

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

ECOLE NATIONALE POLYTECHNIQUE



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département Génie Electrique

Spécialité Automatique

Projet de fin d'études

Pour l'obtention du diplôme

D'Ingénieur d'Etat en Automatique

THÈME

**Programmation et supervision d'une commandes avancées
et identification en ligne avec acquisition de données pour
la station de travail FESTO**

Réalisé par :

CHAIB Ibtissam

Proposé et dirigé par :

Mr. R. ILLOUL

Mr.M.BOUKHTALA

Juin 2015

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

Remerciements

Je tien tout d'abord à exprimer mes profondes gratitudees au Dieu le Tout Puissant qui m'a aidé à réaliser ce modeste travail.

Je remercie vivement ma mère, mon père qui a contribué à la finalisation de ce travail, ainsi que tous les membres de ma famille pour leur soutien financier et moral durant tout mon cursus étudiantin.

La préparation de ce mémoire n'aurait pas été possible sans le suivi de mes promoteur : R.ILLOUL Professeur à l'ENP, et M.BOUKHATAA Professeur à l'ENP à qui je tiens à assurer également ma sincère et totale gratitude et ma profonde reconnaissance pour avoir dirigé et accordé beaucoup d'intérêt à mes travaux, car sans leurs aides précieuses ce travail n'aurait jamais pu voir le jour .Je le remercie pour leurs qualité humaine et surtout de m'avoir toujours soutenu par leurs conseils judicieux.

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

Je tiens à remercier 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 modeste travail trouvent ici l'expression de mon sincère gratitude.

Je voudrais également passé mes très vifs remerciements à mes camarades de promotion 2015.

Dédicaces

Je dédie ce modeste travail à mes très chers parents que j'aime par-dessus tout, qui par leur sacrifices, leur amour, leur patience, leur soutien et leurs encouragements m'ont permis de toujours persévérer, de viser haut et surtout de toujours donner le meilleurs de moi-même, afin de réussir et c'est le cas aujourd'hui.

Je dédie ce travail également,

A mes frères : Issam, Djallal, Mohammed, Lotfi.

A mes sœurs : Karima, Amel, Nedjma, Kalthoum, meriem.

A mes Nièces et mes neveux.

A Tous mes enseignant de l'Ecole Nationale Polytechnique

A mes amis Madjda, Nessrine, Wafa, sofia, Intissar.

A toute personne me portant de l'estime et à toute personne pour qui je porte de l'estime.

ملخص:

العمل المنجز في المذكرة يتمحور على استعمال مسير صناعي مبرمج "سيمنس" من أجل التحكم و ضبط المحطة التعليمية التجريبية المتواجدة في دائر الهندسة الالية في المدرسة الوطنية المتعددة التقنيات, حيث قمنا بإنجاز برنامج تحكم و ضبط متقدم (التحكم الترتيبي في مستوى الماء بالقيمة التدفق) كذلك قمنا ببرمجة التصور البيني للمحطة, و تسجيل المعطيات القادمة من المحطة على شكل ملف نصي.

الكلمات المفتاحية:

مسير صناعي "سيمنس", تيا بورتال, ستاب7, وينسيبي, ملتقطات و محركات المحطة فاسطو, تحكم متقدم, تسجيل المعطيات

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 Polytechnique ayant acquis une station de contrôle des processus (débit, niveau, pression et Température), FESTO, nous avons réalisé un programme de commande avancée (régulation de niveau en cascade), et un programme pour l'acquisition des données depuis l'automate.

Mots clés :

Automate programmable Siemens S7-313C, TIA PORTAL, STEP7, WinCC, capteurs et actionneurs FESTO, commande avancée, acquisition des données

ABSTRACT:

The work presented in this paper is based primarily on the use of programmable SIEMENS. Since The Department of Automation of the national polytechnic school have acquired a process control (flow, level, temperature and pressure), FESTO, we undertook a program of advanced control (cascade control level), and a program for the data acquisition from the API

Key words:

Siemens PLC S7-313C, TIA PORTAL, STEP7, WinCC, sensors and actuators FESTO, advanced control, data acquisition.

Introduction générale :

L'automate programmable industriel A.P.I ou Programmable Logic Controller PLC est un appareil électronique programmable. Il est défini suivant la norme française EN61131-1, adapté à l'environnement industriel, et réalise des fonctions d'automatisme pour assurer la commande de préactionneurs et d'actionneurs à partir d'informations logiques, analogiques ou numériques. C'est aujourd'hui le constituant essentiel des automatismes. On le trouve non seulement dans tous les secteurs de l'industrie, mais aussi dans les services et dans l'agriculture.

Dans le cadre de ce projet de fin d'étude nous avons travaillé sur l'automates siemens s7-313c qui contrôle le système d'apprentissage didactique pour l'automatisation FASTO conçu pour répondre à un certain nombre d'exigences de formation didactique, il contient quatre sous-systèmes(niveau, débit, pression, température) qui peuvent être utilisé individuellement ou en cascade. En effet le matériel utilisé pour la fabrication de cette station de travail didactique est identique à celui disponible dans l'industrie.

Dans notre système d'apprentissage didactique FESTO, un programme de régulation des différents sous-systèmes (température pression débit et niveau) et sa supervision ont été déjà faite par des boucles de régulation à contre réaction simple, et les données utilisé pour l'identification de ses sous-système sont des données graphique

Alors Le 1^{er} problème qui se pose c'est le type des données acquis depuis l'automate on veut qu'il soit sous forme de fichier de valeur, et le 2eme problème dans cette station c'est la régulation de niveau dans le réservoir qui a une dynamique lent, car ce sous-système est un system de type intégrateur

afin d'amélioré la réponse de ce sous-système (système de régulation de niveau) on nous a proposé de programmer un régulation en cascade qui présente en générale de meilleures caractéristique de stabilité qu'un commande à contre réaction simple de plus elle est moins sensible aux erreurs de modélisation , on nous a proposé aussi d'archiver les mesure des capteurs et les valeurs de consigne afin de pouvoir appliquer les différent méthodes d'identification sur ces données.

Ce travail de mémoire est divisé en trois chapitres essentiels.

Le premier chapitre contient la présentation de la station didactique FESTO et tous les instruments utilisés pour la régulation des différents sous-systèmes, ainsi la description de ces sous-systèmes.

Et dans le deuxième chapitre, nous présenterons l'historique et l'architecteur des automates programmables, ainsi les langages de programmation utilisée. Nous nous intéresserons, également à présenter les différentes gammes SIMATIC de Siemens.

Et pour le troisième chapitre, nous allons étudier le logiciel de programmation et de Supervision TIA PORTAL, et nous réalisons un programme en langage LADDER et une supervision HMI, pour piloter la station Festo, ainsi nous allons configurer notre programme pour qu'il puisse archiver les données venant depuis l'automate

Chapitre I

Présentation du système didactique du contrôle de processus « FESTO » [3]

I.1. Introduction

Le projet "Station Festo" est réalisé en tant qu'installation industrielle de régulations de processus composé de quatre boucles de régulations.

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 le système de régulation de niveau en cascade avec le débit.

I.2. Présentation du système didactique du contrôle de processus FESTO



Figure I.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 quatre 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 va être décrit dans le chapitre suivant, 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.

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 côté, de convertir les signaux envoyés par les différents capteurs analogiques constituant l'installation en signaux standard 0...10V et d'un autre côté de transformer les signaux analogiques envoyés par l'API vers les éléments de contrôles en signaux standard 0...24V ou 0...10V.

Quant à l'installation, 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
Eléments de contrôle	
Pompe (0.....24 VDC)	Tension 0...10V
Vanne proportionnelle 2/2 voies	Tension 0...10V

Eléments 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 I.1: Caractéristiques techniques de l'installation.

Le diagramme synoptique de l'installation est le suivant :

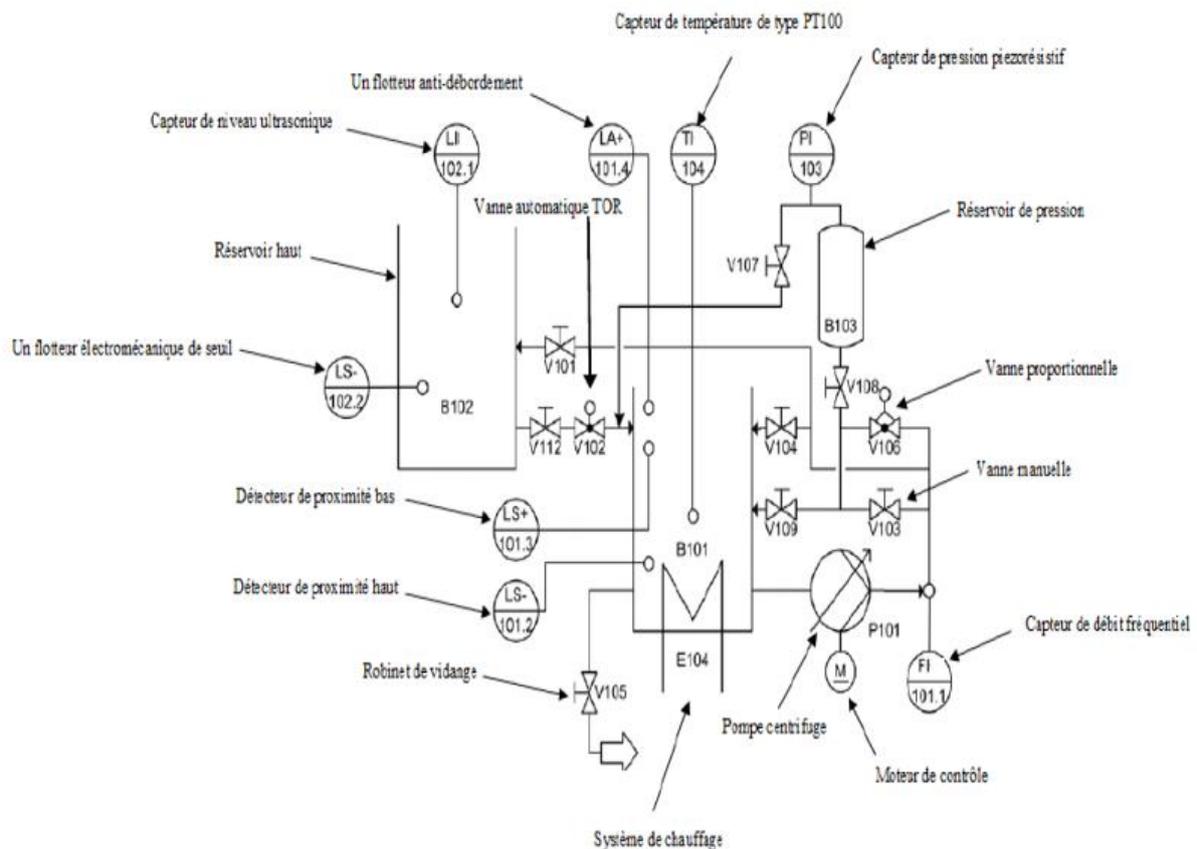


Figure I.2 : Diagramme synoptique de l'installation didactique FESTO.

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...1 bar.
- 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.

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

➤ Connexion API/Station : Le Plug digital XMA1 (1) 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) 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) 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 (voir l'annexe).

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

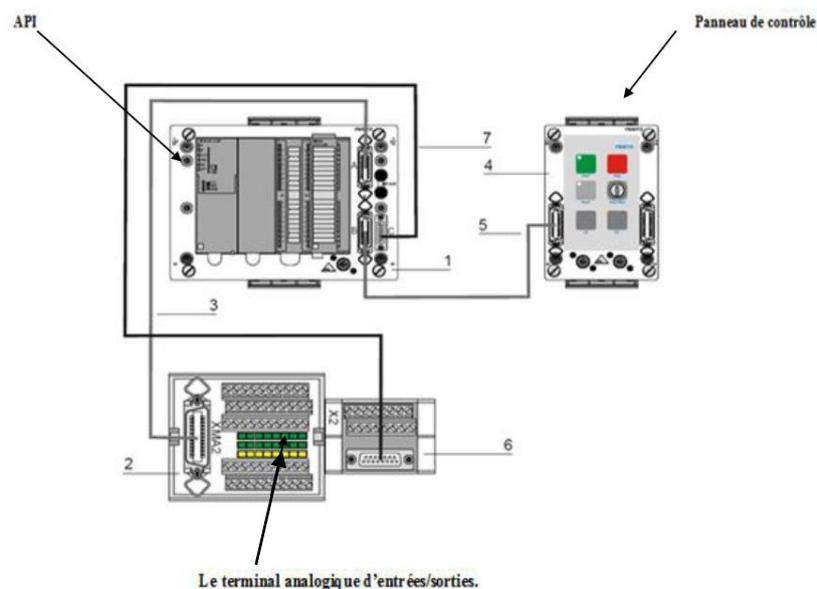


Figure I.3: Schéma de connexion API/Panneau de contrôle/Terminal analogique.

I.3. Design et fonction des différents systèmes constituant l'installation

I.3.1. La pompe

La pompe centrifuge P101 (1) est l'élément de contrôle (actionneur) commun à toutes les boucles de régulation. Elle permet de délivrer l'eau stockée dans le réservoir B101 vers le réservoir B102 à travers un système de canalisation.



Figure I.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. Un relai 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.

I.3.2. 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 I.5: 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.

I.3.3. Eléments 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 I.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 le niveau maximum pour le même réservoir.

- **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 I.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 I.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.

I.3.4. 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.

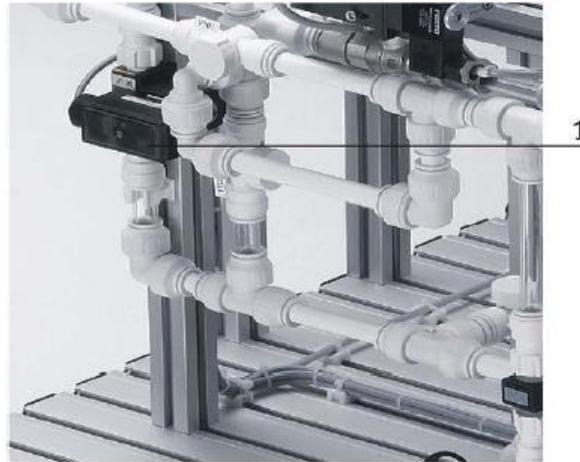


Figure I.9: Vanne proportionnelle.

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 0à10V. 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.

I.3.5. Le capteur de niveau ultrasonique :

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



Figure I.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 analogue 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).

I.3.6. 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 I.11: capteur de débit.

I.3.7. 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).

