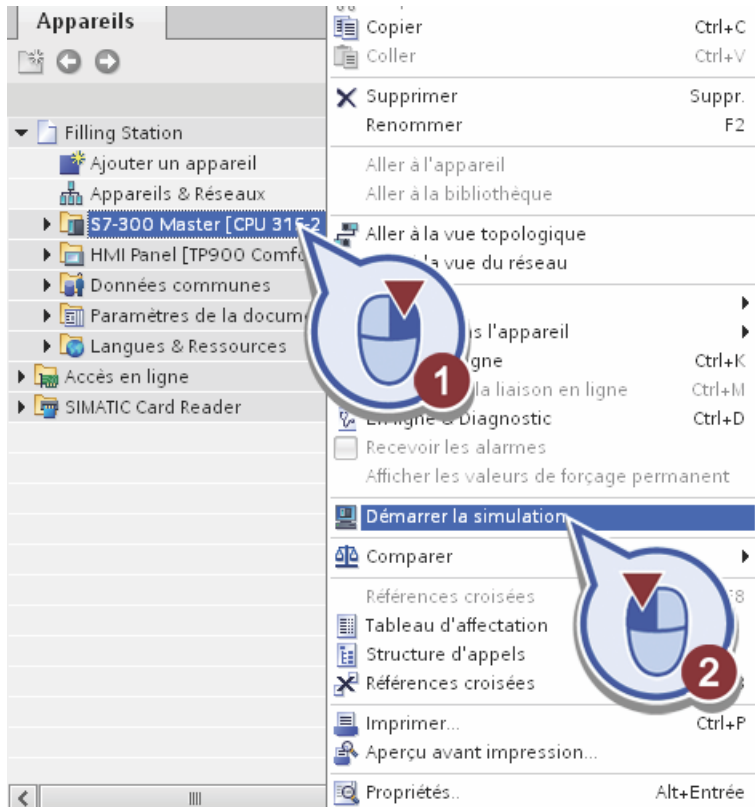
Avancement du projet

Le graphique suivant nous indique la prochaine étape de configuration que nous exécutons :

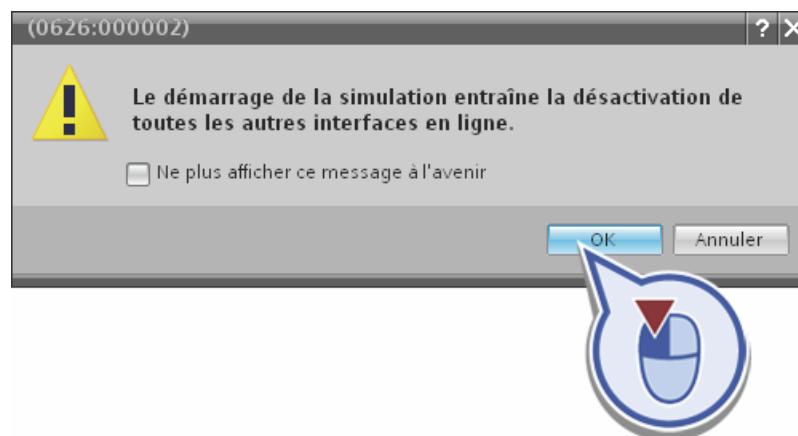


On Procède comme suit pour démarrer le logiciel PLCSIM :

1. Dans la navigation du projet, cliquez droit avec la souris sur la CPU "S7-313C". Sélectionnez "Démarrer la simulation" dans le menu contextuel



2. Confirmer le dialogue suivant avec "OK".

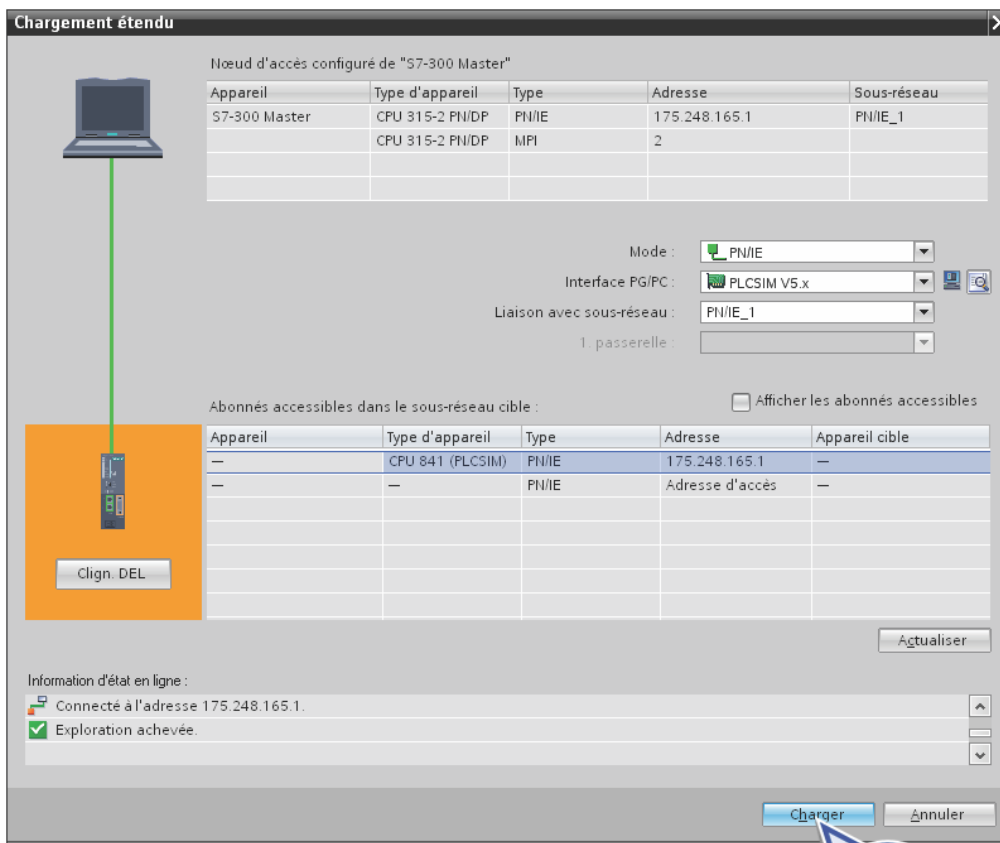


Le logiciel PLCSIM démarre à l'arrière-plan et la boîte de dialogue "Chargement étendu" s'ouvre.

3. Choisissez dans la fenêtre de dialogue "Chargement étendu" les réglages suivants :

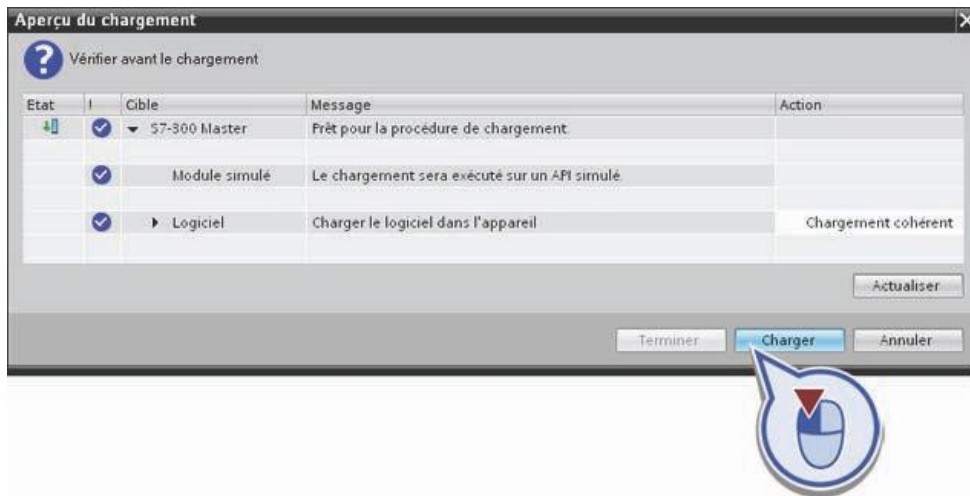
- Type de l'interface PG/PC : PN/IE
- Interface PG/PC : PLCSIM V5.x
- Liaison avec sous-réseau : PN/IE_1

Cliquez ensuite sur le bouton "Charger".

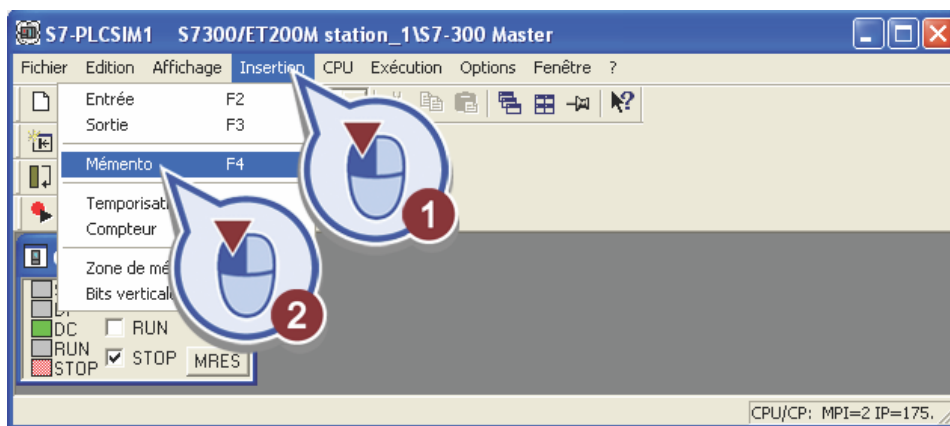


La boîte de dialogue "Compiler" est exécutée

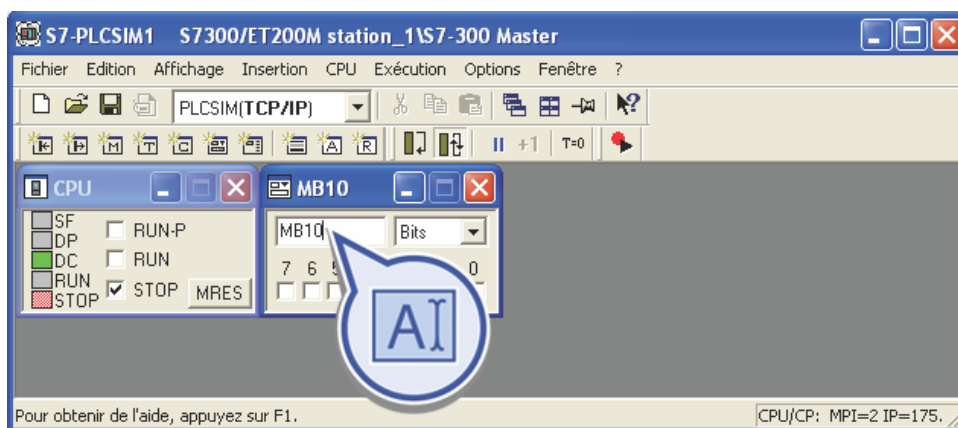
4. Confirmez la procédure de chargement dans le module simulé via PLCSIM.



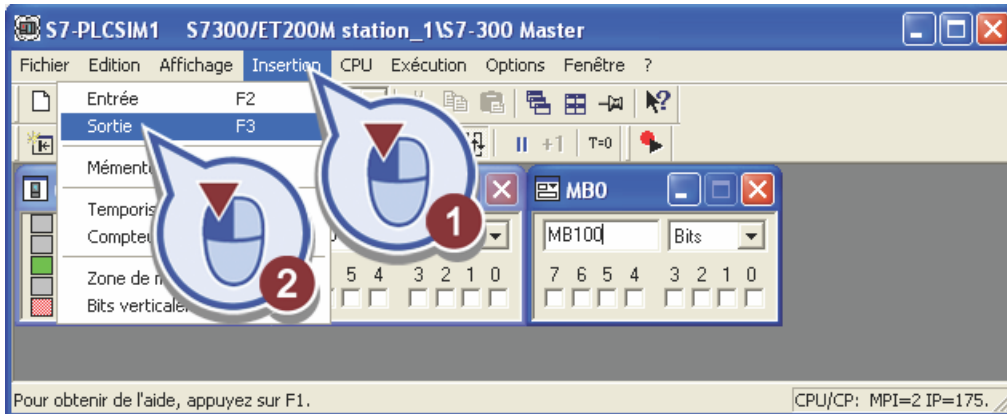
5. Insérez une sous-fenêtre "Memento" pour forcer et afficher les variables API avec bits de memento dans la page d'adresse voulu.



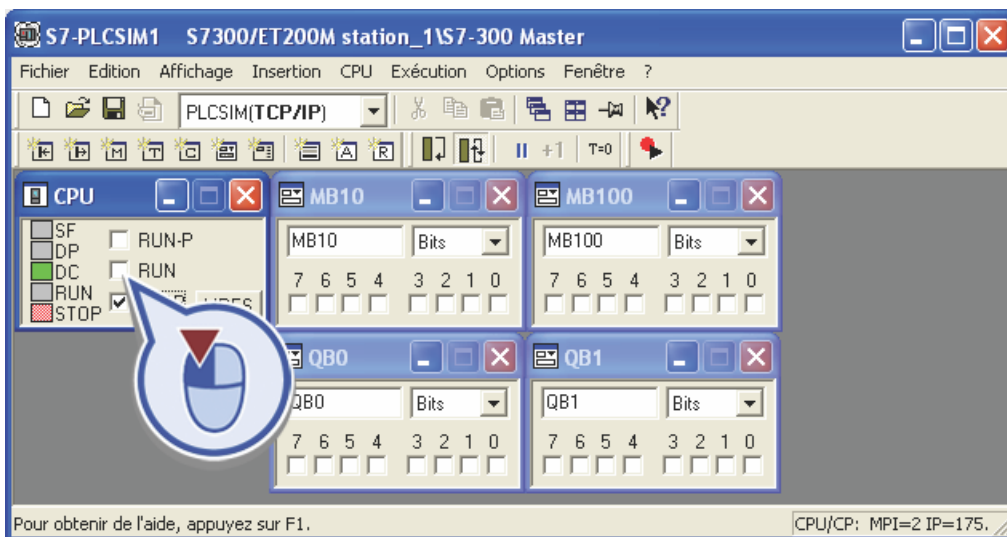
6. Entrez par exemple l'octet de memento MB10 comme plage d'adresse. Laissez le format sur "Bits".