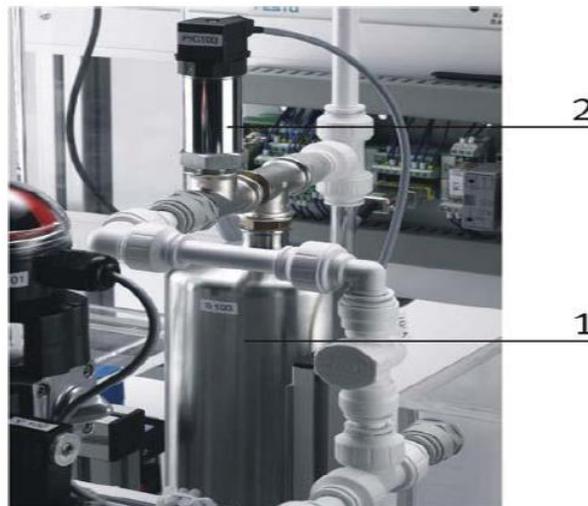


Figure I.12: Capteur de pression.

I.3.8. 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 I.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 I.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 I.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 I.4: Tableau d'adressage des entrées/sorties analogiques.

I.4. 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, nous résumons les différents éléments entrant en jeu dans chaque boucle dans le tableau suivant :

Composants	Système de contrôle de niveau	Système de contrôle de débit avec la pompe P101 comme actionneur	Système de contrôle de débit avec la vanne proportionnelle V106 comme actionneur	Système de contrôle de pression avec la pompe P101 comme actionneur	Système de contrôle de pression avec la vanne proportionnelle V106 comme actionneur	Système de contrôle de température
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	fermé	fermé	fermé	fermé	fermé
Vanne automatique tout ou rien V102	Ouvert/fermé	fermé	fermé	fermé	fermé	fermé
Vanne manuelle V103	fermé	fermé	fermé	ouverte	fermé	ouverte
Vanne manuelle V104	fermé	ouverte	fermé	fermé	fermé	Fermé
Vanne manuelle de drainage v105	fermé	fermé	fermé	fermé	fermé	Fermé
Vanne manuelle v107	fermé	fermé	fermé	fermé	fermé	fermé
Vanne manuelle v108	fermé	fermé	fermé	ouverte	ouverte	Fermé
Vanne manuelle v109	fermé	fermé	ouverte	Ouverte/fermé	Ouverte/fermé	Ouverte
Vanne manuelle v112	ouverte	fermé	fermé	fermé	fermé	fermé

Tableau I.5: Tableau des configurations relatives aux différentes boucles de régulation.

Pour la régulation de niveau en cascade avec le débit qui et le but de notre travail :

Elément de contrôle		La pompe p101	
Point de mesure		LIC102 B101	FIC101 B102
Elément chauffant E104		off	
Vanne proportionnelle V106	off	Vanne manuelle V103	Fermée
Vanne manuelle V112	Ouverte	Vanne manuelle V104	fermée
Vanne manuelle V101 ouverte	Ouverte	Vanne manuelle de drainage V105	Fermée
Vanne automatique TOR V102	Ouverte/fermée	Vanne manuelle V107	Fermée
Vanne manuelle V108	Fermée	Vanne manuelle v109	fermée

Tableau I.6: Tableau des configurations relatives à la boucle de régulation en cascade.

I.5. Système de régulation de niveau en cascade avec le débit :

Le rôle du système de régulation de niveau en cascade avec le débit est de contrôler le niveau de remplissage du réservoir B102 par la régulation de débit analogique avec la pompe comme élément de contrôle. pour permettre à la régulation de niveau qui a une dynamique lente d’avoir une réponse rapide face aux perturbations extérieures ainsi qu’aux changements de consigne.

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

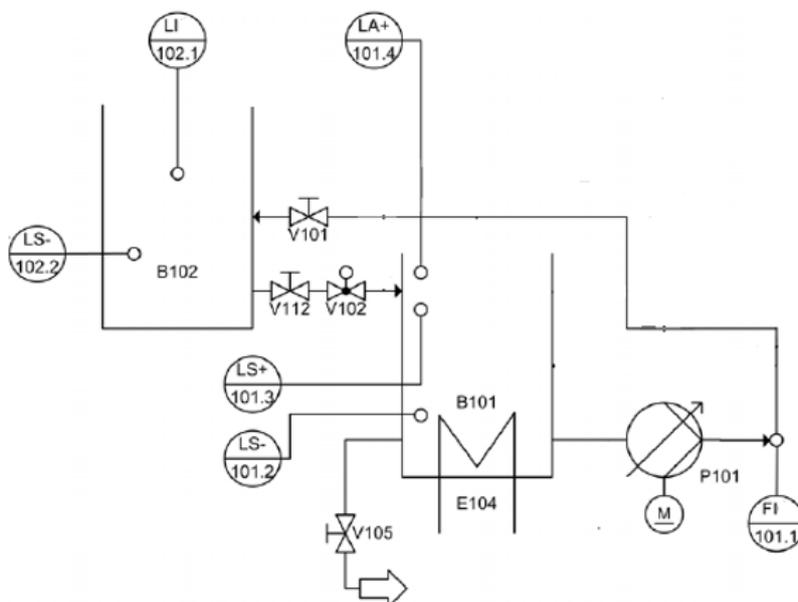


Figure I.14: Circuit relatif à la régulation de niveau en cascade avec le débit

- **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 mesuré par un capteur analogique au point de mesure (LIC102). Et 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 (FIC101).

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.

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 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.

I.6. Conclusion

Dans ce chapitre nous avons vu les différentes boucles de régulations dans la station FESTO et on a vu les différents instruments utilisés et la configuration utilisée pour la boucle de régulation de niveau en cascade afin de pouvoir élaborer un programme de régulation et la supervision de ce système, qui est l'objet de notre étude

Chapitre II

Généralité sur les automates programmables

II.1. Introduction

L'automate Programmable Industriel, API (en anglais Programmable Logic Controller, PLC) est un type particulier d'ordinateur, robuste et réactif, ayant des entrées et des sorties physiques, utilisé pour automatiser des processus comme la commande des machines sur une ligne de montage dans une usine, ou le pilotage de systèmes de manutention automatique. [2]

Dans ce chapitre, nous présenterons l'historique et l'architecteur de ces outils puissants ainsi les langages de programmation utilisée. Nous nous intéresserons, également à présenter les différentes gammes SIMATIC de Siemens.

II.2. Généralité sur les automates

II.2.1. Historique

Les automatismes séquentiels ont été réalisés, depuis longtemps, à base de relais électromagnétiques. L'inconvénient c'est qu'il s'agit d'un système câblée ce qui impose la refonte complète du câblage et ceci pour la moindre modification dans l'ordonnancement des séquences. En 1966, l'apparition des relais statiques a permis de réaliser des divers modules supplémentaires tel que le comptage, la temporisation, le pas à pas ... Cependant cette technologie avait le même problème : technologie câblée.

En 1968 et à la demande de l'industrie automobile nord-américaine, sont apparus les premiers dispositifs de commande logique aisément modifiable : Les PLC (Programmable Logic Controller) par *Allen Bradley, Modicom* [4]

II.2.2. API – Définition

Norme NFC 63-850 : « Appareil électronique qui comporte une mémoire programmable par un utilisateur automaticien à l'aide d'un langage adapté, pour le stockage interne des instructions composant les fonctions d'automatisme comme par exemple: logique séquentiel et combinatoire, temporisation, comptage, décomptage, comparaison ; calcul arithmétique ; réglage, asservissement, régulation, etc. pour commander, mesurer et contrôler au moyen de modules d'entrées et de sorties (logiques, numériques ou analogiques) différentes sortes de machines ou de processus, en environnement industriel . »

C'est donc une machine électronique qui se place entre deux grands courants : la logique câblée et le calculateur universel. Elle se distingue par plusieurs caractéristiques : conçu pour fonctionner dans des ambiances industrielles qui peuvent être sévères.[4]

II.3. Architecture des automates [1]

II.3.1. Aspect extérieur :

Les automates peuvent être de type compact ou modulaire.

De type compact, on distinguera les modules de programmation (LOGO de Siemens, ZELIO de Schneider, MILLENIUM de Crouzet ...) des microautomates. Il intègre le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage rapide, E/S analogiques ...) et recevoir des extensions en nombre limité. Ces automates, de fonctionnement simple, sont généralement destinés à la commande de petits automatismes.

De type modulaire, le processeur, l'alimentation et les interfaces d'entrées / sorties résident dans des unités séparées (modules) et sont fixées sur un ou plusieurs racks contenant le "fond de

panier" (bus plus connecteurs). Ces automates sont intégrés dans les automatismes complexes où puissance, capacité de traitement et flexibilité sont nécessaires.

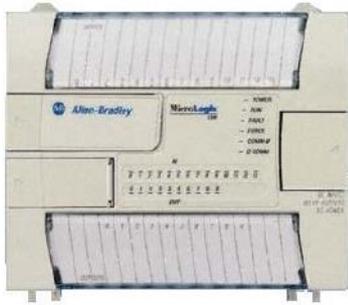


Figure II.1: Automate compact



Figure II.2: Automate modulaire (Modicon)

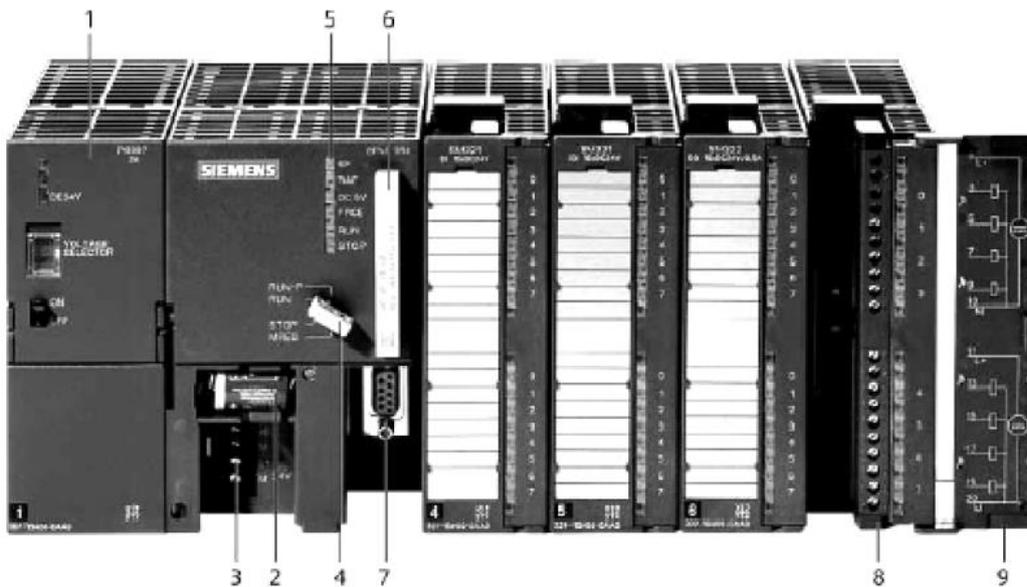


Figure II.3: Automate modulaire (Siemens)

- | | |
|---|------------------------------|
| 1 Module d'alimentation | 6 Carte mémoire |
| 2 Pile de sauvegarde | 7 Interface multipoint (MPI) |
| 3 Connexion au 24V cc | 8 Connecteur frontal |
| 4 Commutateur de mode (à clé) | 9 Volet en face avant |
| 5 LED de signalisation d'état et de défauts | |

II.3.2. Structure interne

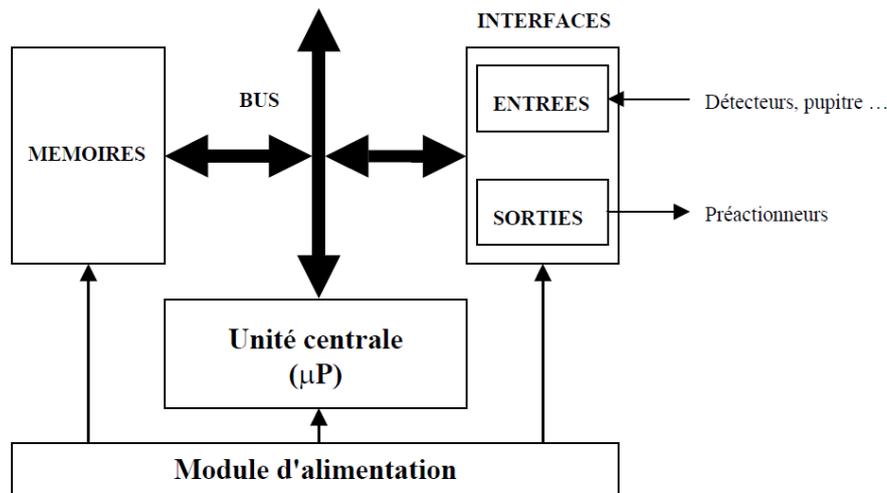


Figure II.4: La structure interne d'un automate [1]

- **Module d'alimentation** : il assure la distribution d'énergie aux différents modules.
- **Unité centrale** : à base de microprocesseur, elle réalise toutes les fonctions logiques, arithmétiques et de traitement numérique (transfert, comptage, temporisation ...).
- **Le bus interne** : il permet la communication de l'ensemble des blocs de l'automate et des éventuelles extensions.
- **Mémoires** : Elles permettent de stocker le système d'exploitation (ROM ou PROM), le programme (EEPROM) et les données système lors du fonctionnement (RAM). Cette dernière est généralement secourue par pile ou batterie. On peut, en règle générale, augmenter la capacité mémoire par adjonction de barrettes mémoires type PCMCIA.
- **Interfaces d'entrées / sorties** :
 - Interface d'entrée : elle permet de recevoir les informations du processus. ou du pupitre
 - Interface de sortie : elle permet de commander les divers préactionneurs et éléments de signalisation du processus. tout en assurant l'isolement électrique. [1]

II.4. Fonctions réalisées

Les automates compacts permettent de commander des sorties en T.O.R et gèrent parfois de fonctions de comptage et de traitement analogique. Les automates modulaires permettent de réaliser de nombreuses autres fonctions grâce à des modules intelligents que l'on dispose sur un ou plusieurs racks. Ces modules ont l'avantage de ne pas surcharger le travail de la CPU car ils disposent bien souvent de leur propre processeur.

➤ Principales fonctions

- **Cartes d'entrées / sorties**
 Au nombre de 4, 8, 16 ou 32, elles peuvent aussi bien réaliser des fonctions d'entrées, de sorties ou les deux.
 Ce sont les plus utilisées et les tensions disponibles sont normalisées (10, 24, 48, 110 ou 230V continu ou alternatif ...).

Les voies peuvent être indépendantes ou posséder des "communs". Les cartes d'entrées permettent de recueillir l'information des capteurs, boutons ... qui lui sont raccordés et de la matérialiser par un bit image de l'état du capteur.

Les cartes de sorties offrent deux types de technologies : les sorties à relais électromagnétiques (Bobine plus contact) et les sorties statiques (à base de transistors ou de triacs).

- **Cartes de comptage rapide** elles permettent d'acquérir des informations de fréquences élevées incompatibles avec le temps de traitement de l'automate.
Exemple : signal issu d'un codeur de position.
- **Cartes de commande d'axe** Elles permettent d'assurer le positionnement avec précision d'élément mécanique selon un ou plusieurs axes. La carte permet par exemple de piloter un servomoteur et de recevoir les informations de positionnement par un codeur.
L'asservissement de position pouvant être réalisé en boucle fermée.
- **Cartes d'entrées / sorties analogiques** Elles permettent de réaliser l'acquisition d'un signal analogique et sa conversion numérique (CAN) indispensable pour assurer un traitement par le microprocesseur.
La fonction inverse (sortie analogique) est également réalisée.
Les grandeurs analogiques sont normalisées : 0-10V ou 4-20mA.

Autres cartes :

- Cartes de régulation PID
- Cartes de pesage
- Cartes de communication (Ethernet ...)
- Cartes d'entrées / sorties déportées [3]

II.5. Langages de programmation

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 [6]:

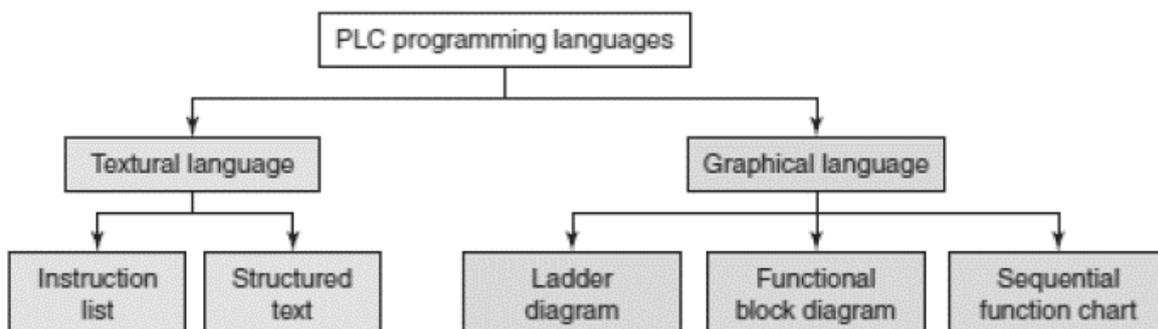


Figure II.5: langages de programmation [5]

- **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

II.5.1. Les langages graphiques

1) 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 :

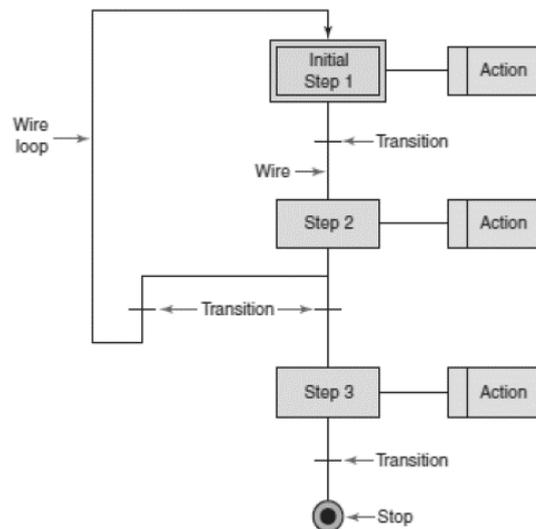


Figure II.6: Le GRAFCET

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.

2) 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:

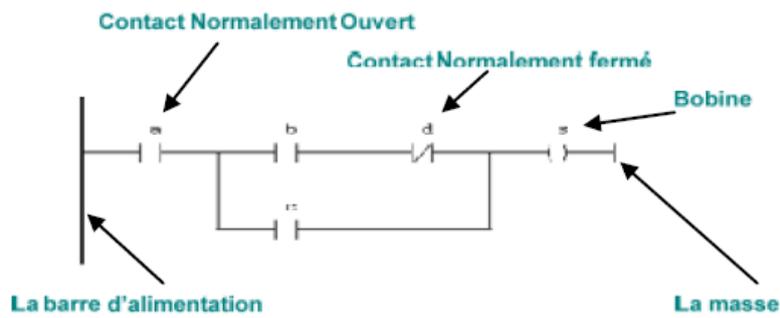


Figure II.7: Exemple 1 d'un programme en Ladder.

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.

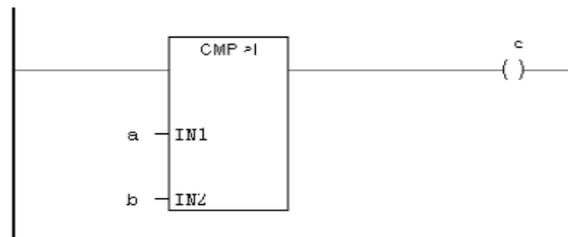


Figure II.8: Exemple 2 d'un programme en Ladder.

3) 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 II.9: Exemple d'un programme en Fonction Bloc.

II.5.2. Les langages textuels

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

Figure II.10: Exemple d'un programme en Langage Structuré.

2) 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 II.11: Exemple d'un programme en Liste d'Instructions. [7]

II.6. Présentation de la gamme SIMATIC de SIEMENS [1]

Siemens reste le seul à proposer une gamme complète de produits pour l'automatisation industrielle, par le biais de sa gamme SIMATIC. L'intégration globale de tout l'environnement d'automatisation est réalisée grâce à :

- Une configuration et une programmation homogène des différentes unités du système.
- Une gestion cohérente des données.
- Une communication globale entre tous les équipements d'automatisme mis en œuvre.

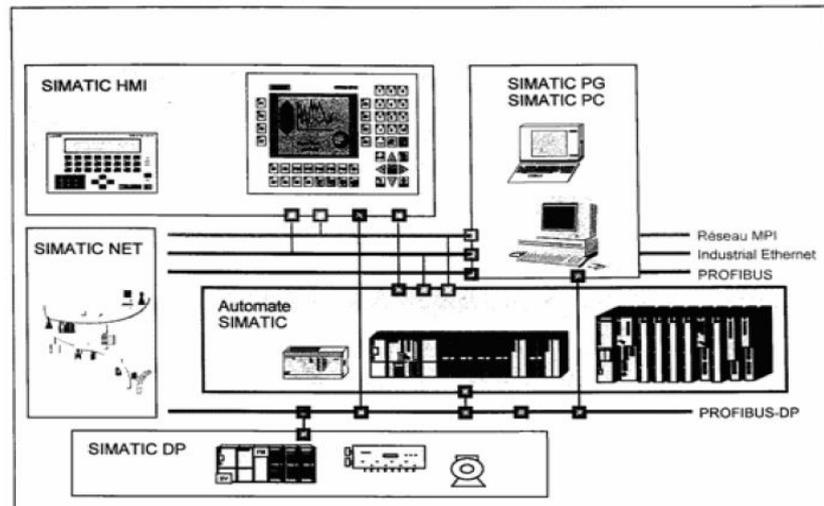


Figure II.12: Présentation de la gamme de SIMATIC

II.6.1. Les différentes variantes dans la gamme SIMATIC

1. SIMATIC S7

Cette gamme d'automates comporte trois familles :

- S7 200, qui est un Micro-automate modulaire pour les applications simples, avec possibilité d'extensions jusqu'à 7 modules, et une mise en réseau par l'interface multipoint (MPI) ou PROFIBUS.

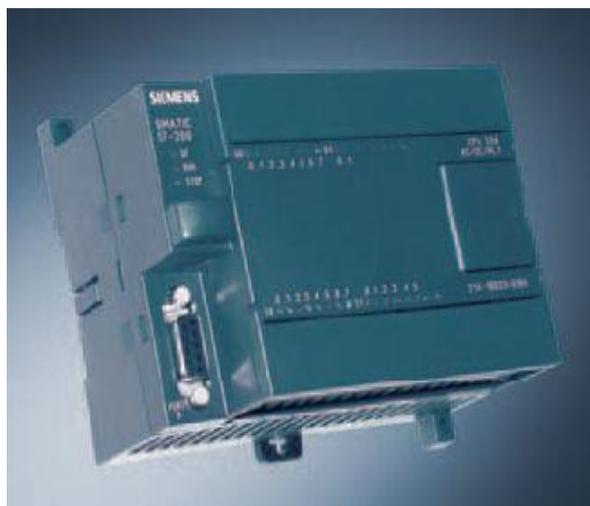


Figure II.13: L'API S200

- S7300 est un Mini-automate modulaire pour les applications d'entrée et de milieu de gamme, avec possibilité d'extensions jusqu'à 32 modules, et une mise en réseau par l'interface multipoint (MPI), PROFIBUS et Industriel Ethernet



Figure II.14: L'API S300

- S7 400 est un automate de haute performance pour les applications de milieu et haut de gamme, avec possibilité d'extension a plus de 300 modules, et une possibilité de mise en réseau par l'interface multipoint (MPI), PROFIBUS ou Industriel Ethernet.



Figure II.15: L'API S400

2. SIMATIC C7

Le SIMATIC C7 combine automate programmable et panneau operateur dans une seule unité L'automate compte la CPU, les modules d'entrées/sorties, et le panneau operateur qui est utilisé comme une interface Homme/Machine HMI.

Le C7 permet la visualisation des états de fonctionnement, des valeurs actuelles du Processus et des anomalies.



Figure II.16: La gamme SIMATIC C7.

3. SIMATIC M7

Les SIMATIC M7 sont des calculateurs industriels compatibles PC. Il s'agit d'un système modulaire sous boîtier, construit dans la technique des automates SIMATIC S7. Il peut être intégré dans un automate S7 300/400 ou être utilisé comme système autonome avec une périphérie choisie dans la gamme S7.

Le M7 300/400 est capable d'effectuer simultanément avec une seule CPU des opérations en temps réel, par exemple des algorithmes complexes de commande, de régulation ainsi que des tâches de visualisation et de traitement informatique. Les logiciels sous DOS ou Windows sont exploitables sur le M7-300. Par ailleurs, avec son architecture normalisée PC, il permet une extension programmable et ouverte de la plate-forme d'automatisation S7.



Figure II.17: La gamme SIMATIC M7.

II.6.2. Description de l'automate programmable S7-313



Figure II.18: L'API S7-313C.

L'automate programmable utilisé est le S7-313C, c'est un automate compact constitué des éléments suivant :

- CPU 313C.
- Module E/S digitales intégré.
- Module E/S analogique intégré.

Remarque : l'automate programmable compact SIEMENS ne possède pas d'alimentation intégrée ; il est donc nécessaire d'alimenter le reste des composants par une source externe. Dans ce qui va suivre nous nous proposons de décrire à travers des tableaux les caractéristiques de ces différents composants.

II.6.2.1. La CPU

Sur la plupart des CPU 300, on peut remarquer les éléments suivants :

- Un commutateur de fonctionnement.
- Des LEDs de visualisation.
- Une pile.
- Une carte mémoire.
- Une interface MPI.

En ce qui concerne la CPU 313C le tableau récapitulatif des différentes positions du commutateur du mode de fonctionnement est le suivant :

Position	Signification	Explications
RUN	Mode de marche	La CPU traite le programme utilisateur. Ce dernier peut être modifié.
STOP	Mode d'arrêt	La CPU ne traite aucun programme utilisateur.
MRES	Effacement général	Position instable du commutateur pour effacement général de la CPU

Tableau II.1 : Position du commutateur du mode de fonctionnement.

Le tableau récapitulatif des LEDs de visualisation est le suivant :

LED	Signification
SF (rouge)	Défaut de matériel ou logiciel
DC5V (verte)	L'alimentation 5V CC et bus S7-300 est correcte
FRCE (jaune)	Le forçage permanent est actif
RUN (verte)	CPU en RUN ; la LED clignote à 1 HZ au démarrage ; à 0.6 HZ en mode d'attente
STOP (jaune)	CPU en STOP ou ATTENTE ou en démarrage ; la LED clignote en cas d'effacement général

Tableau II.2: Les LEDs de visualisations.

Quant aux éléments : pile, carte mémoire, interface MPI seront décrit dans le tableau suivant, avec les caractéristiques techniques de la CPU :

CPU et version	
N° de référence	6ES7 313-5BF03-0AB0
Version matériel	1
Version microprogramme	V2.6
Pack de programmation correspondant	STEP 7 à partir de V5.4 + SP3 ou STEP 7 à partir de V5.3 + SP2 avec HSP 0123
Mémoire	
Mémoire de travail intégrée	64 Ko
Mémoire de travail extensible	Non
Mémoire e chargement	Enfichable via microcarte mémoire (max 8 Mo)
Conservation des données sur la microcarte mémoire (après la dernière programmation)	Minimum 10 ans
Sauvegarde	Garantie par la microcarte mémoire

Temps de traitement (μs)	
Opérations sur bit	Min. 0.1
Opérations sur mots	Min. 0.2
Opérations arithmétiques sur nombres entiers	Min. 2
Opérations arithmétiques sur nombres réels	Min. 3
Compteurs	
Compteurs S7 :	
Nombre	256
Plage de comptage	0 à 999
Rémanence	Réglable
Rémanence par défaut	De Z0 à Z7
Compteurs CEI :	
Type	SFB
Nombre	Illimité (suivant la mémoire de travail)
Temporisations	
Temporisation S7 :	
Nombre de temporisations	256
Plage de temps	10 ms à 9990 s
Rémanence	Réglable
Rémanence par défaut	Pas de rémanence
Temporisation CEI :	
Type	SFB
Nombre	Illimité (suivant la mémoire de travail)

Zones de données	
Mémentos :	
Taille	256 octets
Rémanence réglable	De MB 0 à MB 71
Par défaut	De MB 0 à MB 15
Mémentos de cadence	8 bits (1 octet de memento)
Blocs de données :	
Nombre	511 : de DB1 à DB551
Taille	Max. 16Ko
Blocs	
Nombre total	1024 (DB, FB, DB)
OB :	
Taille	Max. 16Ko
cycle libre	OB1
alarmes horaires	OB10
alarmes temporisées	OB20
alarmes cycliques	OB35
alarmes processus	OB40
alarme de démarrage	OB100
alarmes d'erreur asynchrones	OB80, 82, 85, 87
alarmes d'erreur asynchrones	OB121, 122
FB :	
Nombre	1024 : de FB1 à FB1024
Taille	Max. 16Ko
FC :	
Nombre	1024 : de FB1 à FB1024
Taille	Max. 16 Ko

Zones d'adresses (E/S)	
Périphérie totale :	
Entrées	1024 octets (adressage libre)
Sorties	1024 octets (adressage libre)
Mémoire image des E/S :	
Entrées	128 octets
Sorties	128 octets
Voies TOR :	
Voies intégrées d'entrées	24 voies intégrées
Voies intégrées de sorties	16 voies intégrées
Entrées	1016
Sorties	1008
Voies analogiques :	
Voies intégrées d'entrées	4+une entrées résistance
Voies intégrées de sorties	2
Entrées	253
Sorties	250
Interface MPI	
Services :	
Communication PG/PC(OP)	Oui
Routage	Non
Communication par données globales	Oui
Communication de base S7	Oui en tant que client
Vitesse de transmission	187.7 kBauds max
Programmation	
Langage de programmation	CONT, LIST, LOG
Niveaux de parenthèses	8
Fonctions système (SFC)	Oui
Blocs fonctionnels système (SFB)	Oui (dont la fonction PID)
Protection du programme utilisateur	Oui
E/S intégrées	

Adresses par défaut des :	
Entrées TOR	De 124.0 à 126.7
Sorties TOR	De 124.0 à 125.7
Entrées analogiques	De 752 à 761
Sorties analogiques	De 752 à 755
Dimensions	
montage L x H x P (mm)	120 x 125 x 130
Poids	660g
Tensions, courants	
Tension d'alimentation (valeur nominale)	24VDC
Plage admissible	De 20.4 à 28.8VDC
Courant absorbé (en marche à vide)	150 mA
Courant d'appel à l'enclenchement	11 A
Courant absorbé (valeur nominale)	700mA
Protection externe des conducteurs de l'alimentation (conseillée)	Commutateur LS, type C min. 2 A, Commutateur LS, type B min. 4 A
Puissance dissipée	14 W

Tableau II.3: Caractéristiques techniques de la CPU-313C [1].