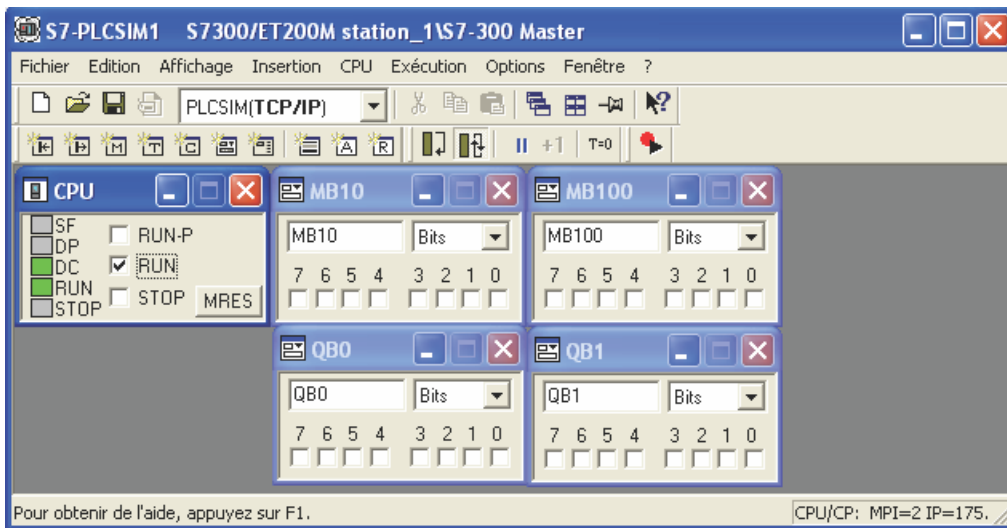
7. Insérez une sous-fenêtre "Sortie" pour délivrer les sorties dans la plage d'adresse des module



8..Mettez la simulation de la CPU en mode RUN.



Nous avons configuré la simulation de la CPU dans PLCSIM. Une fois en RUN, les deux LED "RUN" (mode de fonctionnement) et "DC" (alimentation) sont allumées en vert.



Laissez la fenêtre PLCSIM ouverte. nous allons maintenant nous servir de la simulation pour tester l'exemple de projet "Festo Station" avec les fonctions en ligne et de diagnostic.

2.11 Tester la visualisation du processus

Démarrer WinCC Advanced Runtime :

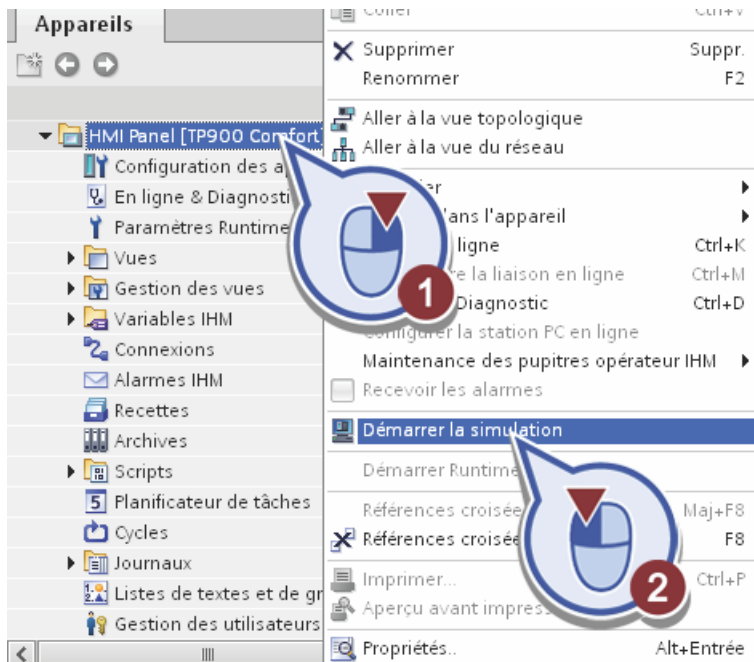
La fonctionnalité de la visualisation peut être testée avec le logiciel de simulation "WinCC Runtime Advanced". Nous allons maintenant démarrer la simulation du HMI. Cette simulation nous permet de tester le bon fonctionnement de la visualisation avant de démarrer du processus

Condition :

La configuration matérielle et les blocs de programme de la CPU sont chargés dans PLCSIM, le portail TIA est dans la vue en ligne. Le logiciel "WinCC Runtime Advanced" a été installé avec le portail TIA.

Marche à suivre :

1. Cliquez droit avec la souris sur le HMI RT dans la navigation du projet et démarrez la simulation du Runtime dans le menu contextuel.



Les éléments configurés du HMI RT sont automatiquement compilés avant le démarrage du Runtime. L'état de compilation s'affiche dans la fenêtre d'inspection de l'onglet "Info".

Résultat :

Le logiciel "WinCC Runtime Advanced" est démarré. La première vue qui s'affiche est la vue racine "Production".

Tester la visualisation du processus

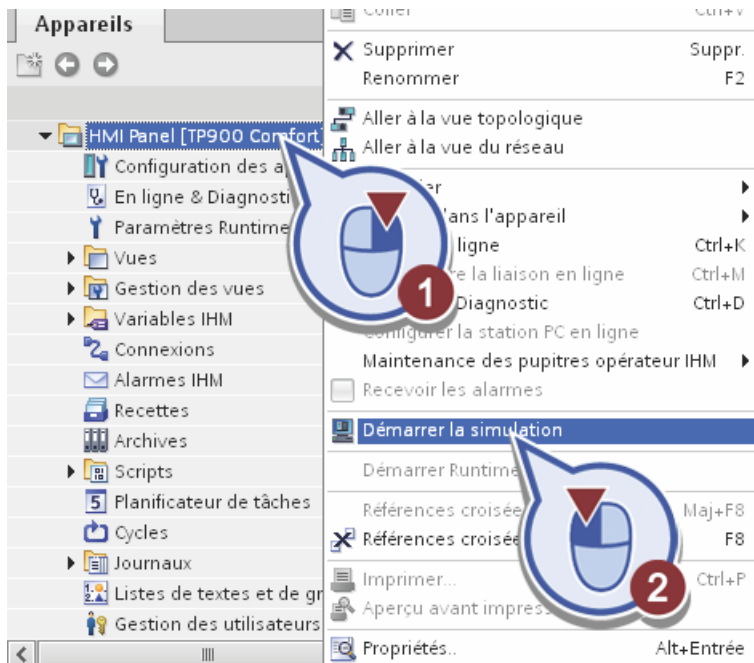
La fonctionnalité de la visualisation peut être testée avec le logiciel de simulation "WinCC Runtime Advanced". Nous allons maintenant démarrer la simulation du HMI. Cette simulation nous permet de tester le bon fonctionnement de la visualisation avant de démarrer du processus.

Condition :

La configuration matérielle et les blocs de programme de la CPU sont chargés dans PLCSIM, le portail TIA est dans la vue en ligne. Le logiciel "WinCC Runtime Advanced" a été installé avec le portail TIA.

Marche à suivre :

1. Cliquez droit avec la souris sur le HMI RT dans la navigation du projet et démarrez la simulation du Runtime dans le menu contextuel.



Les éléments configurés du HMI RT sont automatiquement compilés avant le démarrage du Runtime. L'état de compilation s'affiche dans la fenêtre d'inspection de l'onglet "Info".

Test de la vue principale

Nous allons tout d'abord tester les fonctions de choix de type de régulation.

Condition requise

Le programme API a été chargé dans PLCSIM et la CPU se trouve dans l'état de fonctionnement RUN-P. Le HMI RT a été chargé dans le logiciel de simulation "WinCC Runtime Advanced".

1. dans la vue principale on choisit une par une les boucle de régulation.

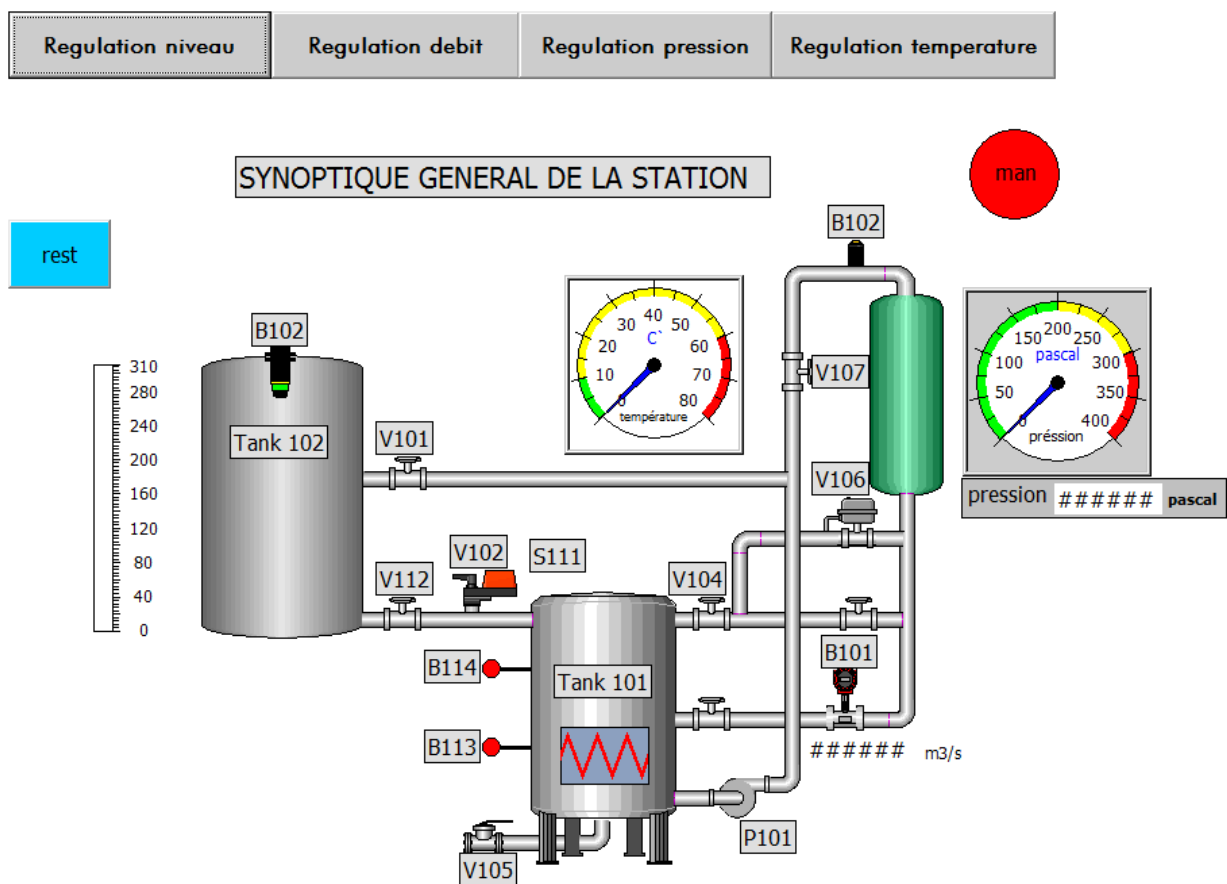
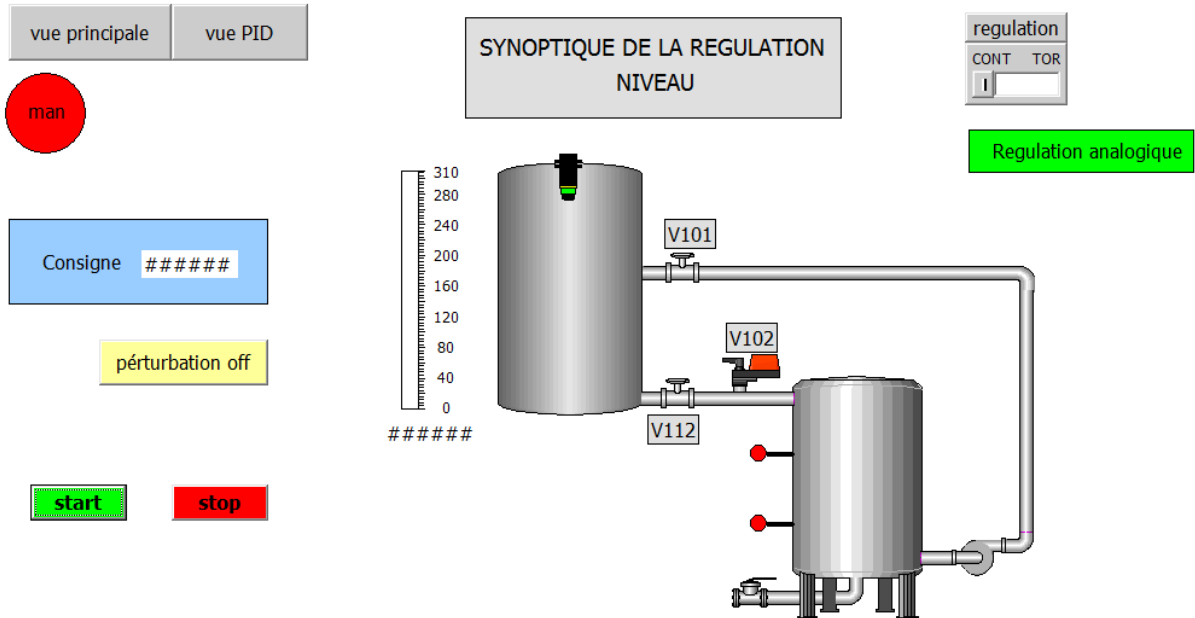
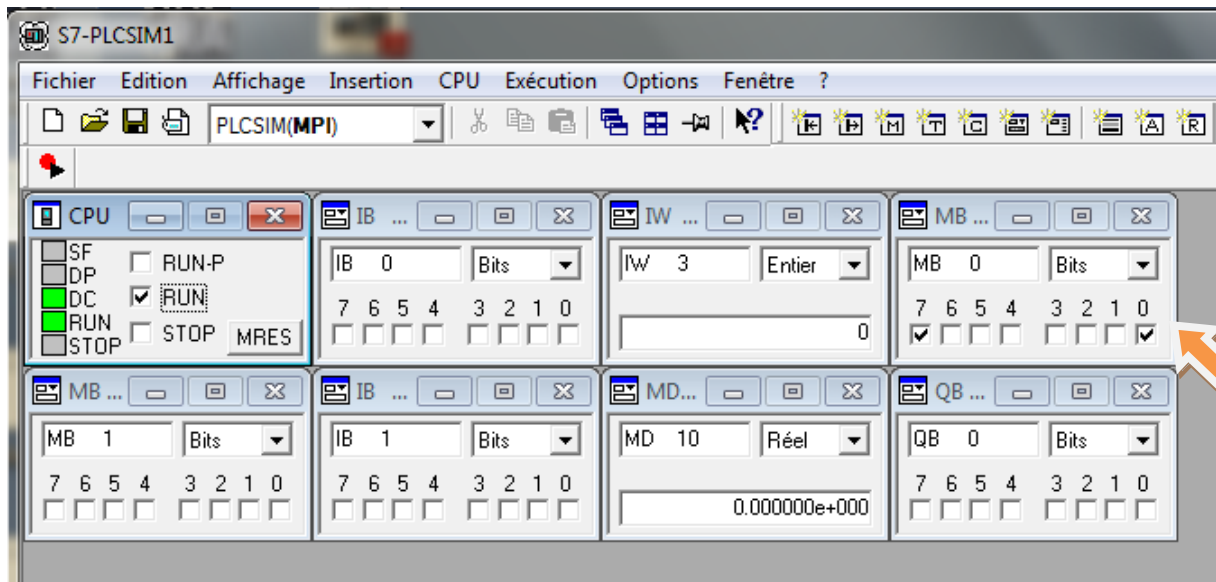


Figure 2.13: VUE PRINCIPALE.