II.6.2.2. Le module E/S digitales intégré (DI24/DO16)

Nous avons vu que l'automate S7-313C possède son propre module E/S digitales, ses caractéristiques sont résumées dans le tableau suivant :

Données spécifiques au module	
Entrées :	
Nombre	24
Entrées utilisables pour les fonctions technologiques	12
Longueur de câble	600m
Visualisation	LED verte par voie
Sorties :	
Nombre	16
Sorties rapide (utilisées)	4
Longueur	600m
Visualisation	LED verte par voie

Tensions, courants, potentiels	
Entrées :	
Tension de charge nominale	24VDC
Irréversibilité	Oui
Nombre d'entrées pouvant être activées simultanément	24 jusqu'à 40°C en position horizontale 12 jusqu'à 60°C en position horizontale 12 jusqu'à 40°C en position verticale
Séparation galvanique :	
Entre voies et buses internes	Oui
Entre voies	Non
Courant absorbé par la tension de charge	70mA au max
Sorties :	24VDC
Tension de charge nominale	Non
Irréversibilité	3A jusqu'à 40°C en position horizontale
Courant total des sorties	2A jusqu'à 60°C en position horizontale 2A jusqu'à 40°C en position verticale
Séparation galvanique :	Oui
Entre voies et buses internes	Oui par groupe de 8
Entre voies	
Courant absorbé par la tension de charge	100mA au max
Caractéristiques pour la sélection d'un capteur (entrées)	
Tension d'entrée :	
Valeur nominale	24VDC
Pour le signal '1'	De 15 à 30VDC
Pour le signal '0'	De -3 à 5VDC
Courant d'entrée avec le signal '1'	9mA
Courant de repos admissible	1.5mA
Retard à l'entrée nominal	3ms, paramétrable (0.1/0.5/3/15ms)
Raccordement avec capteurs	Type BERO 2 fils

Caractéristiques pour la sélection d'un actionneur (sorties)	
Tension de sortie avec le signal '1'	24VDC (-0.8V)
Courant de sortie :	
Avec le signal '1'	
Valeur nominale	0.5A
Plage admissible	De 0.005A à 0.6A
Avec le signal '0'	0.5mA
Plage de résistance de charge	De 400Ohm à 4kOhm
Commande d'un actionneur TOR	possible

Tableau II.4: Caractéristiques techniques du module E/S digitales [1].

II.6.2.3. Le module E/S analogiques intégré (AI5/AO2) :

Les caractéristiques du module E/S analogique sont résumées dans le tableau suivant :

Donnée spécifiques au module	
Entrées :	
Nombre	4 voies entrées courant/tension 1 voie entrée résistance
Longueur de câble	100m
Sorties :	
Nombre	2
Longueur	200m
Tension, courants, potentiels	
Entrées :	
Entrée de résistance :	
Tension de marche à vide	2.5V
Courant de mesure	De 1.8 à 3.3 mA
Séparation galvanique :	
Entre voies et buses internes	Oui
Entre voies	Non
Sorties :	
Tension de charge nominale	24VDC
Irréversibilité	Oui
Séparation galvanique :	
Entre voies et buses internes	Oui
Entre voies	Non

Formation des valeurs analogiques	
Entrées :	
Principe de mesure	Codage de la valeur actuelle (approximation successive)
Temps d'intégration/conversion/résolution :	
Paramétrable	Oui
Temps d'intégration en ms	2.5/16.6/20
Fréquence d'entrée autorisée	Max .400Hz
Résolution	11bits+VZ
Constante de temps du filtre d'entrée	0.38 ms
Temps d'exécution de base	1 ms
Sorties :	
Résolution	11bits+VZ
Temps de conversion	1 ms
Temps d'établissement :	
Pour la charge ohmique	0.6 ms
Pour la charge capacitive	1 ms
Pour la charge inductive	0.5 ms
Caractéristiques pour la sélection d'un capteur	
Etendues d'entrée/Résistance d'entrée :	
Tension	$\pm 10 \text{ V}/100 \text{ k}\Omega$; 0 V à $10 \text{ V}/100 \text{ k}\Omega$
Courant	$\pm 20 \text{ mA}/50 \Omega$; 0 mA à $20 \text{ mA}/50 \Omega$; 4 mA à $20 \text{ mA}/50 \Omega$
Résistance	0Ω à $600 \Omega/10 \text{ M}\Omega$
Thermomètre de résistance	Pt 100/10 $\text{M}\Omega$
Tension d'entrée autorisée (limite) :	
Pour la tension d'entrée	Max. 30V
Pour l'entrée de courant	Max. 2.5V
Courant d'entrée autorisé :	
Pour la tension d'entrée	Max. 0.5mA

Pour l'entrée de courant	Max. 50mA
Raccordement du capteur de signaux :	
Pour la mesure de la tension	Possible
Pour la mesure du courant :	
Transducteur de mesure 2 fils	Possible avec alimentation externe
Transducteur de mesure 4 fils	Possible
Pour la mesure du courant :	
Borne de ligne 2fils	Possible
Borne de ligne 3fils	Impossible
Borne de ligne 4fils	Impossible
Caractéristiques pour la sélection d'un actionneur	
Tension	$\pm 10 \text{ V}$; 0 V à 10 V
Courant	$\pm 20 \text{ mA}$; 0 mA à 20 mA ; 4 mA à 20 mA
Résistance de charge :	
Capacitive (sortie tension)	min. 1 k Ω ; max. 0,1 μF
Inductive (sortie courant)	max. 300 Ω ; 0,1 mH
Sortie tension :	
Protection contre les courts-circuits	Oui
Courant de court-circuit	55mA
Sortie de courant :	
Tension de marche à vide	17V
Limite des courants/tensions appliqués de l'extérieur :	
Tension au niveau des sorties	16V
Courant	50mA
Raccordement des actionneurs :	
Pour la sortie du courant	
Raccordement 2 fils	Possible
Montage à 4 fils	Impossible
Pour la sortie courant :	
Raccordement à 2fils	Possible

Tableau II.5: Caractéristiques techniques du module E/S analogique [1].

Remarque :

1. Les entrées et sorties du module sont adressées à partir de l'adresse initiale du module. L'adresse d'une voie correspond à l'adresse de début du module plus un déport d'adresse (2 octets).
2. Les voies d'entrées non utilisées doivent être court-circuitées. Par cette mesure, on obtient une immunité optimale aux perturbations pour le module analogique.
3. Les voies de sortie libres doivent être laissées à l'air libre.
4. Le module d'E/S analogique n'a pas d'étendues de mesure négatives.

La correspondance tensions valeurs analogiques est représentée dans le tableau qui suit

Système			Etendue de tension	
Pourcentage	Décimale	Hexadécimal	Tension	Domaine
117.56 %	32511	7EFF	11.76 V	Domaine de dépassement
	27649	6C01		
100 %	27648	6C00	10 V	
75 %	20736	5100	7.5 V	Etendue nominale
0.0036 %	1	1	361.7 μ V	
0 %	0	0	0 V	
	-1	FFFF	0 V	Impossible la tension min. d'entrées ou de sortie est limitée à 0 V
	-32768	8000	0 V	

Tableau II.6: Correspondance valeurs analogiques tension.

La correspondance courant valeurs analogiques est représentée dans le tableau qui suit :

Système			Etendue de tension	
Pourcentage	Décimale	Hexadécimal	Courant	Domaine
117.56 %	32511	7EFF	32.52 mA	Domaine de dépassement
	27649	6C01		

100 %	27648	6C00	20 mA	Etendue nominale
75 %	20736	5100	15 mA	
0.0036 %	1	1	723.4 nA	
0 %	0	0	0 mA	
	-1	FFFF	0 mA	Impossible la tension min. d'entrées ou de sortie est limitée à 0 V
	-32768	8000	0 mA	

Tableau II.7: Correspondance valeurs analogiques courant [1].

II.7. Conclusion

L'automate programmable est devenu un produit incontournable dans les systèmes automatisés de production. Grâce à la mobilité, la flexibilité de son architecture, la facilité de sa programmation, de sa connexion et de son adaptation dans les milieux industriels, Néanmoins la diversité des possibilités qu'il offre dans sa mise en œuvre et son coût ne Constituent pas des conditions suffisantes lors de l'élaboration d'une solution d'automatisme. En effet, une bonne analyse du problème à résoudre, ainsi que le respect des règles d'installation sont de rigueur.

Chapitre III

La programmation et la supervision sur le TIA Portal

III.1. Introduction

Les API ont pour fonctions de remplir des tâches de commande en élaborant des actions en suivant une algorithmique appropriée à partir d'informations données par des capteurs, cette algorithmique est écrite par un logiciel de programmation des API.

Dans ce qui va suivre nous allons étudier le logiciel de programmation et de Supervision TIA PORTAL, et de réaliser un programme en langage LADDER pour piloter la station Festo et de le tester dans un Automate Siemens S7-300, nous ferons aussi la configuration nécessaire pour l'acquisition des données depuis l'automate.

III.2. Concepts fondamentaux du portail TIA [8]

III.2.1. Présentation du portail TIA

Le portail Totally Integrated Automation (portail TIA) intègre différents produits SIMATIC dans une application logicielle qui nous permet d'accroître notre productivité et notre efficacité. Les produits TIA fonctionnent ensemble dans le portail TIA pour nous assister dans toutes nos tâches de création de solutions d'automatisation.

Dans une tâche d'automatisation typique :

- Un automate commande le processus à l'aide du programme,
- Un pupitre opérateur permet de conduire et de visualiser le processus.

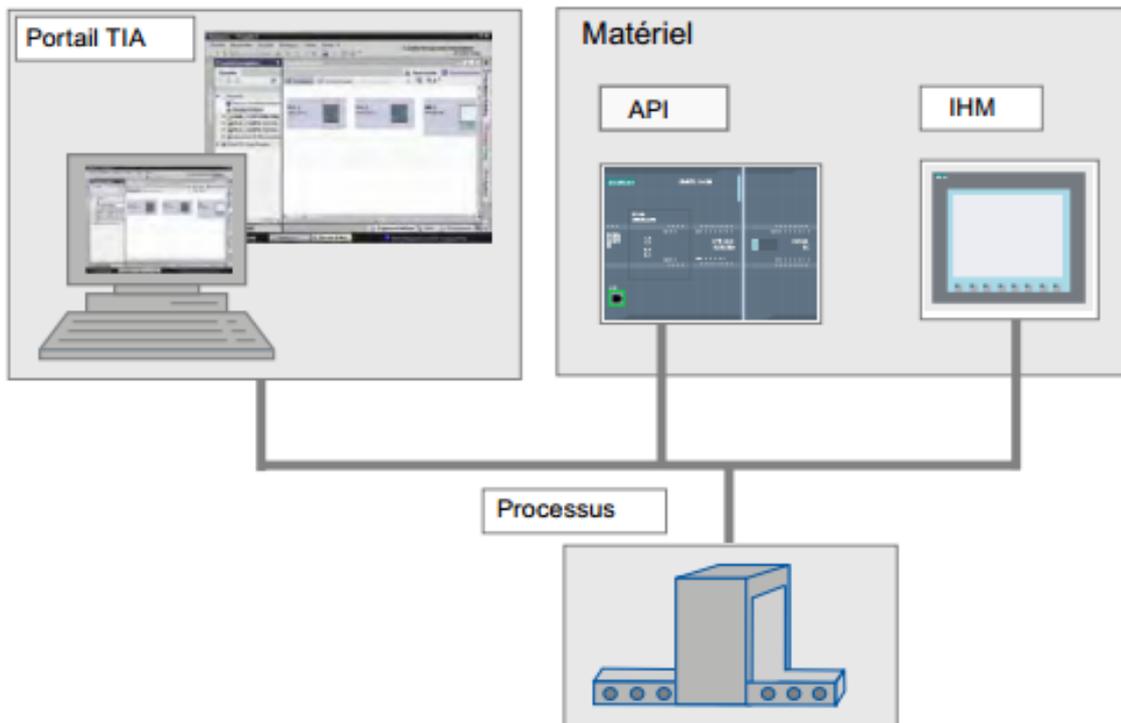


Figure III.1: TIA PORTAL

III.2.2. Concept d'ingénierie :

Grâce au portail TIA, nous configurons aussi bien la commande que la visualisation dans un système d'ingénierie homogène. L'ensemble des données est stocké dans un projet. Les composants pour la programmation (STEP 7) et la visualisation (WinCC) ne sont pas des programmes autonomes mais les éditeurs d'un système qui accède à une base de données commune. Toutes les données sont enregistrées dans un fichier de projet commun. Nous utilisons pour toutes les tâches une interface utilisateur commune nous permettant d'accéder à tout moment à toutes les fonctions de programmation et de visualisation.

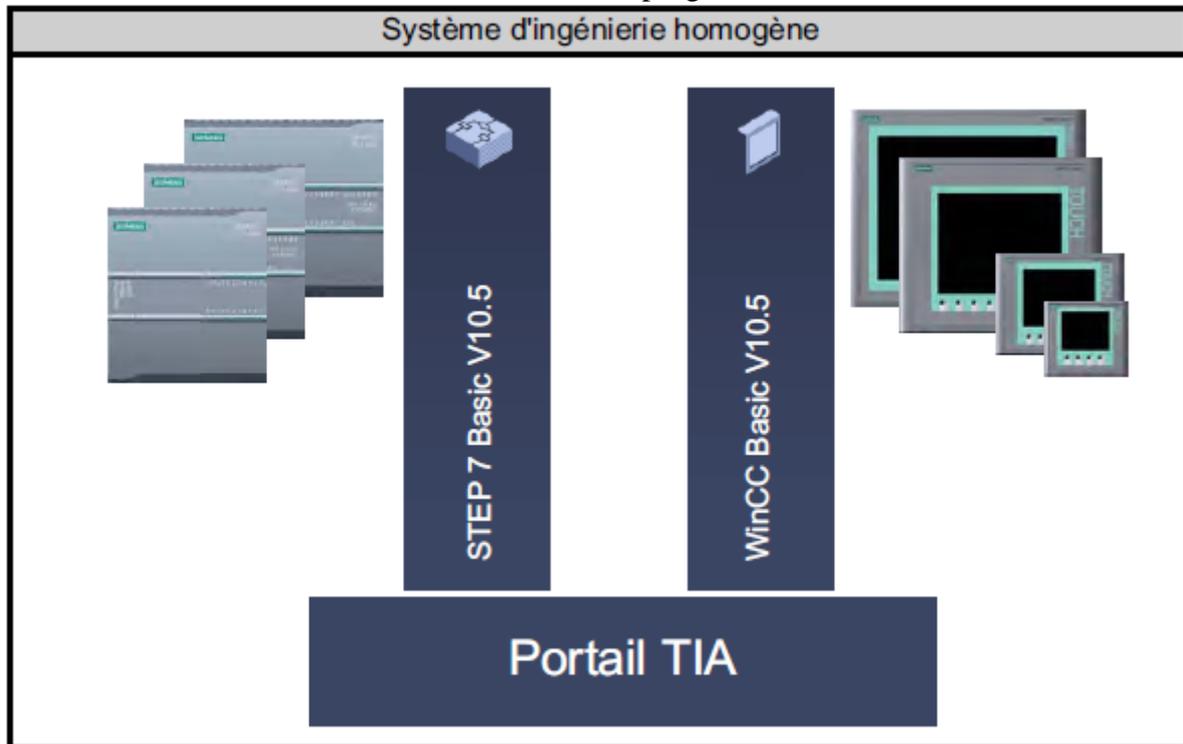


Figure III.2: Système d'ingénierie

III.2.3. Gestion des données

- Gestion centralisée des données

Dans le portail TIA, l'ensemble des données est stocké dans un projet. Les données d'application modifiées, par exemple les variables, sont actualisées automatiquement dans tout le projet, sur plusieurs appareils si nécessaire.

- Adressage symbolique global

Lorsque vous utilisez une variable de process dans plusieurs blocs de différents automates et dans des vues IHM, vous pouvez la créer ou la modifier à n'importe quel endroit dans le programme. Le bloc et l'appareil dans lesquels vous procédez à la modification sont Indifférents. Le portail TIA offre les possibilités suivantes pour la définition de variables API

- Définition dans la table des variables API
- Définition dans l'éditeur de programmes
- Définition par association avec des entrées et sorties de l'automate

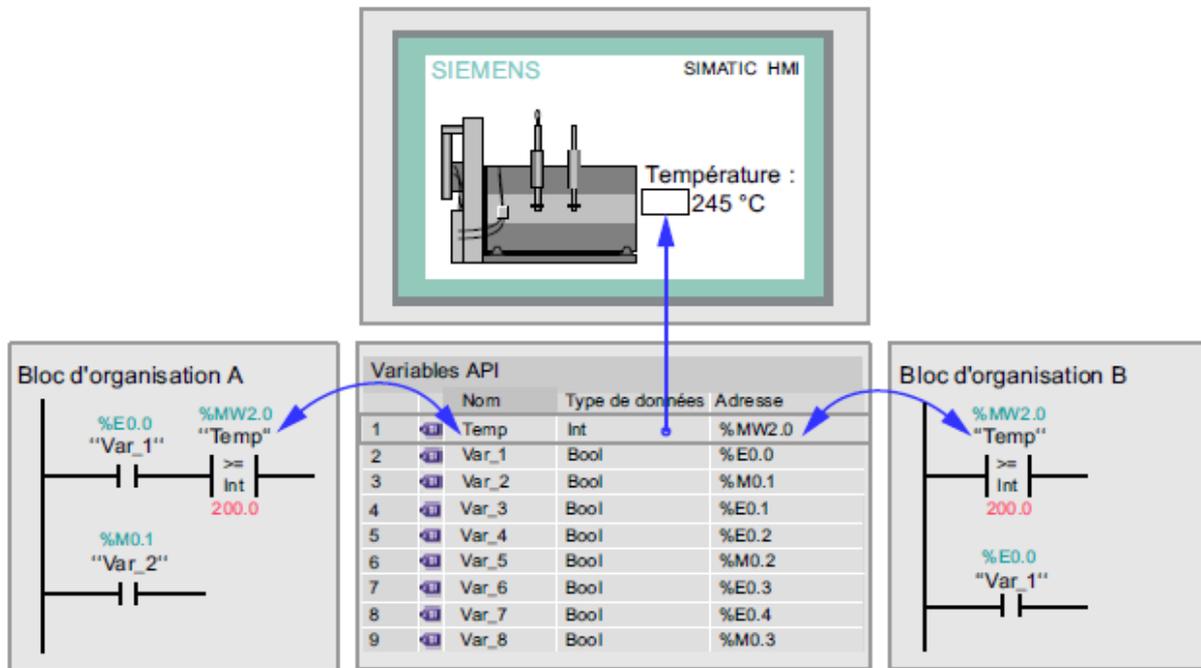


Figure III.3: Gestion des données

III.2.4. 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.
- 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.

III.3. Vues du portail TIA

III.3.1. Navigation dans le portail TIA

Nous travaillerons avec différentes vues lors de la création des projets. Aussi le paragraphe suivant nous donne-t-il un aperçu des vues disponibles dans le portail TIA.

Le portail TIA offre pour nos projets d'automatisation deux vues de travail différentes qui permettent un accès rapide aux outils et aux composants individuels du projet :

- Vue du portail : La vue du portail permet la configuration orientée tâche.
- Vue du projet : La vue du projet permet la configuration orientée objet.

➤ Navigation

Nous pouvons commuter à tout moment entre la vue du portail et la vue du projet grâce au lien situé dans le coin inférieur gauche de l'interface utilisateur.

Pendant la configuration, la vue change automatiquement en fonction du type de tâche exécutée.

➤ Enregistrement global des données du projet

Lors de l'enregistrement, c'est toujours l'ensemble du projet qui est sauvegardé, quelle que soit la vue ou quel que soit l'éditeur ouvert.

III.3.2. Vue du portail

La vue du portail offre une vue orientée tâche sur les outils. La vue du portail a pour but de nous faciliter la navigation dans les tâches et données du projet. A cet effet, les fonctions de l'application sont accessibles via des portails distincts, en fonction des principales tâches à réaliser. La figure suivante montre la structure de la vue du portail :

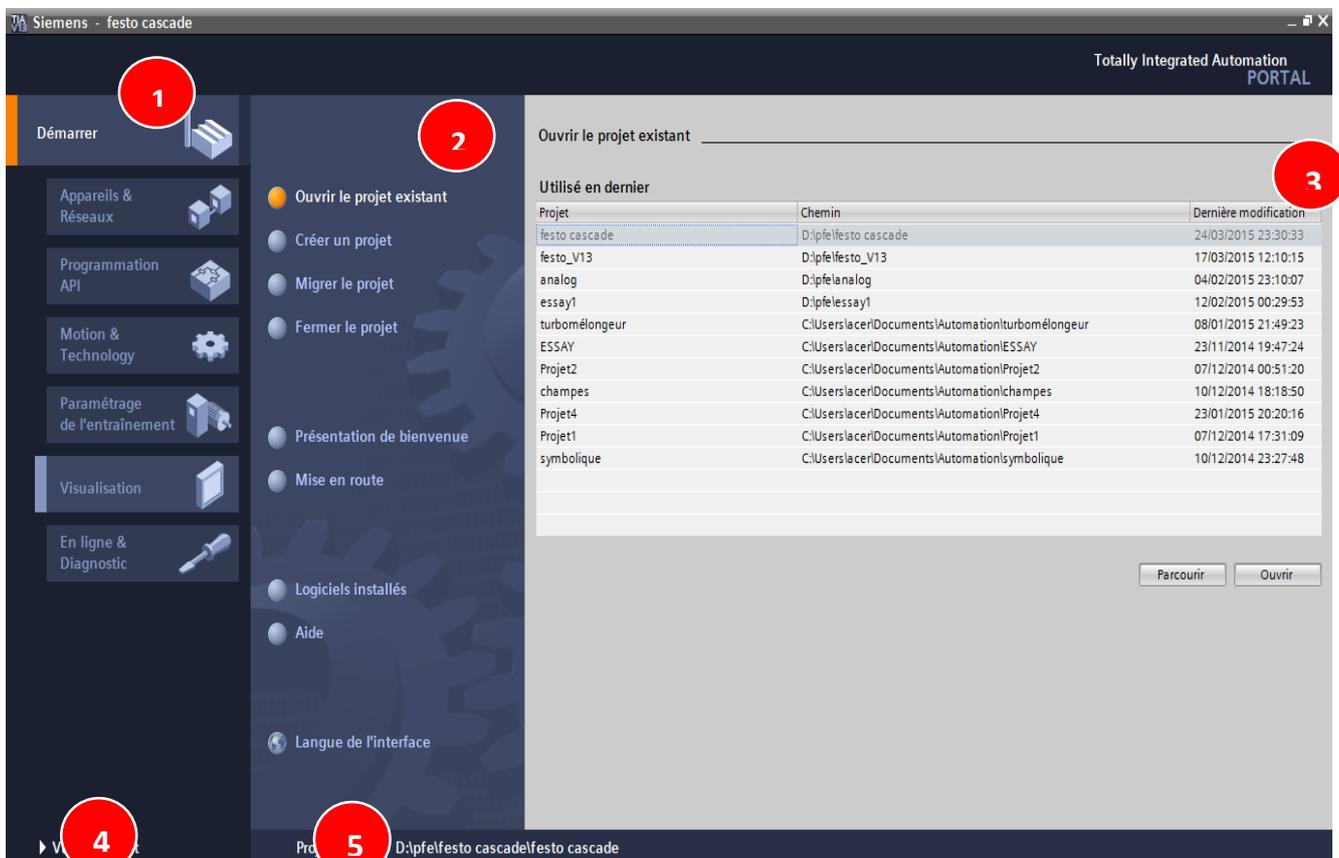


Figure III.4 : la vue du portail

① Portails pour les différentes tâches:

Les portails mettent à disposition les fonctions élémentaires requises par chaque type de tâche. Les portails proposés dans la vue du portail dépendent des produits installés.

② Actions correspondant au portail sélectionné:

En fonction du portail sélectionné, les actions que nous pouvons exécuter dans ce portail nous sont proposées ici. L'appel d'une aide contextuelle nous 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 Basculement dans la vue du projet:

Le lien "Vue du projet" vous permet de basculer dans la vue du projet.

⑤ Affichage du projet actuellement ouvert:

Indique quel est le projet actuellement ouvert.

III.3.3. Vue du projet

La vue du projet correspond à une vue structurée de l'ensemble des composants d'un projet. Dans la vue du projet, nous disposons de différents éditeurs à l'aide desquels nous créons et éditons les composants de projet correspondants.

La figure suivante montre la structure de la vue du projet :

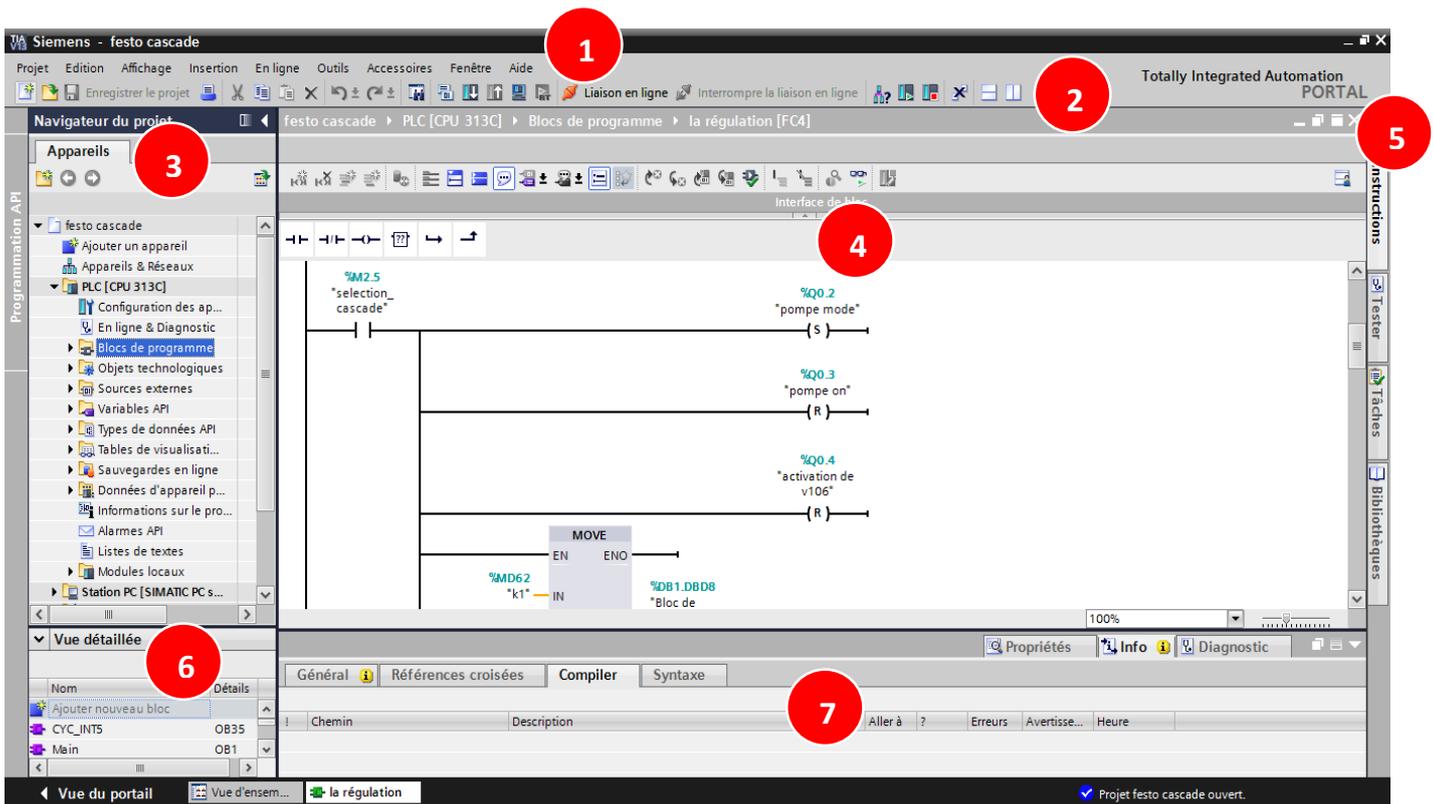


Figure III.5 : la vue du projet

① Barre des menus:

La barre des menus contient toutes les commandes dont vous avez besoin pour réaliser notre tâche.