3. chaque fois qu'on ouvre une vue de boucle de régulation, on vérifie dans PLCSIM, Si les mnémonique relatifs à la régulation choisi se sont mise à jour.



On voit bien dans PLCSIM que MB0.0 et MB0.7 sont activé



1. pour chaque régulation on test si les données des processus se transmettent bien entre HMI et API, par exemple dans l'interface on donne une valeur pour la variable MD10 qui correspond à la consigne niveau

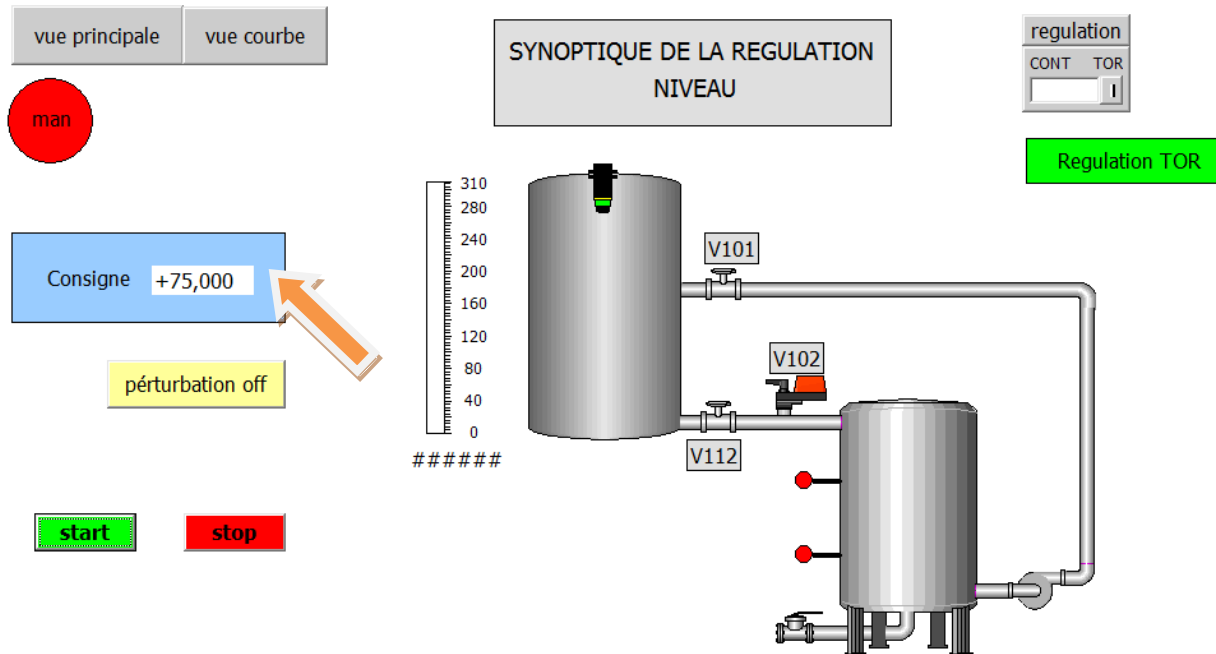
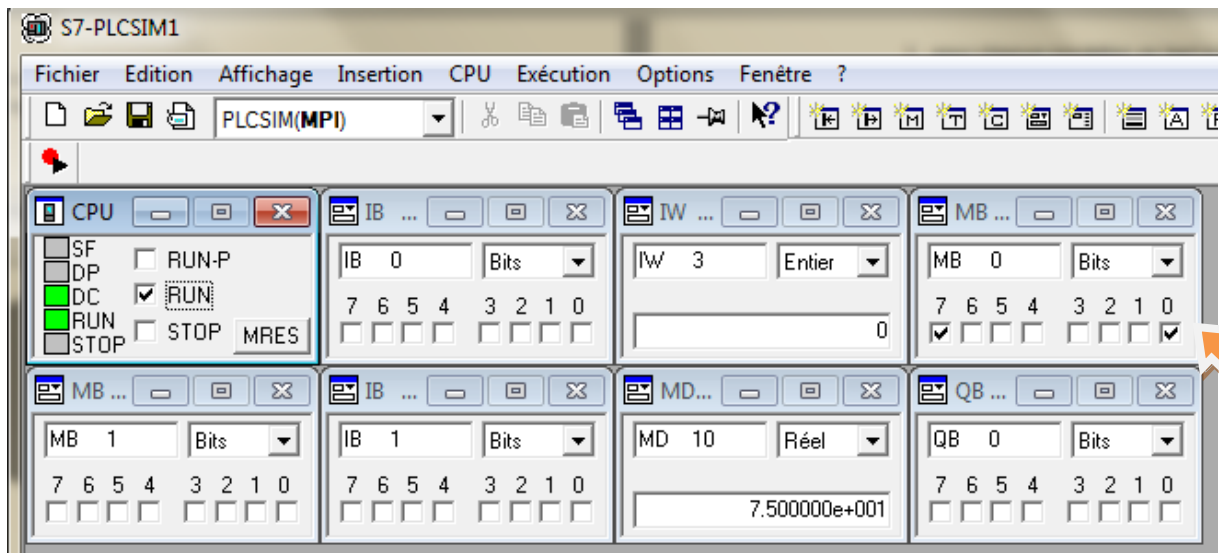


Figure 2.14: VUE REGULATION NIVEAU.



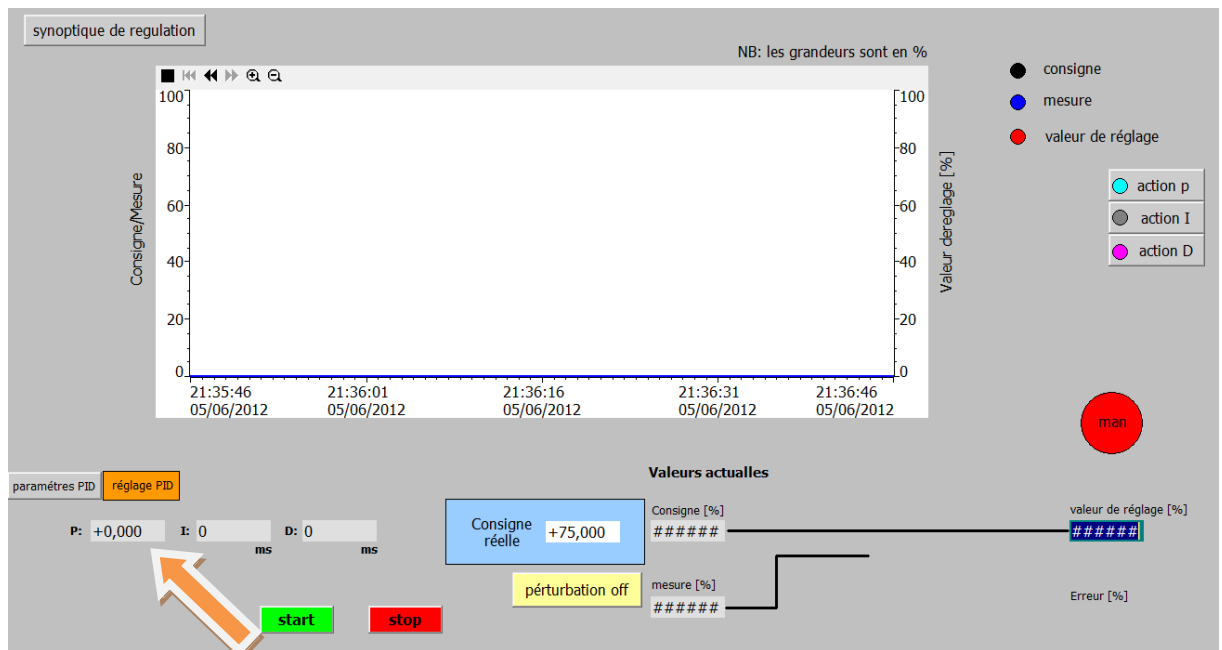
2. on peut vérifier dans PLCSIM que la valeur a bien été transmis à la variable MD10

3. on refait le même travail avec toutes les variables

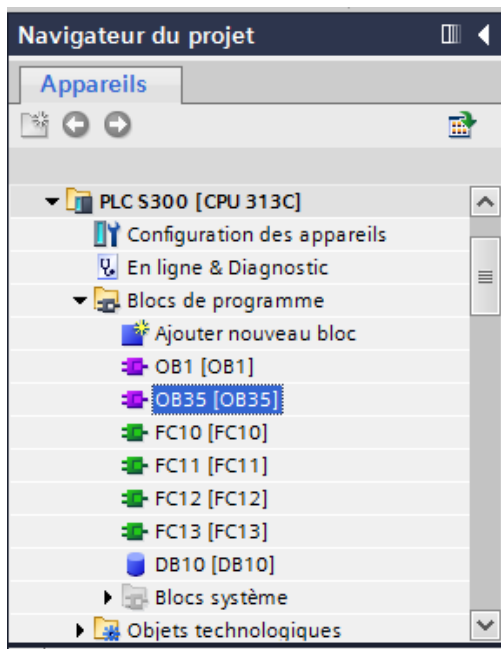
Test de la vue PID

Pour la vue PID on ne peut pas le faire avec PLCSIM, le test doit être passé en condition réelle

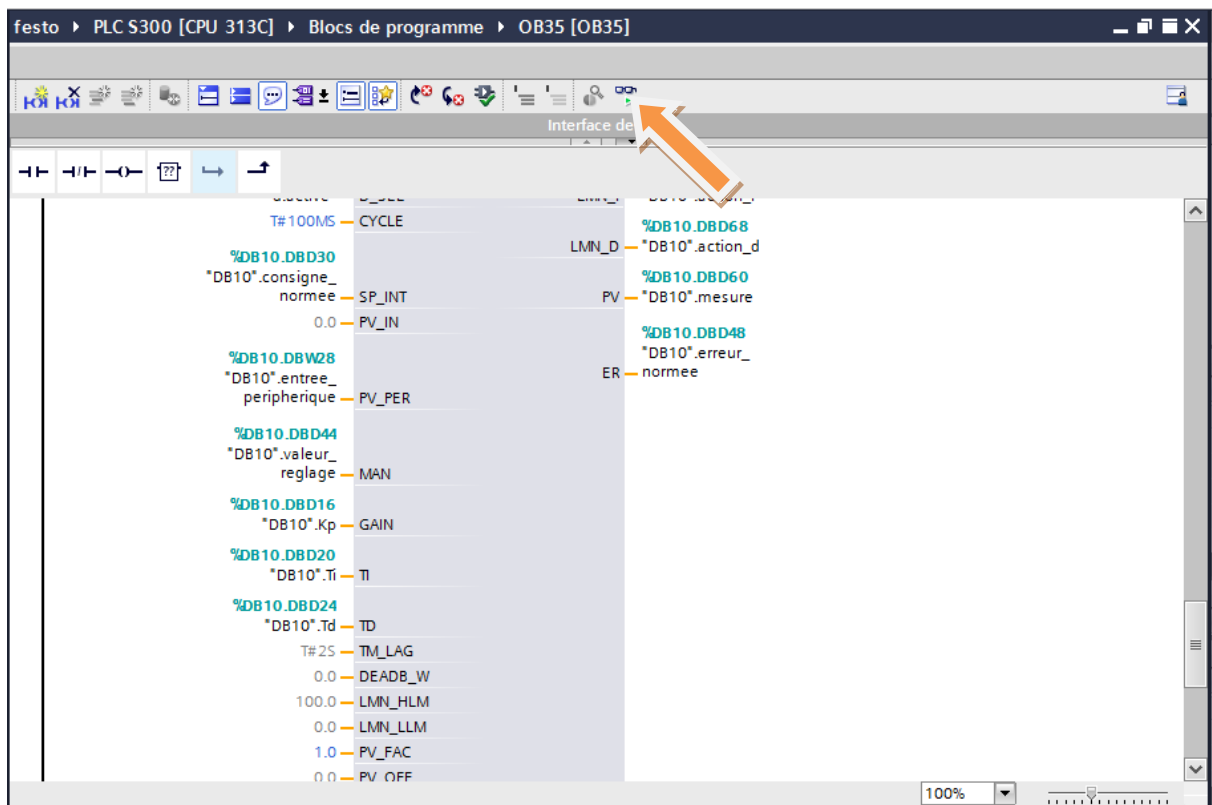
1. on doit vérifier si les paramètres PID se transmettent bien vers le bloque PID ,dans la vue PID on introduit les paramètres comme le montre la figure suivante :



2. pour vérifier si les paramètres ont bien été reçus par le bloc PID on ouvre l'éditeur pour le bloc OB35



3. on active la fonction de visualisation online et, on vérifie si les paramètres ont été transmis



Chapitre 3

Introduction :

La station PCS a été mise au point par FESTO dans le but de réaliser plusieurs applications.