② Barre d'outils:

La barre d'outils met à notre disposition des boutons nous permettant d'exécuter les commandes les plus fréquemment utilisées. L'accès à ces commandes est ainsi plus rapide que via des menus.

③ Navigateur du projet:

Le navigateur du projet nous permet d'accéder à tous les composants et données du projet. Nous pouvons par exemple réaliser les actions suivantes dans le navigateur du projet :

- Ajouter de nouveaux composants
- Editer des composants existants
- Interroger et modifier les propriétés de composants existants

④ Zone de travail :

La zone de travail affiche les objets que nous ouvrons afin de les éditer.

⑤ Task Cards:

Nous disposons de Task Cards en fonction de l'objet édité ou sélectionné. Les Task Cards disponibles figurent dans une barre sur le bord droit de l'écran.

Nous pouvons les ouvrir ou les fermer à tout moment.

⑥ Vue détaillée :

La vue détaillée affiche certains contenus d'un objet sélectionné. Il peut s'agir par exemple de listes de textes ou de variables.

⑦ Fenêtre d'inspection:

La fenêtre d'inspection affiche des informations supplémentaires sur un objet sélectionné ou sur des actions exécutées.

⑧ Basculement dans la vue du portail:

Le lien "Vue du portail" nous permet de basculer dans la vue du portail.

III.4. la programmation de station FESTO

Introduction

Une régulation est nécessaire lorsqu'une grandeur physique doit avoir une valeur précise dans le processus et que cette valeur peut varier suite à des conditions externes non prévisibles, par exemple une température, une pression ou un niveau,

Dans cette partie nous allons programmer une régulation avancée (la régulation de niveau en cascade avec le débit) afin d'avoir une réponse rapide face aux perturbations extérieures ainsi qu'aux changements de consigne.

Les étapes de travail suivantes sont prévues :

- Création du programme
- Chargement du programme dans l'automate
- Création d'une vue IHM
- Test du programme
- Programmation de l'acquisition des données

III.4.1. Création du projet

Les étapes suivantes décrivent comment créer un nouveau projet. Les données et les programmes qui sont générés lors de la création d'une tâche d'automatisation sont stockés de manière ordonnée dans le projet.

Marche à suivre

Procédez de la manière suivante pour créer un projet :

1. Démarrez le portail Totally Integrated Automation.
Cliquez sur Démarrer > Programmes > Siemens Automation > TIA-Portal V11.

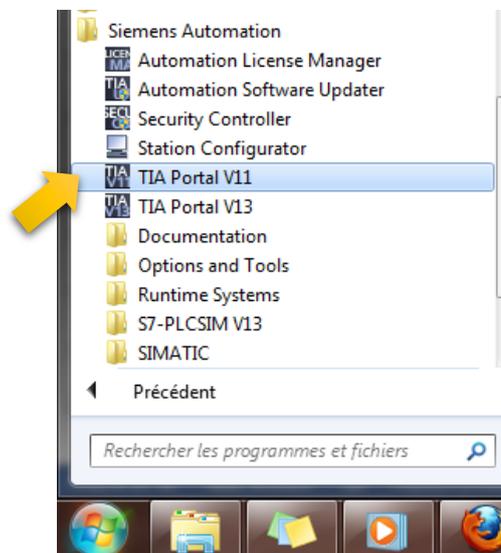


Figure III.6 : démarrage de TIA PRDAL V11

Le portail Totally Integrated Automation s'ouvre dans la vue du portail.

2. Créons le projet "festo cascade" dans un chemin au choix.

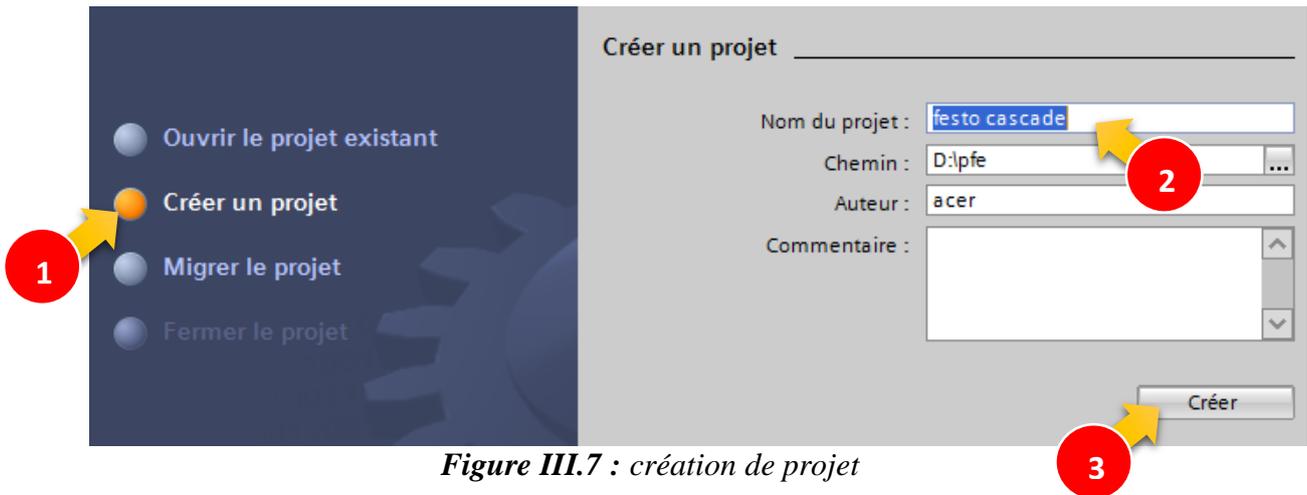


Figure III.7 : création de projet

III.4.2. Insertion et configuration d'un automate :

Les étapes suivantes décrivent comment insérer un automate par le biais de la vue du portail et comment ouvrir sa configuration dans la vue du projet. Le type d'automate que nous créons dans le projet doit correspondre à l'automate de la station FESTO.

Marche à suivre

Procédez de la manière suivante pour ajouter un nouvel appareil dans le projet :

1. On insère un nouvel appareil par le biais du portail.

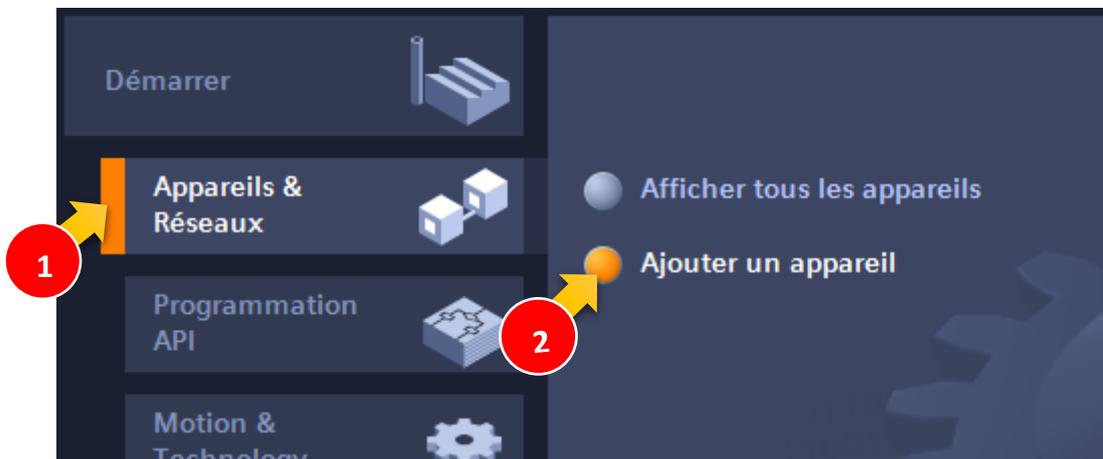


Figure III.8: Insertion et configuration d'un automate (1)

2. On sélectionne l'automate désiré qui est dans notre cas l'automate SIMATIC s7-300 CPU 313C.

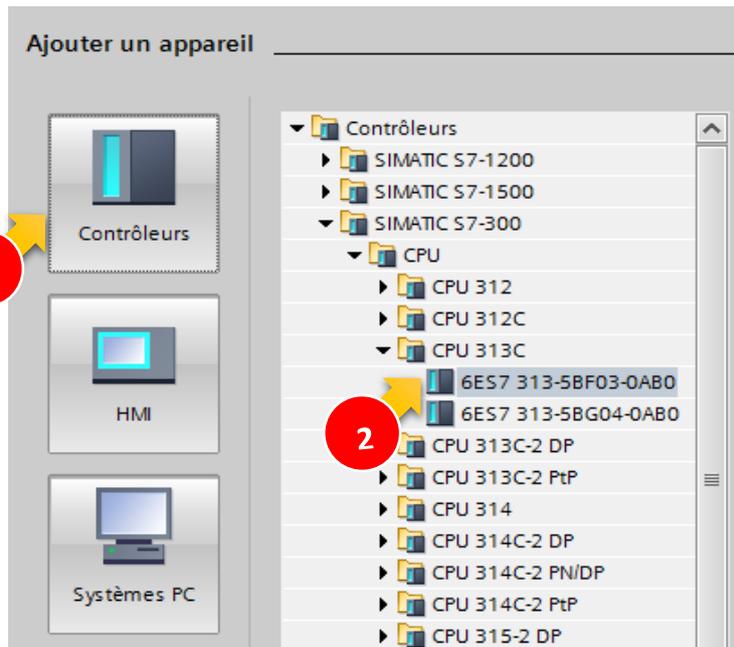


Figure III.9 : Insertion et configuration d'un automate (2)

3. On vérifie que l'option "Ouvrir la vue des appareils" est activée. Si elle est désactivée, On clique avec le bouton gauche de la souris sur l'option pour l'activer.
4. Cliquez sur le bouton "Ajouter".

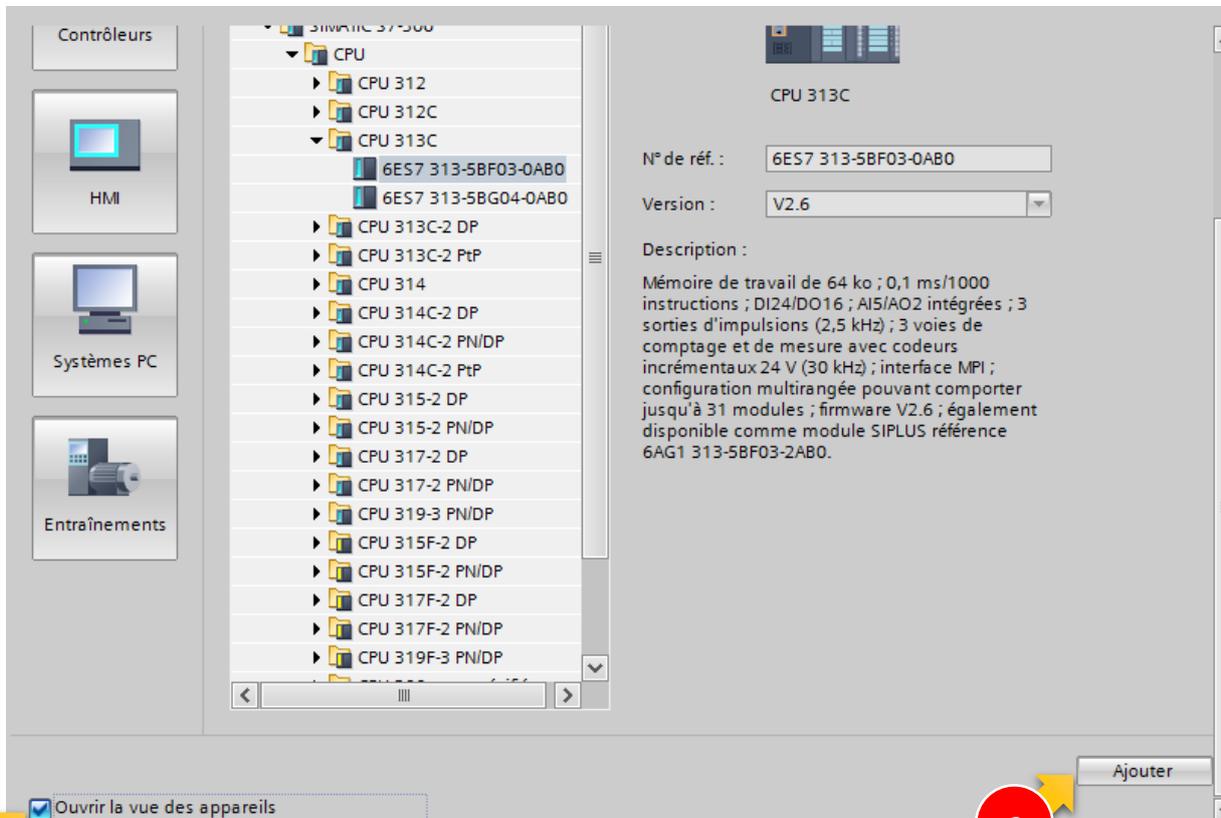


Figure III.10 : Insertion et configuration d'un automate (3)

Résultat

Vous avez créé un nouvel automate dans le projet et l'avez ouvert dans la vue des appareils de l'éditeur Appareils et réseaux.

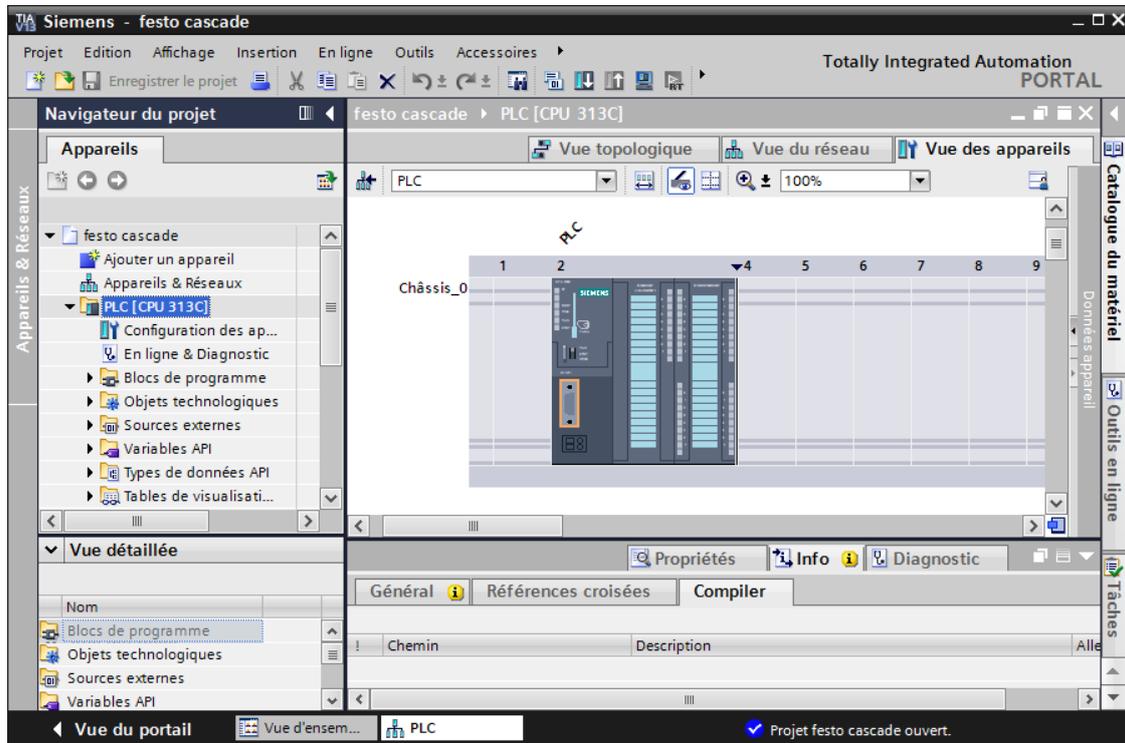


Figure III.11 : la vue des appareils de l'éditeur Appareils et réseaux.

III.4.3. Présentation de l'éditeur d'appareils et de réseaux :

L'éditeur d'appareils et de réseaux est l'environnement de développement intégré pour la configuration, le paramétrage et la mise en réseau des appareils et des modules. Il est composé d'une vue du réseau et d'une vue des appareils. Nous pouvons à tout moment basculer entre ces deux éditeurs.

III.4.3.1. Vue du réseau

La vue du réseau est la zone de travail de l'éditeur d'appareils et de réseaux dans laquelle nous pouvons exécuter les tâches suivantes :

- Configurer et paramétrer les appareils
- Interconnecter les appareils

La figure suivante montre la structure de la vue du réseau :

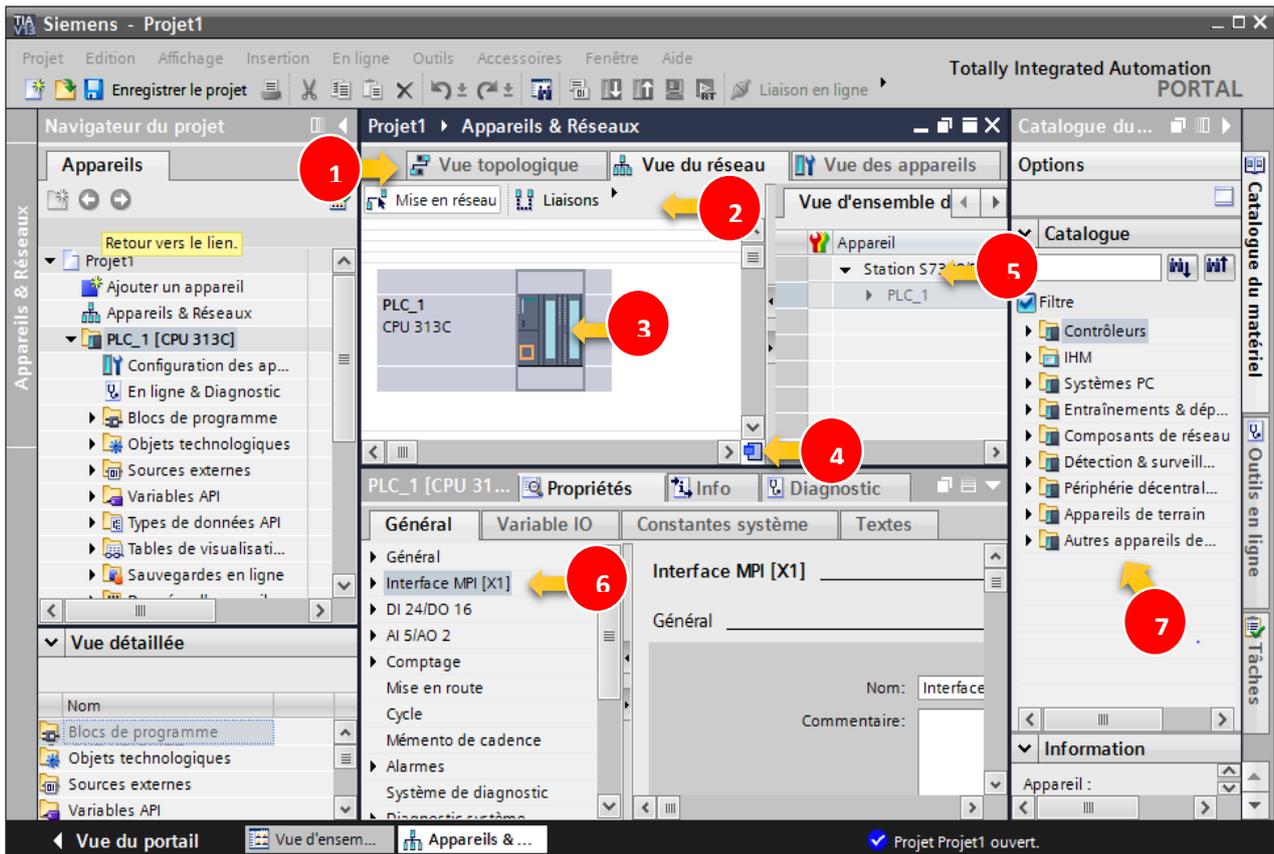


Figure III.12 : de la vue du réseau.

① Onglets pour commuter entre la vue des appareils et la vue du réseau

② Barre d'outils :

La barre d'outils contient les outils pour la mise en réseau graphique d'appareils, la configuration de liaisons et l'affichage des informations d'adresse. La fonction zoom nous permet de modifier la représentation dans la zone graphique.

③ Zone graphique :

La zone graphique permet d'afficher les appareils mis en réseau, les réseaux, les liaisons et relations. Nous pouvons insérer des appareils du catalogue du matériel (7) dans la zone graphique et les interconnecter par le biais de leurs interfaces.

④ Navigation générale :

La navigation générale donne une vue d'ensemble des objets créés dans la zone graphique. En maintenant le bouton de la souris enfoncé, vous pouvez naviguer rapidement dans la navigation générale vers les objets que vous souhaitez afficher dans la zone graphique.

⑤ Zone tabellaire :

La zone tabellaire présente une vue d'ensemble des appareils, connexions et liaisons de communication utilisés.

⑥ Fenêtre d'inspection :

La fenêtre d'inspection affiche les informations sur les objets actuellement sélectionnés. Dans

l'onglet "Propriétés" de la fenêtre d'inspection, vous pouvez éditer les paramètres des objets sélectionnés.

⑦ 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 du réseau.

III.4.3.2. Vue des appareils

La vue du réseau est la zone de travail de l'éditeur d'appareils et de réseaux dans laquelle vous pouvez exécuter les tâches suivantes :

- Configurer et paramétrer les appareils
- Configurer et paramétrer les modules

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

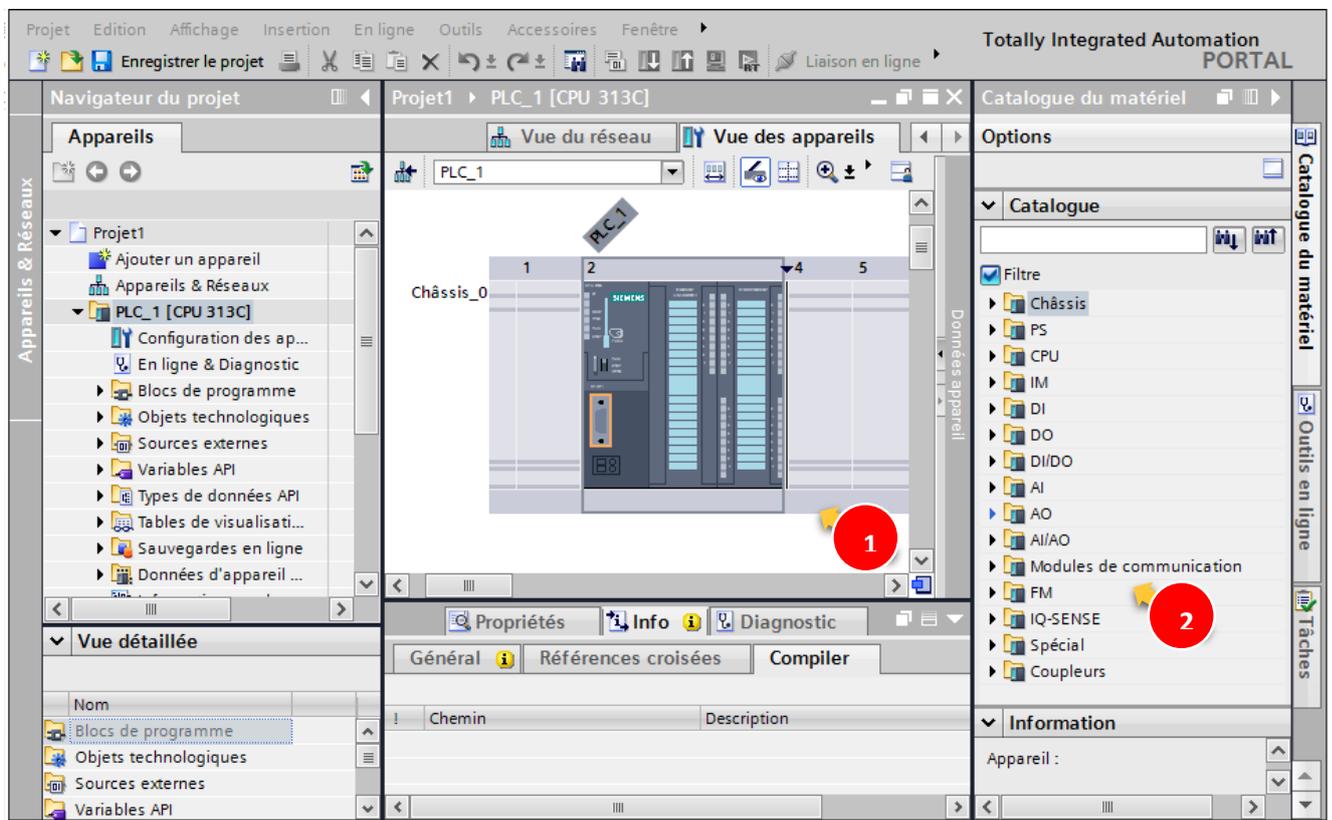


Figure III.13 : Vue des appareils.

① Zone graphique :

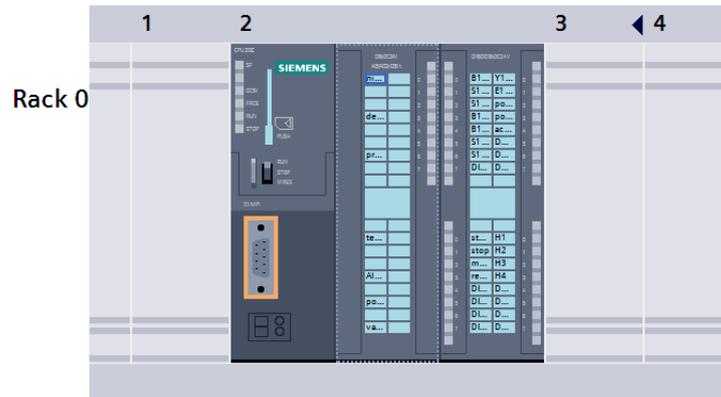
La zone graphique de la vue des appareils affiche les appareils et les modules associés qui sont affectés les uns aux autres via un ou plusieurs châssis. Dans la zone graphique, nous pouvons faire glisser d'autres objets matériels du catalogue du matériel (2) dans les emplacements des châssis et les configurer.

② 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.

III.4.3.3. Les modules de signaux

On travaille avec 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, déjà intégré dans la configuration de l'appareille



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.

Adressage :

Module	Plage d'adresses Entrées	Plage d'adresses Sortie
DI24/DO16	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	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

Création du programme :

Dans cette partie nous allons entamer la programmation de la station FESTO en suivant ce plan de travail, qui va être bien expliqué par la suite :

- Création de la table de variables
- Création du bloc de donnée
- Création du bloc d'organisation « main OB1 »
- Création des fonctions
- Compilation et simulation de programme

III.4.4. Création de la table de variables

Une variable est une grandeur utilisée dans le programme et pouvant prendre différentes valeurs. Les variables sont classées dans les catégories suivantes en fonction de leur domaine de validité :

- Variables locales: Les variables locales sont valables uniquement dans le bloc dans lequel elles ont été définies.
- Variables API: Les variables API sont valables dans tout l'automate.