Tout au long de ce chapitre, nous allons tenter d'étudier cette station tout d'abord d'un point de vue instrumentation, à savoir les différents capteurs ainsi que les actionneurs utilisés.

Ensuite, nous décrivons les différents sous systèmes que nous tenterons d'identifier et de réguler à l'aide de méthodes étudiées au cours de notre formation.

Nous finirons par décrire notre applications qui a pour but de faire la régulation des sous systèmes ainsi que la supervision de ces derniers.

3.1 Présentation du système didactique du contrôle de processus FESTO [11]:



Figure 3.1 : Maquette didactique FESTO.

Le système d'apprentissage didactique pour l'automatisation FESTO est conçu pour répondre à un certain nombre d'exigence de formation professionnelle. En effet le matériel utilisé pour la fabrication de cette station de travail didactique est identique à celui disponible dans l'industrie.

Le système d'apprentissage didactique pour l'automatisation FESTO est divisé en trois parties principales distinctes :

- L'élément de contrôle (Automate programmable industriel SIEMENS S7-313C).
- Le terminal analogique d'entrées/sorties.
- Le terminal de conversion des signaux analogique.
- L'installation contenant les différents systèmes de contrôles.

L'automate programmable S7-313C ayant été décrit dans la chapitre I, nous allons dans ce qui va suivre faire une description des éléments constituant les deux autres parties.

Le terminal analogique d'entrées/sorties permet de relier les capteurs et actionneurs à l'API à travers un câblage SYSLINK (Voir Annexe).

Avant d'être reliés au terminal analogique d'E/S, les différents capteurs et actionneur passe d'abord par le terminal de conversion des signaux analogiques. Ce dernier est constitué de plusieurs types de convertisseurs analogiques, qui permettent, d'un coté, de convertir les signaux envoyés par les différents capteurs analogiques constituant l'installation en signaux standard 0...10V et d'un autre coté de transformer les signaux analogiques envoyés par l'API vers les éléments de contrôles en signaux standard 0...24V ou 0...10V. (Pour plus de détails sur les différents convertisseurs voir l'annexe)

L'installation, quand à elle, combine quatre systèmes en boucle fermée avec différents capteurs, actionneurs numériques et analogiques :

- Système de contrôle de niveau.
- Système de contrôle de débit.
- Système de contrôle de pression.
- Système de contrôle de température.

Grâce à un automate programmable industriel, il est possible de les utiliser individuellement ou en cascade.

Les caractéristiques techniques de l'installation sont résumées dans le tableau suivant :

Paramètres	Valeurs
Pression opérationnelle maximale	50 kPa (0,5 bar)

Alimentation	24 VDC
Support	7000 × 700 × 32mm
débit maximal de la pompe	10l/min
Volume des réservoirs	12l max
canalisations flexibles	DN15 (Ø_a 15mm)
Entrées binaires	7
Sorties binaires	5
Entrées analogiques	4
Sorties analogiques	2
Nombre de réservoirs	3
Éléments de contrôle	
Pompe (0.....24 VDC)	Tension 0...10V
Vanne proportionnelle 2/2 voies	Tension 0...10V
Éléments chauffant (230 VAC, 100Watt)	ON/OFF (relai de contrôle 24VDC)
Systèmes en boucle fermée	
Système de contrôle de niveau	
Plages des valeurs du système de contrôle de niveau	3...350mm
Plages des valeurs du capteur de niveau	50...300mm
Type de signal transmis par le capteur de niveau	courant 4...20mA
Système de contrôle de débit	
Plages des valeurs du système de contrôle de débit	0...7l/min
Plages des valeurs du capteur de débit	0,3...9,0l/min
Type de signal transmis par le capteur de débit	fréquence 0...1200Hz
Système de contrôle de pression	
Plages des valeurs du système de contrôle de pression	0...30kPa (0...300mbar)
Plages des valeurs du capteur de pression	0...10kPa (0...100mbar)
Type de signal transmis par le capteur de pression	Tension 0...10V
Système de contrôle de température	
Plages des valeurs du système de contrôle de température	0...60° C
Plages des valeurs du capteur de température	-50°C...+ 150°C
Type de signal transmis par le capteur de température	Résistance PT100

Tableau 3.1 : Caractéristiques techniques de l'installation.

Le diagramme synoptique de l'installation est le suivant :

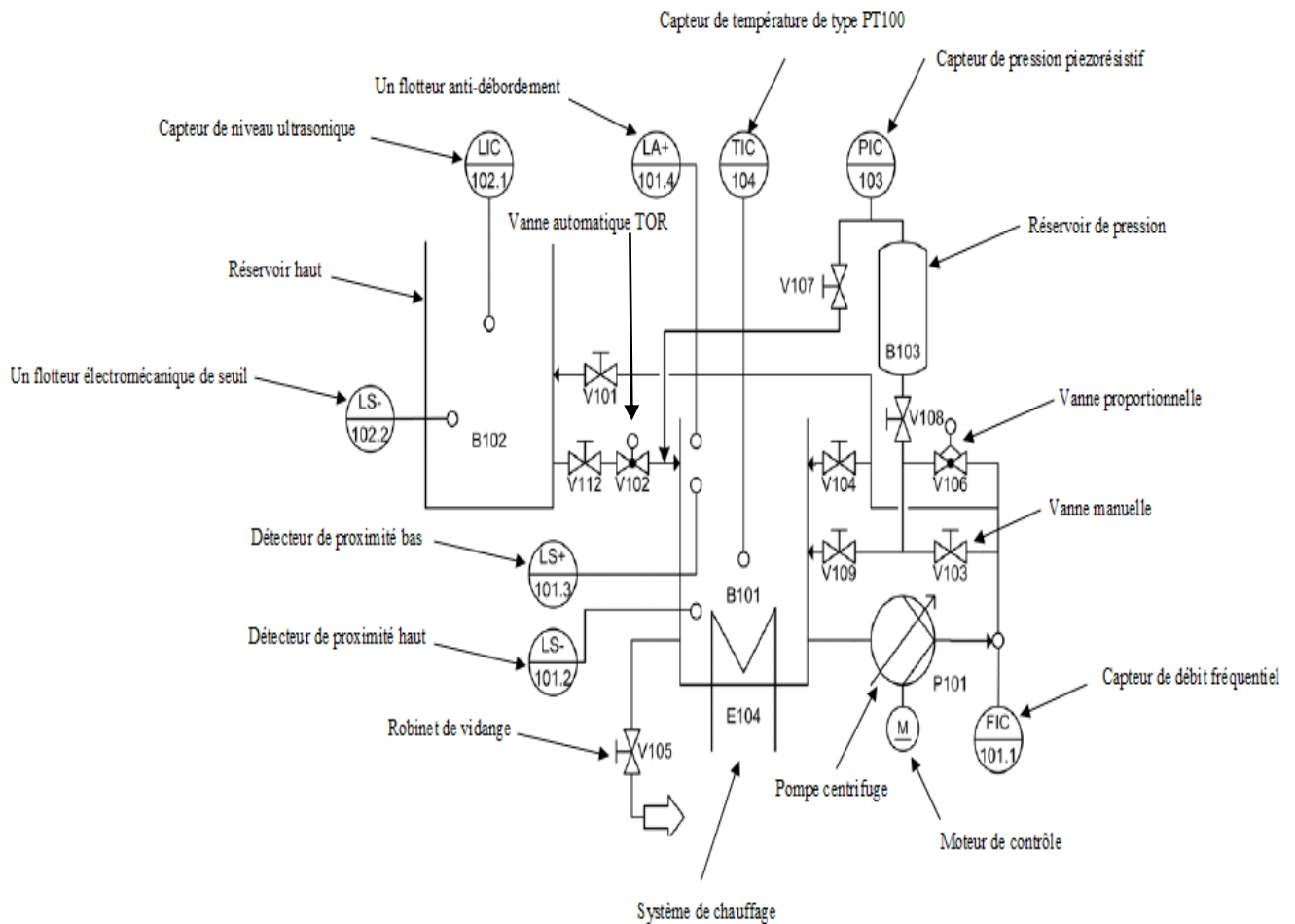


Figure 3.2 : Diagramme synoptique de l'installation didactique FESTO.

Les composants de base constituant l'installation sont les suivants :

- Capteur de niveau ultrasonique (LIC102.1).
- Capteur de débit fréquentiel (FIC101.1).
- Capteur de pression piezorésistif (PIC 103).
- Jauge de pression 0...1bar.
- Capteur de température de type PT100 (TIC104).
- Deux détecteurs de proximité capacitifs dans le réservoir du bas B101 (LS-101.2, LS+101.3).
- Un flotteur électromécanique de seuil dans le réservoir du haut (LS-102.2).
- Un flotteur anti-débordement dans le réservoir du bas (LA+101.4).
- Pompe centrifuge (P101).
- Moteur de contrôle pour la pompe (M1).

- Vanne proportionnelle avec un module de contrôle électronique (V106).
- Système de chauffage avec un relai de contrôle intégré (E104).
- Processus de vanne à bille avec un entraînement rotatif pneumatique type (SYPAR) (V102).
- Un terminal E/S.
- Un terminal pour les signaux analogiques.
- Convertisseurs de signaux : courant/tension, fréquence/tension, PT100/tension.
- Automate programmable industriel.
- Panneau de contrôle.
- Canalisations.
- Réservoir de pression (B103).
- Vannes manuelles.
- Robinet de vidange (V105).
- Unité de service pneumatique.
- Support de montage.

La fonction de chaque système en boucle fermée résulte d'une combinaison spécifique de toutes les vannes manuelles ainsi que de la programmation, configuration de l'automate/système contrôlé.

Dans ce qui va suivre nous allons voir les différentes connexions entre l'automate programmable SIMENS S7-300 CPU 313C, étudié dans le chapitre I, et les différents capteurs et actionneurs présents dans l'installation.

- Connexion API/Station : Le Plug digital XMA1 (1/A) de l'API est connecté avec la prise XMA2 du terminal E/S de la station (2) à l'aide du câble SysLink (3).

Le Plug analogique (1/C) de l'API avec X2 du terminal de conversion de signaux analogiques (6) de la station, à l'aide du câble analogique (7).

- Connexion API/Panneau de contrôle : le Plug XMG (1/B) de l'API est connecté avec le Plug X1 (4) du panneau de contrôle à l'aide du câble SysLink (5).
- Connexion API/Unité d'alimentation : Le Plug d'alimentation de l'API est connecté à l'unité d'alimentation avec un câble de sécurité bleu et rouge (4mm).

- Connexion API/PC : le PC est connecté à L'API avec un câble de programmation MPI.

Le schéma suivant montre de manière plus précise toutes ces connexions :

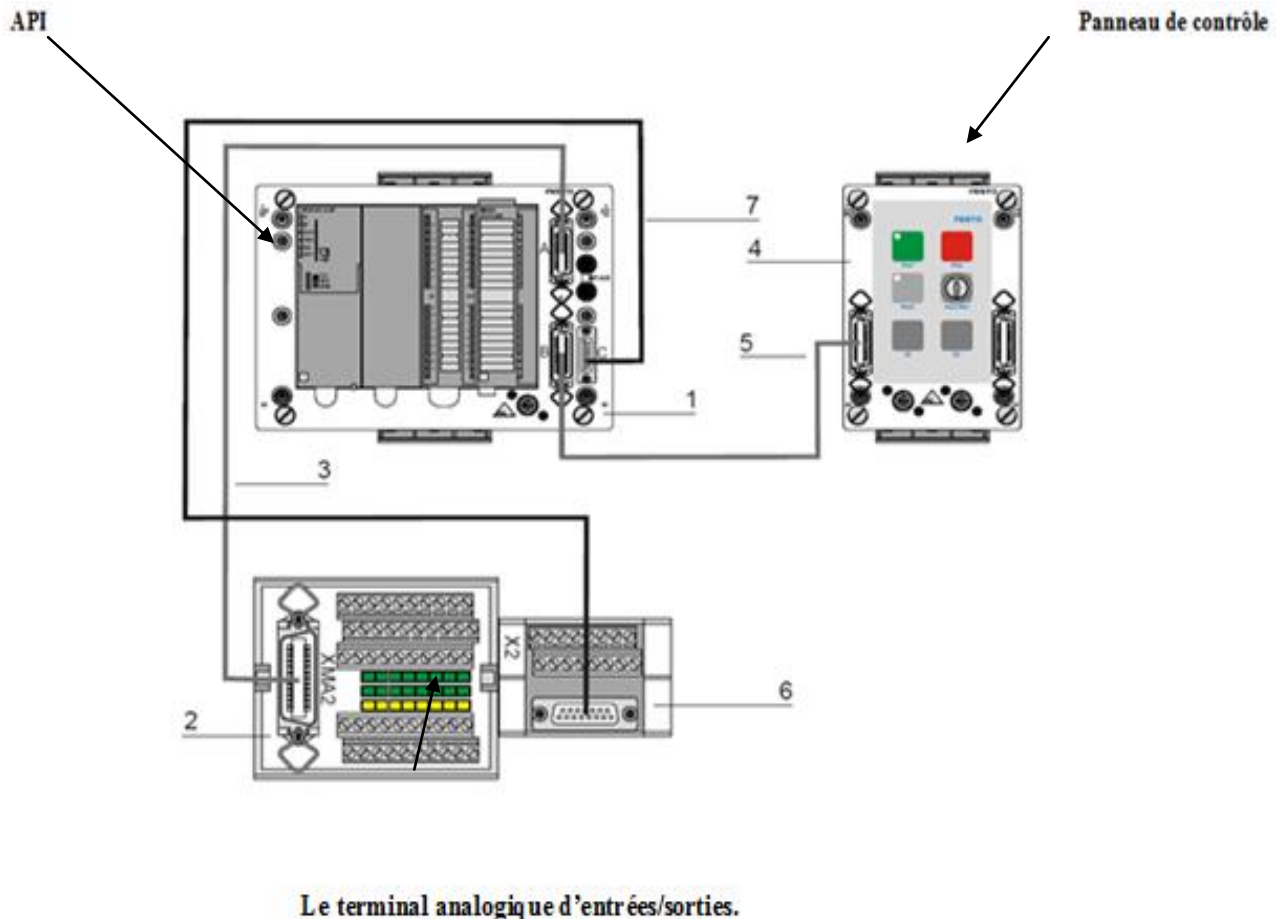


Figure 3.3 : Schéma de connexion API/Panneau de contrôle/Terminal analogique.

3.2 Design et fonction des différents systèmes constituant l'installation :

La pompe :

La pompe centrifuge P101 (1) est l'élément de contrôle (actionneur) commun à toutes les boucles de régulation. Elle permet de délivrer le fluide (eau) du réservoir bas B101 à travers les différents systèmes de canalisation.



Figure 3.4 : Pompe centrifuge.

La pompe ne doit en aucune façon fonctionner à sec. Aussi, avant tout démarrage toutes les canalisations menant à elle doivent être remplies d'eau.

La pompe est entraînée par un moteur de contrôle A4. Un relai K1 connecté comme sortie (relatif à l'API) binaire (O2 dans le XMA1) permet de choisir le mode de fonctionnement de la pompe. Si la sortie O2 est mise à 1, une sortie binaire additionnelle (O3 dans XMA1) permettra le démarrage et l'arrêt de la pompe, dans ce mode de fonctionnement la pompe fonctionnera à vitesse maximale. Si maintenant la sortie O2 est mise à 0, la sortie analogique CHANEL 0 (UA1 de X2) permettra de régler la vitesse de la pompe sous forme de tension 0/10V.

Le Processus de la vanne à bille :

Le montage V102, qui a pour rôle de contrôler l'écoulement du fluide du réservoir haut (B102) vers celui du bas (B101), représente une vanne à bille tout ou rien actionnée par un processus d'entraînement pneumatique.

Cet équipement de contrôle consiste en une vanne en cuivre (1) ouverte/fermée par un actionneur quart de tour à double effet de type SYPAR (4), contrôlé par une électrovanne 5/2 voies de type NAMUR (2,3).



Figure 3.5 : Processus de la vanne à bille.

L'indicateur de position (5) consiste en deux micro-interrupteurs électriques avec levier à roulement, ces deux derniers délivrent des signaux 24VDC (S113, S114) vers le terminal E/S XMA1.

Éléments de contrôle de niveau tout ou rien:

Trois types de surveillance de niveau sont présents dans l'installation :

- Détecteurs de proximité capacitifs. (réservoir B101)
- Flotteur de sécurité anti débordement. (réservoir B101)
- Flotteurs pour la détection des seuils de remplissage. (réservoir B101, B102)

Détecteurs de proximité capacitifs :

Les détecteurs de proximité B113 (1), B114 (2) sont fixés sur une plaque derrière la paroi externe du réservoir (B101), leur position est mécaniquement ajustable. Ils délivrent des signaux 24V et sont reliés au terminal E/S XMA1.



Figure 3.6 : Détecteurs de proximité capacitifs.

Le détecteur de proximité pour le réservoir B101, assure un remplissage minimum de manière à maintenir la résistance chauffante complètement immergée, et par suite la protéger. Celui du haut définit quand à lui le niveau maximum pour le même réservoir. Après chaque RESET du système les deux détecteurs doivent être activés.

Flotteur de sécurité anti débordement :

Le débordement du réservoir B101 est surveillé par flotteur S111. Si le niveau d'eau excède le niveau maximum, le cylindre transparent constituant le flotteur sera poussé vers le haut. Les aimants qui sont à l'intérieur de ce cylindre actionnent un contact électrique.



Figure 3.7: Flotteur de sécurité anti débordement.

Le signal binaire 24 V, généré par le flotteur S111 est transmis (normalement ouverte) au terminal E/S XMA1. Cette entrée peut activer une alarme dans le programme contenu dans l'API, et avoir un effet direct sur l'arrêt de la pompe P101 ou bien sur la fermeture de la vanne V102.

Flotteurs pour la détection des seuils de remplissage :

Le flotteur S112 (1) peut être utilisé, soit comme moyen de contrôler le niveau dans le réservoir B101, soit pour arrêter l'action chauffage de l'eau si le niveau de l'eau est critique.



Figure 3.8: Flotteur pour la détection des seuils de remplissage.

Le signal binaire 24V généré par le flotteur (normalement fermé) transmis entrée dans le terminal E/S XMA1.

la vanne proportionnelle :

La vanne proportionnelle V106 (1) est directement actionnée par une électrovalve 2/2 voies ; elle est utilisée comme élément de réglage dans les systèmes de contrôle de débit. Le principe est que le piston de la vanne est soulevé de son siège par l'intermédiaire d'un électroaimant, et libère ainsi l'écoulement à travers la vanne à un débit proportionnel à la surface ouverte.

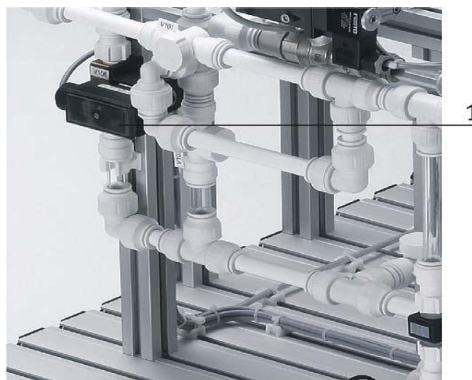


Figure 3.9 : Vanne proportionnelle.

La vanne proportionnelle est activée par une sortie binaire (O4 dans le XMA1). Un signal analogue du canal (UA2 de X2) actionne le signal d'entrée de la vanne proportionnelle avec un signal analogue standard de 0 à 10V. Ce dernier est transformé en pulsations modulées en largeur. L'ouverture de la vanne peut ainsi être ajustée à n'importe quelle position de sa plage de fonctionnement.

Le capteur de niveau ultrasonique :

La fonction du système de contrôle de niveau est de réguler le niveau de remplissage du réservoir B102.



Figure 3.10 : Capteur de niveau ultrasonique.

Le niveau du fluide dans le réservoir B102 (1) est contrôlé à l'aide d'un capteur ultrasonique (2). Le signal analogique délivré par le capteur ultrasonique sous forme de courant, passe par un convertisseur courant/tension avant d'être transmis au CHANNEL 0 du terminal E/S X2 (UE1).

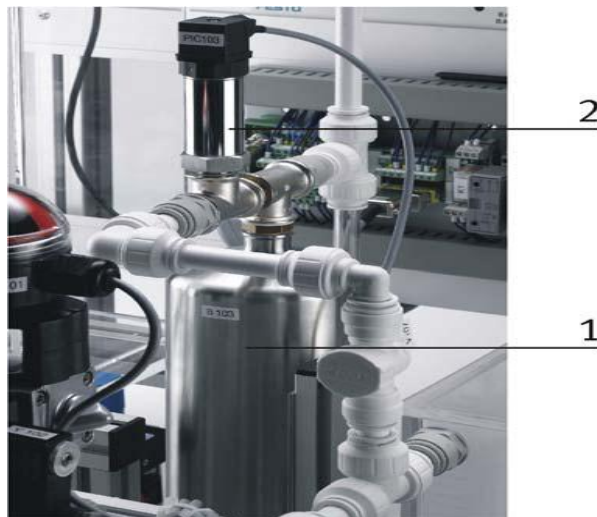
Le capteur de débit :

Le débit de l'écoulement du fluide à travers le système de canalisations est déterminé par le moyen d'un capteur optoélectronique B102 (2). Le signal carré transmis par le capteur sous forme d'une onde modulée en fréquence peut, soit être connecté directement avec une entrée binaire dans le terminal E/S XMA1, soit connecté au terminal analogique X2 (UE2) après avoir été traité par un convertisseur fréquence/tension.



Figure 3.11 : capteur de débit.**Le capteur de pression :**

La pression du fluide circulant dans le système de canalisation qui alimente le réservoir pré-rempli de gaz (air) pressurisé B103 (1), peut être mesurée à l'aide d'un capteur de pression relative piezorésistif. Le signal transmis par ce dernier est directement connecté au terminal analogique X2 (UE3).

**Figure 3.12 : Capteur de pression.****Le capteur de température :**

L'eau stockée dans le réservoir B101 et chauffée par un échangeur de chaleur E104 (3), est recirculée à travers le système de canalisation à l'aide de la pompe P101 (1). Un capteur de température de type PT100 est utilisé pour mesurer la température du système.

**Figure 3.13 : Capteur de température.**

La résistance du capteur de température est connectée à un convertisseur PT100/tension avant d'être connectée au terminal analogique X2 (UE4). Quant à l'échangeur de chaleur, il est contrôlé par une sortie binaire (O1 dans le XMA1).

La liste des cases mémoires allouées à toutes les entrées/sorties de l'instrumentation est résumée dans les trois tableaux suivants :

Pour les entrées digitales, on a la disposition suivante :

Entrées binaires (station)	Identifiant	Adresse
Capteur de débit (fréquence 0...1000Hz)	B111	E 0.0
Flotteurs de sécurité anti débordement (NF) LS+101 (réservoir B101)	S111	E 0.1
Flotteurs de surveillance des seuils de remplissage LS-102 (réservoir B102)	S112	E 0.2
Détecteur de proximité bas LO-101 (réservoir B101)	B113	E 0.3
Détecteur de proximité haut LO+101 (réservoir B101)	B114	E 0.4
Micro-interrupteur pour la vanne V102 (ouverte)	S115	E 0.5
Micro-interrupteur pour la vanne V102 (fermée)	S116	E 0.6
Entrées binaires (panneau de contrôle)		
Bouton de démarrage (NO)	S1	E 1.0
bouton de d'arrêt (No)	S2	E 1.1
Commutateur mode automatique/mode manuel	S3	E 1.2
Bouton de réinitialisation (NF)	S4	E 1.3

Tableau 3.2 : Tableau d'adressage des entrées binaires.

Pour les sorties digitales on a :

Sorties binaires (station)	Identifiant	Adresse
Ouverture du processus de vanne a bille V102	Y 102	A 0.0
Allumage de l'élément chauffant (réservoir)	E 104	A 0.1
Choix du mode de fonctionnement de la pompe =0: mode binaire/=1 : mode analogique	PompePres et	A 0.2
Pompe P101 on/off	PompeOn	A 0.3
Activation de la vanne proportionnelle V106	V106_On	A 0.4
Sorties binaires (panneau de contrôle)		
Indicateur de démarrage	H1	A 1.0

Indicateur de réinitialisation	H2	A 1.1
Indicateur Q1 (opération en boucle ouverte)	H3	A 1.2
Indicateur Q2 (opération en boucle fermée)	H4	A1.3

Tableau 3.3 : Tableau d’adressage des sorties binaires.

Et pour finir pour les entrées/sorties analogique, on a la disposition suivante :

Entrées analogiques (station)	Identifiant	Adresse
Canal d'entrée analogique 0 pour le capteur de niveau(UE1 0...10V)	AE_Niveau	PEW 256
Canal d'entrée analogique 1 pour le capteur de débit (UE2 0...10V)	AE_Débit	PEW 260
Canal d'entrée analogique 2 pour le capteur de pression (UE3 0...10V)	AE_Pression	PEW 264
Canal d'entrée analogique 3 pour le capteur de température (UE4 0...10V)	AE_Température	PEW 268
Sorties analogiques (station)		
Canal de sortie analogique pour la pompe P101	AA_Pompe	PAW 256
Canal de sortie analogique pour la vanne proportionnelle V106	AA_Vanne	PAW 260

Tableau 3.4 : Tableau d’adressage des entrées/sorties analogiques.

3.3 Les systèmes de régulation en boucle fermée :

Comme nous l’avons précisé précédemment, il est possible d’effectuer plusieurs types de régulation sur l’installation, et cela individuellement ou bien en cascade. Avant de procéder à une étude plus détaillée de chaque boucle de régulation, nous allons résumer sous forme de tableau les différents éléments entrant en jeu dans chaque boucle :

Composants	Système de contrôle de	Système de contrôle de débit avec la pompe	Système de contrôle de débit avec la vanne	Système de contrôle de pression avec la pompe	Système de contrôle de pression avec la vanne	Système de contrôle de température

	niveau	P101 comme actionneur	proportionnell e V106 comme actionneur	P101 comme actionneur	proportionnelle V106 comme actionneur	
Point de mesure	LIC102 B101	FIC101 B102		PIC103 B103		LIC104 B104
Pompe P101	Elément de contrôle	Elément de contrôle	Mode binaire on	Elément de contrôle	Elément de contrôle	Mode binaire on
Vanne proportionnelle V106	Off	Off	Elément de contrôle	Off	Elément de contrôle	Off
Elément chauffant E104	Off	Off	Off	Off	Off	Elément de contrôle
Vanne manuelle V101	Ouverte	Fermée	Fermée	Fermée	Fermée	Fermée
Vanne automatique tout ou rien V104	Ouverte/ Fermée	Fermée	Fermée	Fermée	Fermée	Fermée
Vanne manuelle V103	Fermée	Fermée	Fermée	Ouverte	Fermée	Ouverte
Vanne manuelle V104	Fermée	Ouverte	Fermée	Fermée	Fermée	Fermée
Vanne manuelle de drainage V105	Fermée	Fermée	Fermée	Fermée	Fermée	Fermée
Vanne manuelle V107	Fermée	Fermée	Fermée	Fermée	Fermée	Fermée
Vanne manuelle V108	Fermée	Fermée	Fermée	Ouverte	Ouverte	Fermée
Vanne manuelle V109	Fermée	Fermée	Ouverte	Ouverte/ Fermée	Ouverte/ Fermée	Ouverte
Vanne manuelle V112	Ouverte	Fermée	Fermée	Fermée	Fermée	Fermée

Tableau 3.5 : Tableau des configurations relatives aux différentes boucles de régulation.

Système de régulation de débit :

Dans un système de canalisation, le débit de l'écoulement doit être contrôlé, pour se faire deux types de régulation sont possibles :

- Régulation de débit par le biais de la pompe P101.
- Régulation de débit par le biais de la vanne proportionnelle V104.

Deux circuits sont alors possibles comme le montre les synoptiques suivants :

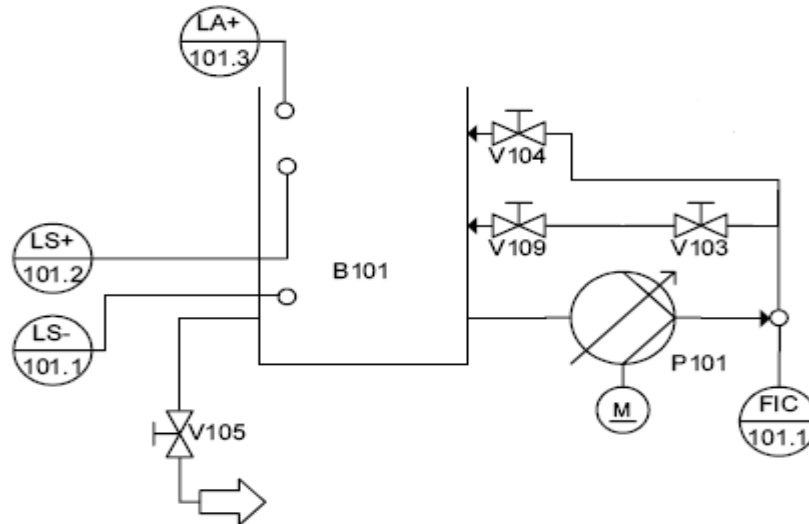


Figure 3.14 : Circuit relatif à la régulation de débit par le biais de la pompe.

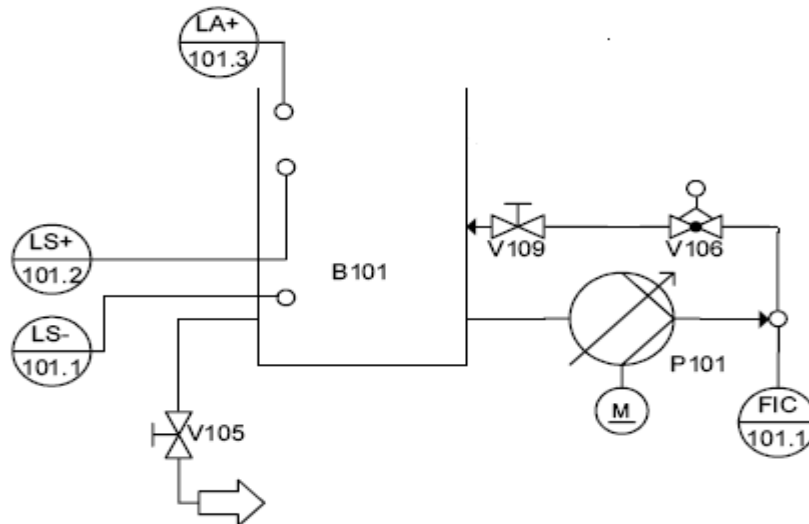


Figure 3.15 : Circuit relatif à la régulation de débit par le biais de la vanne proportionnelle.

Régulation de débit avec la pompe comme contrôleur :

Description du système pompe-débit :

Le débit du fluide délivré par la pompe à travers le système de canalisation est calculé par le biais d'un capteur de débit optoélectronique. Dans ce qui va suivre nous allons étudier le contrôle continu du débit par le moyen de la pompe.

En effet la pompe, utilisée en mode analogique, pompe le fluide du réservoir B101 à travers un système de canalisation pour le réinjecter par la suite dans ce même réservoir. Le débit de l'écoulement doit rester constant et ceci même en présence de perturbations. Ces perturbations peuvent être l'effet de l'ouverture partielle ou complète des vannes manuelles.

Le diagramme électrique, de principe, de la boucle de régulation est schématisé comme suit :

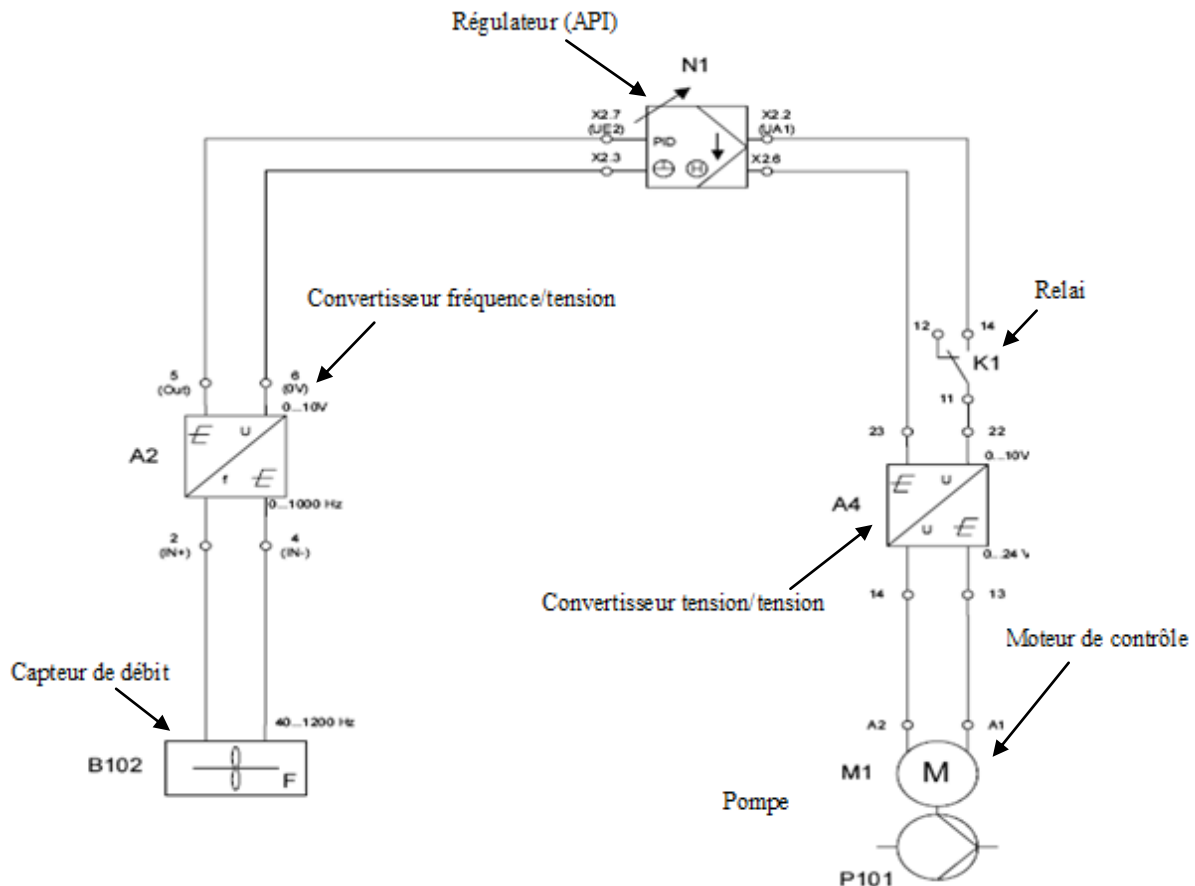


Figure 3.16 : Diagramme électrique de principe de la régulation de débit avec la pompe comme contrôleur.

La valeur actuelle du débit de l'écoulement, mesurée par le capteur optoélectronique, est transmise sous la forme d'une onde carrée (0...1000Hz), et convertie ensuite en une

tension standard (0...10V) par le convertisseur fréquence/tension (A2), avant d'être réceptionnée par l'automate programmable à travers le canal UE2 du terminal analogique X2.

La valeur de réglage (0...10V) générée par l'API est transmise au moteur de contrôle (M1) de la pompe par l'intermédiaire d'un relai K1 et d'un convertisseur tension/tension (A4). Ce dernier permettra d'adapter la tension standard (0...10V) en une tension 0...24V.

Régulation de débit avec la vanne proportionnelle comme contrôleur :

Description :

Dans ce qui va suivre nous allons étudier le contrôle continu du débit par le moyen de la vanne proportionnelle. En effet la pompe, utilisée à vitesse constante, pompe le fluide du réservoir B101 à travers un système de canalisation, passant par la vanne proportionnelle, pour le réinjecter par la suite dans ce même réservoir. La vanne proportionnelle, dont la section de passage est infiniment ajustable, permet la variation du débit.

Le diagramme électrique, de principe, de la boucle de régulation est schématisé comme suit :

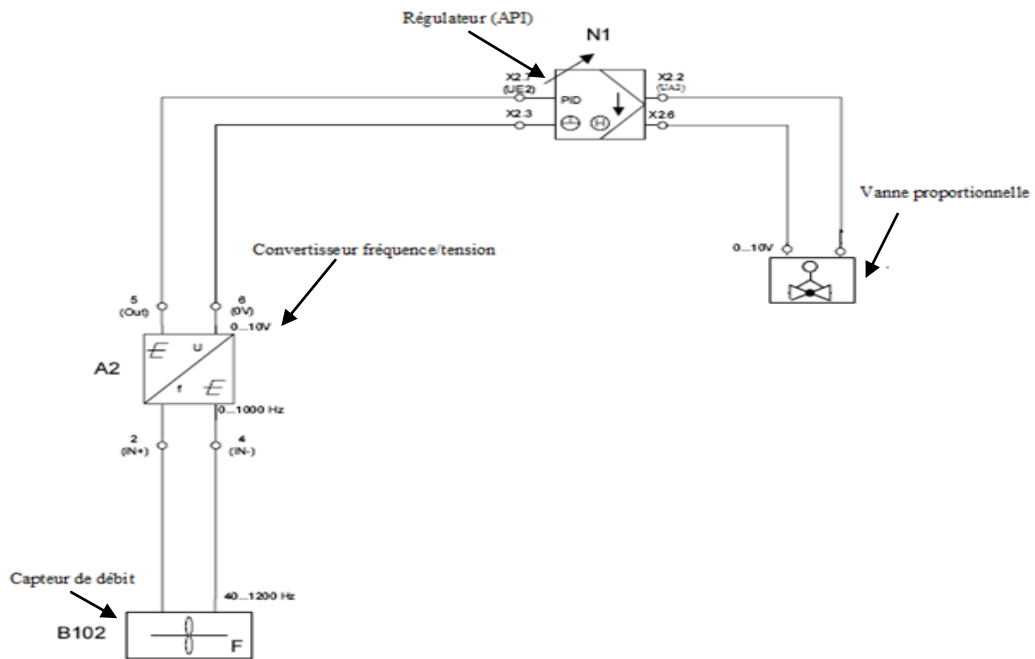


Figure 3.17 : Diagramme électrique de principe de la régulation de débit avec la vanne proportionnelle comme contrôleur.

La valeur actuelle du débit de l'écoulement, mesurée par le capteur optoélectronique, est transmise sous la forme d'une onde carrée (0...1000Hz), et convertie ensuite en une tension standard (0...10V) par le convertisseur fréquence/tension (A2), avant d'être réceptionnée par l'automate programmable à travers le canal UE2 du terminal analogique X2.

La valeur de réglage (0...10V) générée par l'API est transmise directement au contrôleur électronique de la vanne proportionnelle.

Système de régulation de niveau :

Le rôle du système de régulation de niveau est de contrôler le niveau de remplissage du réservoir B102, deux types de régulation sont alors envisageables :

- Régulation analogique avec la pompe comme élément de contrôle.
- Régulation tout ou rien avec la pompe comme élément de contrôle.

Après avoir effectué les configurations nécessaires, le diagramme synoptique de l'installation devient :

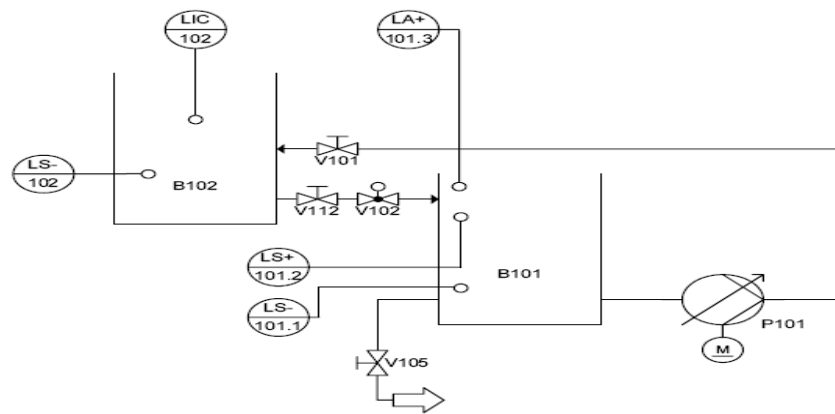


Figure 3.18 : Circuit relatif à la régulation de niveau.

Régulation de niveau analogique :

Description :

Pour cette régulation, Mis à part les informations transmises par les capteurs digitaux cités plus haut, le niveau de remplissage du réservoir B102 est contrôlé par un capteur analogique au point de mesure (LIC102).

En effet la pompe, utilisée en mode analogique, délivre l'eau stockée dans le réservoir B101 vers le réservoir B102 à travers un système de canalisation. Le niveau de remplissage du réservoir B102 doit rester constant même en présence de perturbations. Aussi la valeur de niveau transmise par le capteur analogique doit rester constante.

L'ouverture et la fermeture de la vanne automatique V102 comme moyen de drainage du réservoir B102 vers le B101 peut être considéré comme une perturbation.

Le diagramme électrique, de principe, de la boucle de régulation est schématisé comme suit :

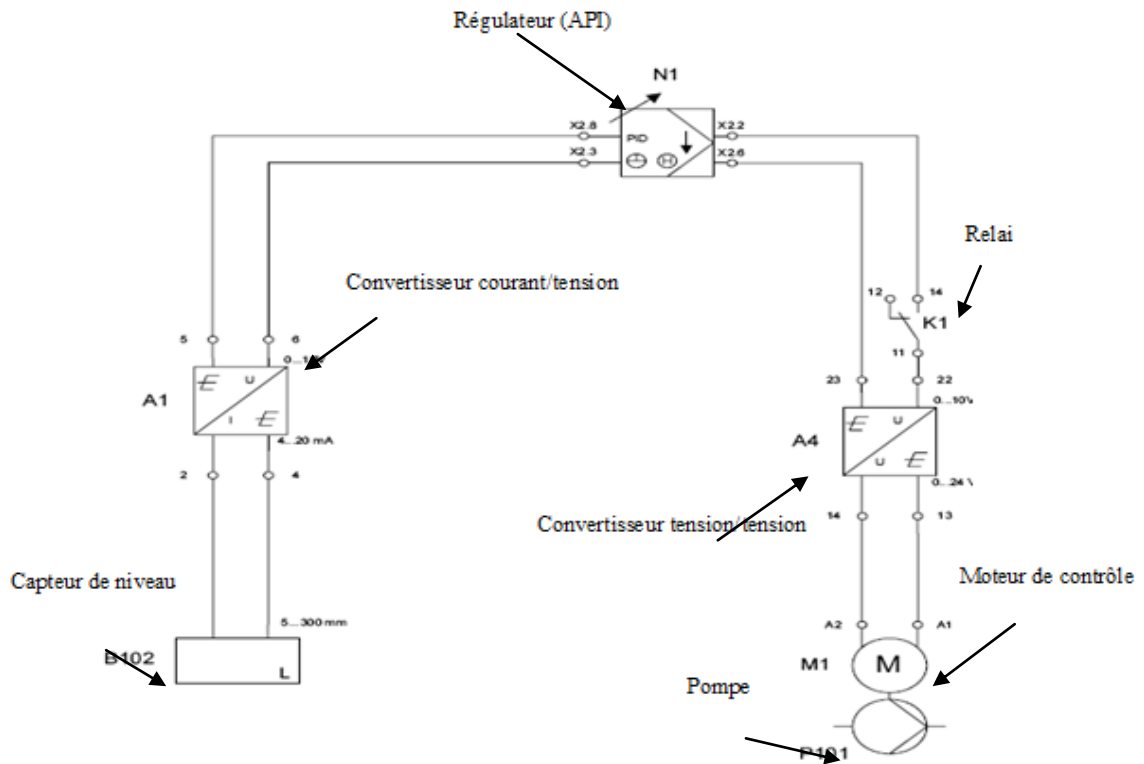


Figure 3.19 : Diagramme électrique de principe de la régulation de niveau analogique.

La valeur actuelle du niveau de remplissage, mesurée par le capteur ultrasonique, est transmise sous forme de courant (4...20mA), et convertie ensuite en une tension standard (0...10V) par le convertisseur courant/tension (A1), avant d'être réceptionnée par l'automate programmable à travers le canal 0 du terminal analogique X2.

La valeur de réglage (0...10V) générée par l'API est transmise au moteur de contrôle (M1) de la pompe par l'intermédiaire d'un relai K1 et d'un convertisseur tension/tension (A4). Ce dernier permettra d'adapter la tension standard (0...10V) en une tension 0...24V.

Système de régulation de pression :

Dans un système de canalisation, la pression de l'écoulement doit être contrôlée, pour se faire deux types de régulation sont possibles :

- Régulation de pression par le biais de la pompe P101.
- Régulation de pression par le biais de la vanne proportionnelle V104.

Deux circuits sont alors possibles comme le montre les synoptiques suivants :

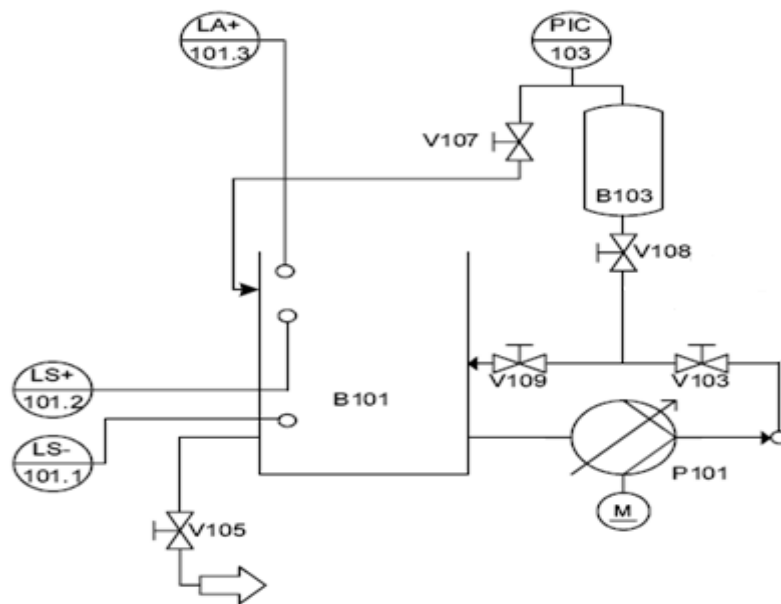


Figure 3.20 : Circuit relatif à la régulation de pression avec la pompe comme actionneur.

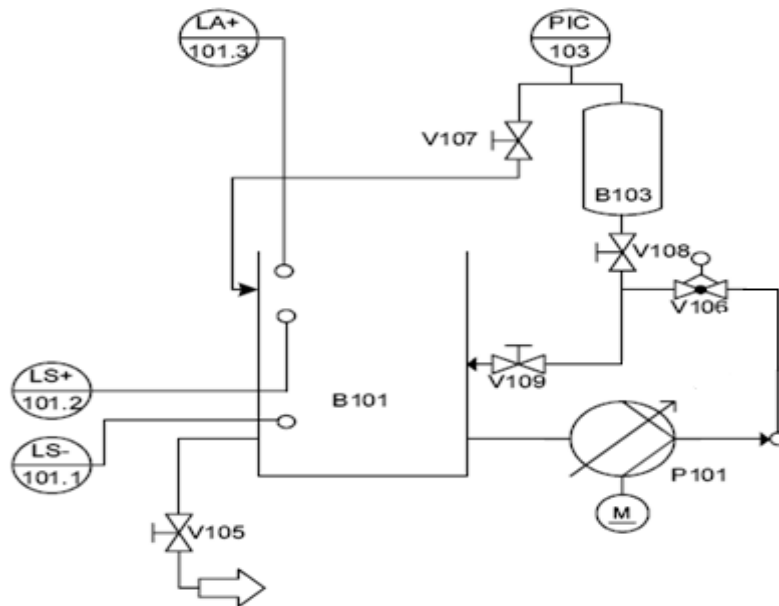


Figure 3.21 : Circuit relatif à la régulation de pression avec la pompe comme actionneur.

Régulation de pression avec la pompe comme actionneur :

Description :

La pompe délivre le fluide à travers le système de canalisation vers le réservoir de pression B103 pré rempli d'air à l'aide de la vanne manuelle V107, la pression de l'air, ainsi comprimé est calculée par un capteur de pression piézorésistif. Dans ce qui va suivre nous allons étudier le contrôle continu de la pression dans le réservoir B103 avec la pompe comme élément de contrôle.

En effet la pompe, utilisée en mode analogique, pompe le fluide du réservoir B101 à travers un système de canalisation vers le réservoir B103. La pression de l'air dans le réservoir B103 doit être maintenue constante et ceci même en présence de perturbations. Ces perturbations peuvent être l'effet de l'ouverture partielle ou complète des vannes manuelles.

Le diagramme électrique, de principe, de la boucle de régulation est schématisé comme suit :

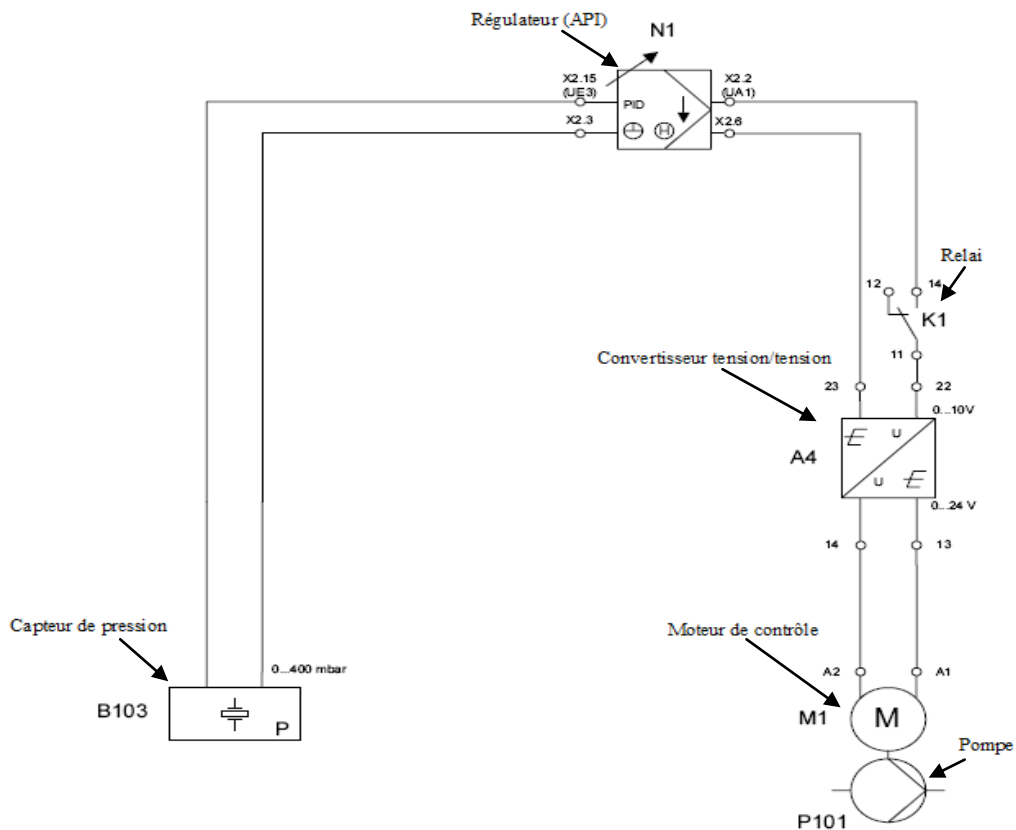


Figure 3.22 : Diagramme électrique de principe de la régulation de pression avec la pompe comme contrôleur.

La valeur actuelle de la pression de l'air comprimé dans le réservoir B103, mesurée par le capteur de pression, est directement transmise à l'API sous la forme d'une tension standard 0...10V, à travers le canal UE3 du terminal analogique X2.

La valeur de réglage (0...10V) générée par l'API est transmise au moteur de contrôle (M1) de la pompe par l'intermédiaire d'un relais K1 et d'un convertisseur tension/tension (A4). Ce dernier permettra d'adapter la tension standard (0...10V) en une tension 0...24V.

Régulation de pression avec la vanne proportionnelle comme actionneur :

Description :

Dans ce qui va suivre nous allons étudier le contrôle continu de la pression par le moyen de la vanne proportionnelle. En effet la pompe, utilisée à vitesse constante, pompe le fluide du réservoir B101 à travers un système de canalisation, passant par la vanne proportionnelle, pour l'injecter par la suite le réservoir de pression B103. La vanne proportionnelle, dont la section de passage est infiniment ajustable, permet la variation du débit et ainsi celle de la pression.

Le diagramme électrique, de principe, de la boucle de régulation est schématisé comme suit :

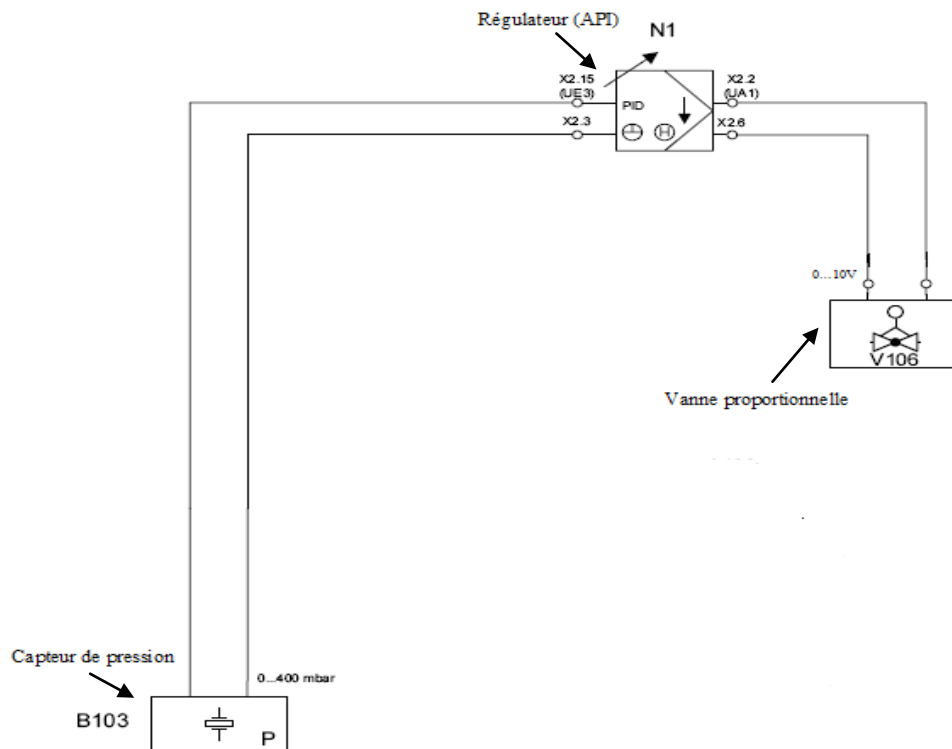


Figure 3.23 : Diagramme électrique de principe de la régulation de pression avec la vanne proportionnelle comme contrôleur.

La valeur actuelle de la pression de l'air comprimé dans le réservoir B103, mesurée par le capteur de pression, est directement transmise à l'API sous la forme d'une tension standard 0...10V, à travers le canal UE3 du terminal analogique X2.

La valeur de réglage (0...10V) générée par l'API est transmise directement au contrôleur électronique de la vanne proportionnelle.

Système de régulation de température :

Le fluide stocké dans le réservoir B101 peut être chauffé par le moyen d'un échangeur de chaleur E104 et recirculé par le moyen de la pompe (à vitesse constante). Un capteur de température de type PT100 est alors utilisé pour mesurer la température de fluide. Pour se faire il ya quelques mesures à respecter :

- La température opérationnelle du réservoir B101 ne doit pas dépasser les 65°C.
- La résistance chauffante ne doit être actionnée que si elle est complètement immergée dans le fluide.

Le circuit utilisé pour cette régulation est le suivant :

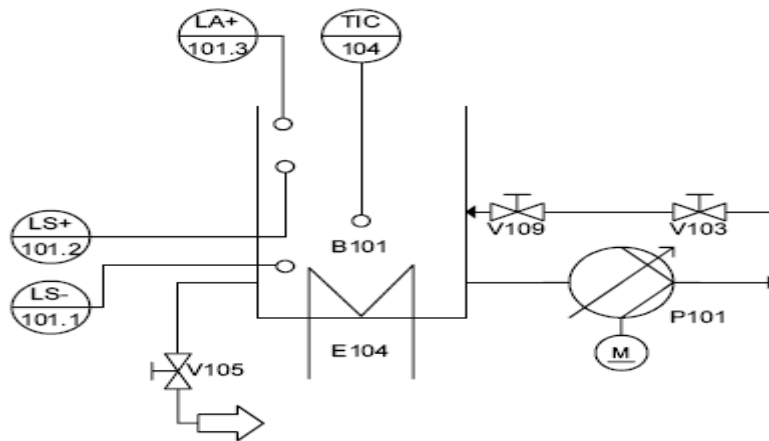


Figure 3.24 : Circuit relatif à la régulation de température.

La période d'allumage de la résistance chauffante, qui est dans ce cas la variable de contrôle, détermine la quantité de chaleur fournie, en sortie, par l'échangeur de chaleur E104. Comme perturbations on peut utiliser un fluide de température différente.

Le diagramme électrique, de principe, de la boucle de régulation est schématisé comme suit

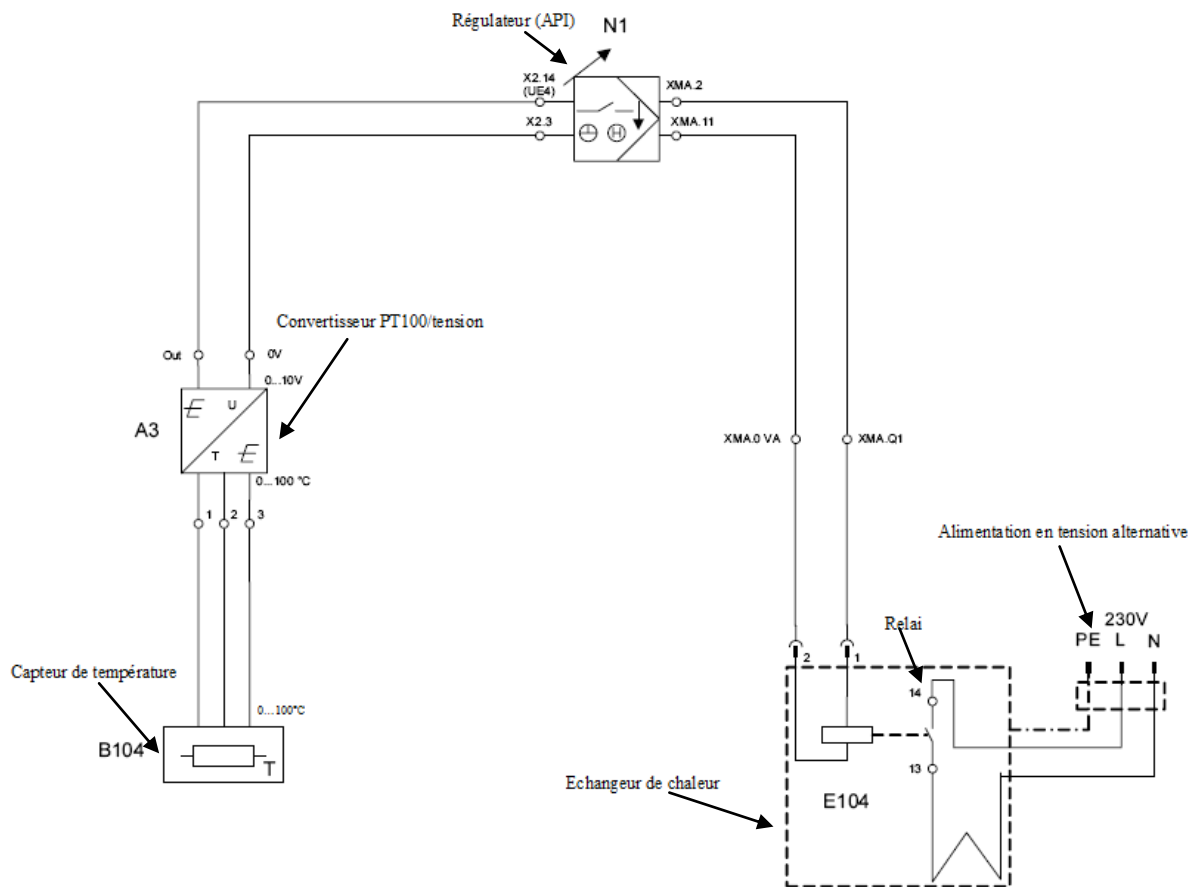


Figure 3.25 : Diagramme électrique de principe de la régulation de température.

La résistance du capteur de température PT100 est connectée au convertisseur PT100/tension A3, ce dernier change la valeur de la résistance en tension standard 0...10V et ceci à travers le canal UE4 du terminal analogique X2.

La résistance chauffante de l'échangeur de chaleur est contrôlée par l'intermédiaire d'un relai interne connecté directement à l'API à travers la sortie binaire O1 de XMA1.

Conclusion générale :

Dans notre projet on a utilisé le nouvel outil TIA PORTAL de SIEMENS de conception d'installation automatique qui comporte les trois logiciels de SIEMENS ,le SIMATIC STEP7,SIMATIC WINCC et SINAMICS StratDRIVE en un environnement commun et par conséquent il reprend à lui seul au six segment les plus important d'une installation automatique à savoir :

- Commande
- Visualisation et supervision
- Motion control
- Communication
- Diagnostique
- Safety

Notre projet nous a permis d'étudier la station didactique de contrôle des processus, FESTO, en particulier et d'avoir une vue globale sur les segments qui compose une installation automatique et comment l'étudier en utilisant l'outil d'ingénierie TIA PORTAL

Notre contribution s'est portée sur l'implémentation des régulateurs déjà synthétisé par d'anciens étudiant et la mise au point d'un système de supervision et de contrôle avec le logiciel TIA PORTAL pour servir à des fins pédagogique comme par exemple l'analyse du comportement d'un régulateur PID.

Néanmoins notre étude nous a permis de proposer deux améliorations qu'il serait intéressant de développer :

- Refaire la disposition des conduite et des actionneurs pour pouvoir faire de la régulation multi variable ;
- En jouant sur l'ouverture des vannes manuelles la fonction de transfert des boucles de régulations présentent dans la station changent complètement, donc il serait intéressant de développer un commande adaptative à base de PID.

Figure 1.1: Architecture d'un API [2].....	4
Figure 1.2: Cycle d'exécution d'un programme utilisateur [2].	5
Figure 1.3: Modules d'E/S tout ou rien (TOR) [2].	6
Figure 1.4: Module spécialisés [2].	7
Figure 1.5: langages de programmation [2].	8
Figure 1.7 : Le GRAFCET.	9
Figure 1.8 : Exemple 1 d'un programme en Ladder.....	10
Figure 1.9 : Exemple 2 d'un programme en Ladder.....	11
Figure 1.10 : Exemple d'un programme en Fonction Bloc.	11
Figure 1.11 : Exemple d'un programme en Langage Structuré.	12
Figure 1.12 : Exemple d'un programme en Liste d'Instructions.....	12
Figure 1.13: Relais électromagnétiques [2].....	13
Figure 1.14: Symbole Relais électromagnétiques [2].	14
Figure 1.15: Interrupteurs Actionnés manuellement [2].	15
Figure 1.16: Interrupteurs Actionnés mécaniquement [2].....	15
Figure 1.17: capteur inductif [2].	16
Figure 1.18: principe de fonctionnement capteur inductif [2].....	16
Figure 1.19: câblage capteur capacitifs [2].	17
Figure 1.20: capteur de température [2].	18
Figure 1.21: boucle de régulation en boucle fermée [2].....	19
Figure 1.22: régulation Tout ou Rien [2].	20
Figure 1.23: la réponse d'un régulateur de température tout ou rien [2].	21
Figure 1.24: la réponse d'un régulateur proportionnel [2].	22
Figure 2.1: TIA PORTAL [1].	26
Figure 2.2: VUE TIA PORTAL	28
Figure 2.1: VUE PROJET.....	29
Figure 2.3: STATION FESTO [3].	32
Figure 2.4 : Organigramme du choix des boucles de régulation.	33
Figure 2.5 : Organigramme de la régulation de niveau.	34
Figure 2.6 : Organigramme de la régulation de débit/pression.	35
Figure 2.7 : Organigramme de la régulation de température.	36
Figure 2.8: VUE PORTAL	39
Figure 2.9 : Schéma de principe de la régulation via un API.	53
Figure 2.10 : Schéma de principe du bloc PID.	54
Figure 2.12: HMI TIA PORTAL [1].	65
Figure 2.13: VUE PRINCIPALE.	78
Figure 2.14: VUE REGULATION NIVEAU.	80
Figure 3.1 : Maquette didactique FESTO.....	84
Figure 3.2 : Diagramme synoptique de l'installation didactique FESTO.	87
Figure 3.3 : Schéma de connexion API/Panneau de contrôle/Terminal analogique.	89
Figure 3.4 : Pompe centrifuge.	90
Figure 3.5 : Processus de la vanne à bille.....	91
Figure 3.6 : Détecteurs de proximité capacitifs.....	91
Figure 3.7: Flotteur de sécurité anti débordement.	92
Figure 3.8: Flotteur pour la détection des seuils de remplissage.....	93

Figure 3.9 : Vanne proportionnelle.....	93
Figure 3.10 : Capteur de niveau ultrasonique.	94
Figure 3.11 : capteur de débit.	95
Figure 3.12 : Capteur de pression.....	95
Figure 3.13 : Capteur de température.	95
Figure 3.14 : Circuit relatif à la régulation de débit par le biais de la pompe.	99
Figure 3.15 : Circuit relatif à la régulation de débit par le biais de la vanne proportionnelle.....	99
Figure 3.16 : Diagramme électrique de principe de la régulation de débit avec la pompe comme contrôleur.....	100
Figure 3.17 : Diagramme électrique de principe de la régulation de débit avec la vanne proportionnelle comme contrôleur.....	102
Figure 3.18 : Circuit relatif à la régulation de niveau.....	103
Figure 3.19 : Diagramme électrique de principe de la régulation de niveau analogique.	104
Figure 3.20 : Circuit relatif à la régulation de pression avec la pompe comme actionneur.	105
Figure 3.21 : Circuit relatif à la régulation de pression avec la pompe comme actionneur.	106
Figure 3.22 : Diagramme électrique de principe de la régulation de pression avec la pompe comme contrôleur.....	107
Figure 3.23 : Diagramme électrique de principe de la régulation de pression avec la vanne proportionnelle comme contrôleur.....	108
Figure 3.24 : Circuit relatif à la régulation de température.	109
Figure 3.25 : Diagramme électrique de principe de la régulation de température.....	110

Tables des matières

Introduction générale :	1
Chapitre 1	2
1.1 Introduction sur les Automates programmables :	3
1.1.1 Les éléments de base d'un API :	4
1.2 Langages de programmation :	8
1.3 Instrumentation utilisé dans la station FESTO	13
1.3.1 Relais électromagnétiques de control :	13
1.3.2 Contacteur :	14
1.3.3 Interrupteurs Actionnés manuellement :	15
1.3.4 Interrupteurs Actionnés mécaniquement :	15
1.3.5 Capteur :	16
1.4 Structure des systèmes de contrôle :	19
1.4.1 REGULATEUR TOUT OU RIEN :	20
1.4.2 Régulateur PID :	22
Chapitre 2	25
2.1 Introduction au portail TIA	26
Introduction :	26
2.2 Vues du portail TIA	28
2.3 Introduction au projet	31
2.4 Description de l'application :	33
2.5 Structure de la Tâche d'automatisation	37
2.6 Créer le projet "Festo Station" :	39
2.7 Insertion et configuration des matériels	42
2.8 Programmation de l'API	46
2.9 Visualisation du processus :	65
2.10 Test en ligne du projet	70
2.11 Tester la visualisation du processus	75
Chapitre 3	83
Introduction :	84
3.1 Présentation du système didactique du contrôle de processus FESTO [11]:	84
3.2 Design et fonction des différents systèmes constituant l'installation :	89
La pompe :	89
Le Processus de la vanne à bille :	90

Eléments de contrôle de niveau tout ou rien:	91
Détecteurs de proximité capacitifs :	91
Flotteur de sécurité anti débordement :	92
Flotteurs pour la détection des seuils de remplissage :	92
la vanne proportionnelle :	93
Le capteur de niveau ultrasonique :	94
Le capteur de débit :	94
Le capteur de pression :	95
Le capteur de température :	95
3.3 Les systèmes de régulation en boucle fermée :	97
Système de régulation de débit :	99
Système de régulation de niveau :	102
Système de régulation de pression :	105
Système de régulation de température :	109
Conclusion générale :	111

Liste des tableaux

Tableau 2.1 : Entrées du bloc PID.....	59
Tableau 2.2 : Sorties du bloc PID.....	61
Tableau 3.1 : Caractéristiques techniques de l’installation	98
Tableau 3.2 : Tableau d’adressage des entrées binaires.....	98
Tableau 3.3 : Tableau d’adressage des sorties binaires.....	99
Tableau 3.4 : Tableau d’adressage des entrées/sorties analogiques.....	99
Tableau 3.5 : configurations relatives aux différentes boucles de régulation.....	101

Bibliographie :

- [1] G.MICHEL, « Les A.P.I Architecture et application des automates programmables industriels », Edition DUNOD, 1987
- [2] Frank D .Petruzella, « Programmable Logic Controllers », McGraw-Hill, 2005
- [3] P.JARGOT, « Langages de programmation pour API, Norme IEC 61131-3 », Technique de l'ingénieur, Vol. S 8 030
- [4] SIEMENS, « PLC S7-300, CPU spécifications », SIMATIC, 2001
- [6] SIEMENS, « Logiciel de base pour S7 300/400 Régulation PID », SIMATIC, 2001
- [7] SIEMENS, « STEP 7 Professional /WinCC Advanced V11 pour l'exemple de projet "Station de remplissage " Mise en route», SIMATIC HMI, 2000
- [8] J.HELMICH, «Compact workstation manual », FESTO, 2004