Les variables sont gérées de manière centrale dans le portail TIA. Nous pouvons créer une variable API indifféremment dans l'éditeur de programmes ou dans la table des variables. Si la variable est utilisée à plusieurs endroits dans le programme ou dans la vue IHM, les modifications apportées à la variable sont immédiatement appliquées dans tous les éditeurs. La figure suivante montre la création de la de variables :

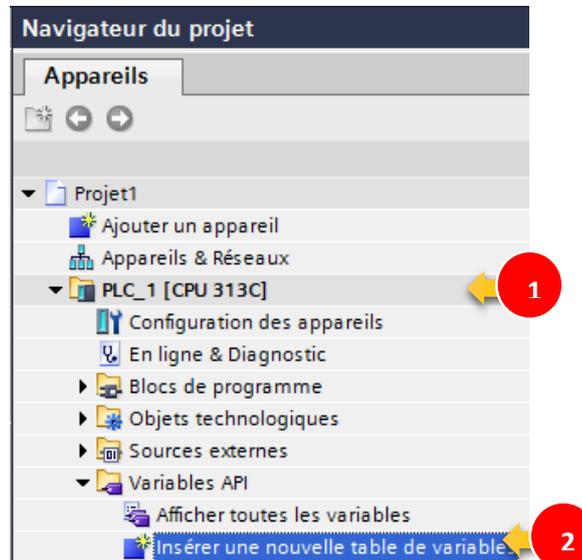


Figure III.14 : la création de la de variables

- Après l'insertion de la table de variables on la remplit de la manière suivante :

festo cascade > PLC [CPU 313C] > Variables API > Table de variables standard [51]

Variables | Constantes utilisateur | Cons

Table de variables standard							
	Nom	Type de données	Adresse	Réma...	Visibl...	Acces...	Commentaire
1	vanne v106	Int	%QW4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	bouton restart	Bool	%I1.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	capteur de niveau	Int	%IW3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	pompe analogique	Int	%QW2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	pompe mode	Bool	%Q0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	pompe on	Bool	%Q0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	b111	Bool	%I0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	bouton de demarage	Bool	%I1.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	bouton d'arret	Bool	%I1.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	b113	Bool	%I0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	niveau min
11	b114	Bool	%I0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	niveau max
12	man/ auto	Bool	%I1.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	activation de v106	Bool	%Q0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	capteur de debit	Int	%IW5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	h1	Bool	%Q1.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	indicateur de demmarage
16	h2	Bool	%Q1.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	indecateur reset
17	h3	Bool	%Q1.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	indecateur boucle ouvert
18	h4	Bool	%Q1.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	indecateur boucle fermé
19	Td1	DWord	%MD54		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
20	Td2	DWord	%MD58		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
21	k1	Real	%MD62		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
22	k2	Real	%MD66		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

23	 T1	DWord	%MD70		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
24	 T2	DWord	%MD74		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
25	 consigne niveau	Real	%MD10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
26	 reg debit pompe	Bool	%M0.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
27	 reg debit vanne	Bool	%M0.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
28	 restart	Bool	%M1.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
29	 erreur debit	Real	%MD30		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
30	 erreur niveau	Real	%MD34		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
31	 p1_active	Bool	%M1.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
32	 p2_active	Bool	%M1.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
33	 T1_active	Bool	%M2.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
34	 T2_active	Bool	%M2.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
35	 Td1_active	Bool	%M2.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
36	 Td2_active	Bool	%M2.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
37	 man/auto au	Bool	%M2.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
38	 selection_cascade	Bool	%M2.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
39	 Tag_1	Timer	%T1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
40	 marche/arret analogique	Bool	%M1.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
41	 s111	Bool	%I0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
42	 y102	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
43	 activation regulation	Bool	%M0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
44	 courbe p1	Bool	%M0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
45	 courbe ti1	Bool	%M1.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
46	 courbe td1	Bool	%M1.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
47	 courbe p2	Bool	%M1.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
48	 courbe ti2	Bool	%M1.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
49	 courbe td2	Bool	%M2.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
50	 pression_analogique	Int	%IW7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
51	 temperature_analogique	Int	%IW9		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
52	 ...				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figure III.15 : Tableau de référence

Le tableau suivant donne la signification des différentes colonnes de l'onglet "Variables" :

Colonne	Explication
	Symbole sur lequel vous pouvez cliquer pour faire glisser une variable dans un réseau et l'y utiliser comme opérande.
Nom	Nom univoque à l'intérieur de l'automate, que vous définissez pour la variable.
Type de données	Type de données que vous définissez pour la variable.
Adresse	Adresse de la variable
Rémanence	Les valeurs des variables rémanentes sont conservées même après coupure de la tension d'alimentation.
Valeur de visualisation	Valeur de données en cours dans l'automate. Cette colonne n'apparaît que s'il existe une liaison en ligne et que vous sélectionnez le bouton "Visualiser".
Commentaire	Commentaire pour la documentation de la variable

Tableau III.1 : la signification des différentes colonnes de l'onglet "Variables".

III.4.5. Création d'un bloc de donnée globale

Un bloc de données global (qui peut être appelé par n'importe quel bloc de code du programme) est un tableau formé d'un certain nombre d'étagère (maximum 256) sur lesquelles on peut ranger des données sous forme de booléens, d'entiers simple ou doubles, de réels simples ou doubles, de caractères, etc..

Remarque : il ne faut pas confondre avec un bloc de données d'instance qui est affecté à un bloc Fbi et qui est créé automatiquement au moment de la programmation de ce Fbi

- La figure suivante montre la création de bloc de données :

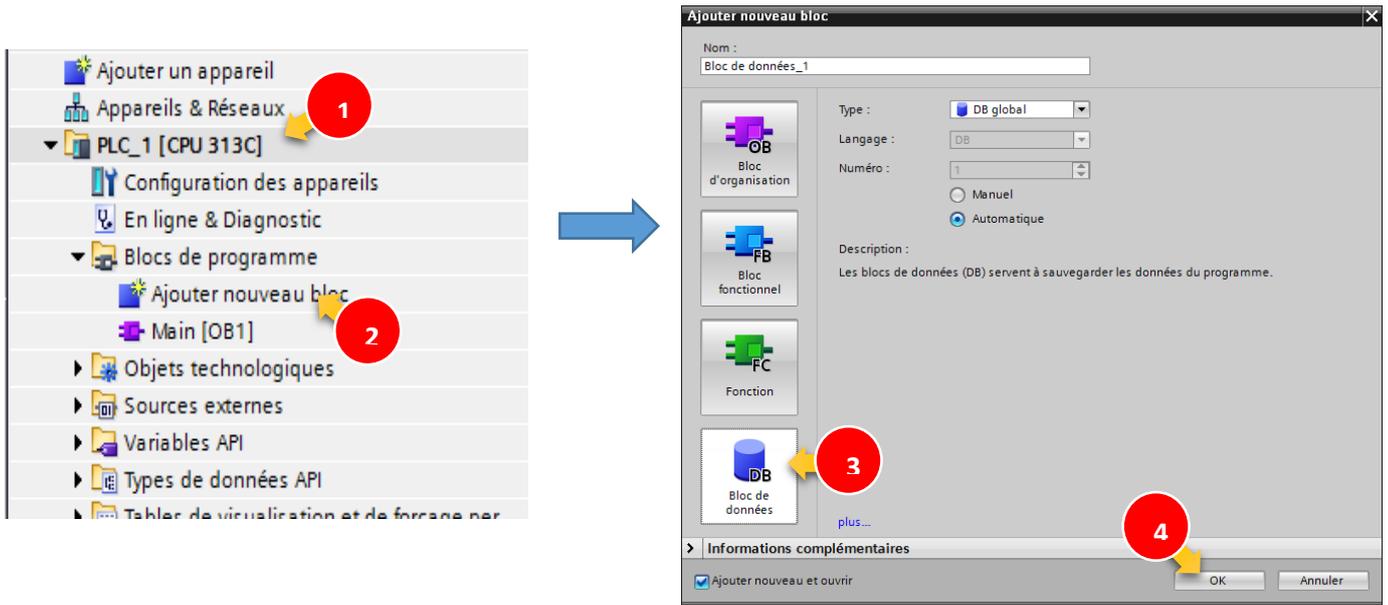


Figure III.16 : création d'un bloc de données globale.

- Après avoir ajouté le bloc de donnée on le remplit de la manière suivante :

festo cascade > PLC [CPU 313C] > Blocs de programme > B

Bloc de données_1		
	Nom	Type de données
1	Static	
2	niveau reel	Real
3	debit reel	Real
4	kp1	Real
5	kp2	Real
6	Ti1	Time
7	Ti2	Time
8	Td1	Time
9	td2	Time
10	entree perephirique1	Word
11	entree perephirique2	Word
12	valeur reglage1	Real
13	valeur reglage2	Real
14	erreur 1	Real
15	erreur2	Real
16	mesure1	Real
17	mesure2	Real
18	action_p1	Real
19	action_p2	Real
20	action_i1	Real
21	action_i2	Real
22	action_d1	Real
23	action_d2	Real
24	erreur_normee1	Real
25	erreur_normee2	Real
26	consign normee	Real
27	commande 1	Real
28	commande 2	Real
29	commande periphirique 1	Word
30	commande periphirique2	Word

Figure III.17 : les variables du bloc de données.

Maintenant on a créé le bloc de donnée "Global_DB" qui va nous aider à gérer ultérieurement les données du projet.

III.4.6. Création d'un bloc d'organisation « main OB1 »

III.4.6.1. Programme utilisateur

Un programme utilisateur peut être composé d'un ou de plusieurs blocs. Il faut utiliser au moins un bloc d'organisation. Les blocs contiennent toutes les fonctions requises pour le traitement de vos tâches d'automatisation spécifiques.

Font partie des fonctions du programme :

- Traitement des données du processus, par exemple combinaison de signaux binaires, lecture et évaluation de valeurs analogiques, détermination des signaux binaires de sortie, sortie des valeurs analogiques
- Réaction aux alarmes, par exemple alarme de diagnostic en cas de dépassement de la plage de mesure d'un module d'extension analogique
- Traitement des perturbations dans l'exécution normale du programme

III.4.3.1. Blocs d'organisation

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

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

Un projet d'automatisation doit comporter au moins un OB de cycle de programme. On écrit dans cet OB de cycle de programme le programme qui détermine le comportement de l'automate. Le système d'exploitation appelle cet OB une fois par cycle et lance ainsi le traitement du programme qu'il contient.

A la fin de chaque traitement du programme, le cycle recommence.

Avec la création de l'automate, le bloc d'organisation "Main [OB1]" est automatiquement créé dans le projet. Dans la suite, nous allons créer le programme utilisateur dans le bloc d'organisation.

La figure suivante montre l'exécution d'un OB de cycle de programme :

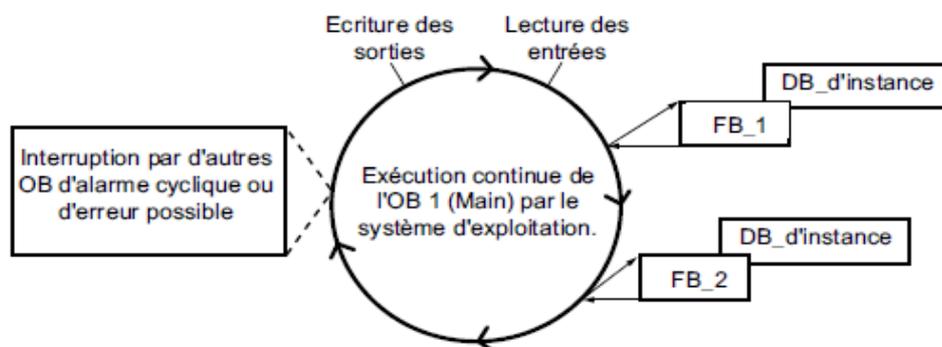


Figure III.18 : l'exécution d'un OB de cycle de programme.

Les étapes suivantes décrivent comment ouvrir le bloc d'organisation dans l'éditeur de programmes. L'éditeur de programmes est l'environnement de développement intégré pour la création du programme.

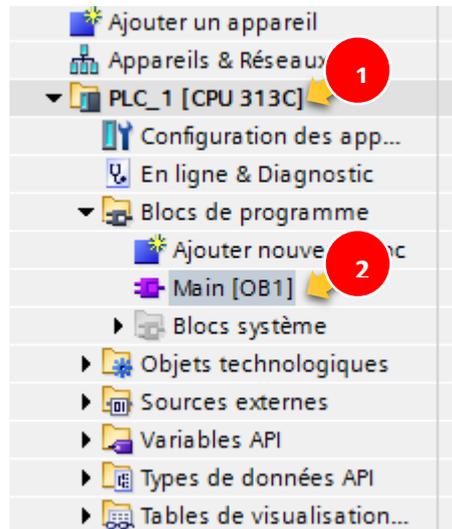


Figure III.19 : L'ouverture d'un bloc d'organisation

Nous avons ouvert le bloc d'organisation "Main [OB1]" dans l'éditeur de programmes et pouvons y créer notre programme.

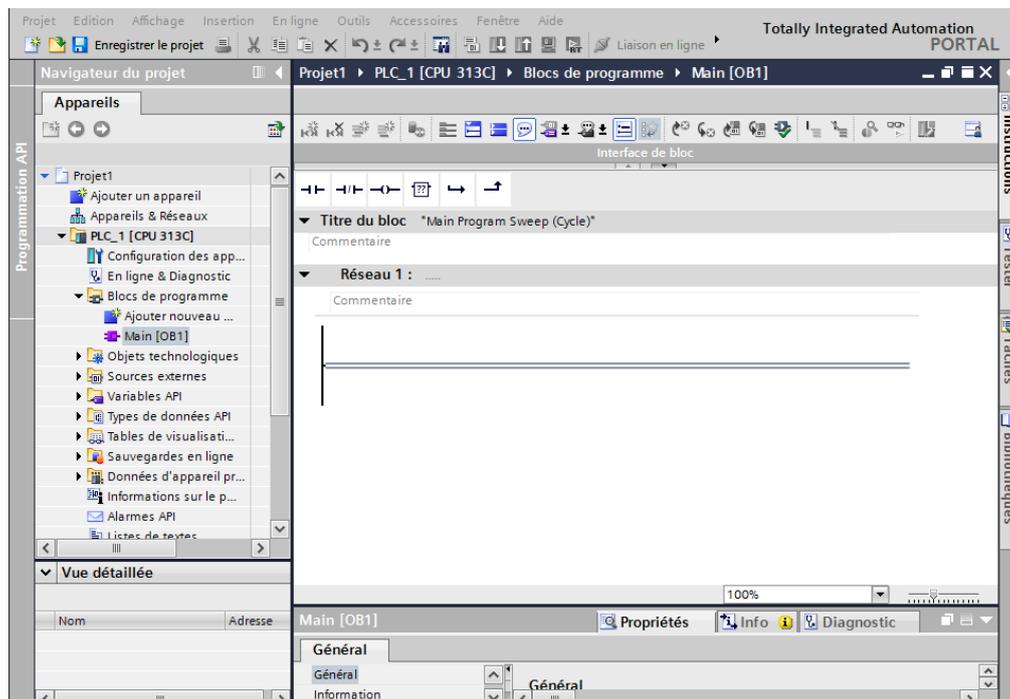


Figure III.20 : Main OB1.

❖ Présentation de l'éditeur de programmes :

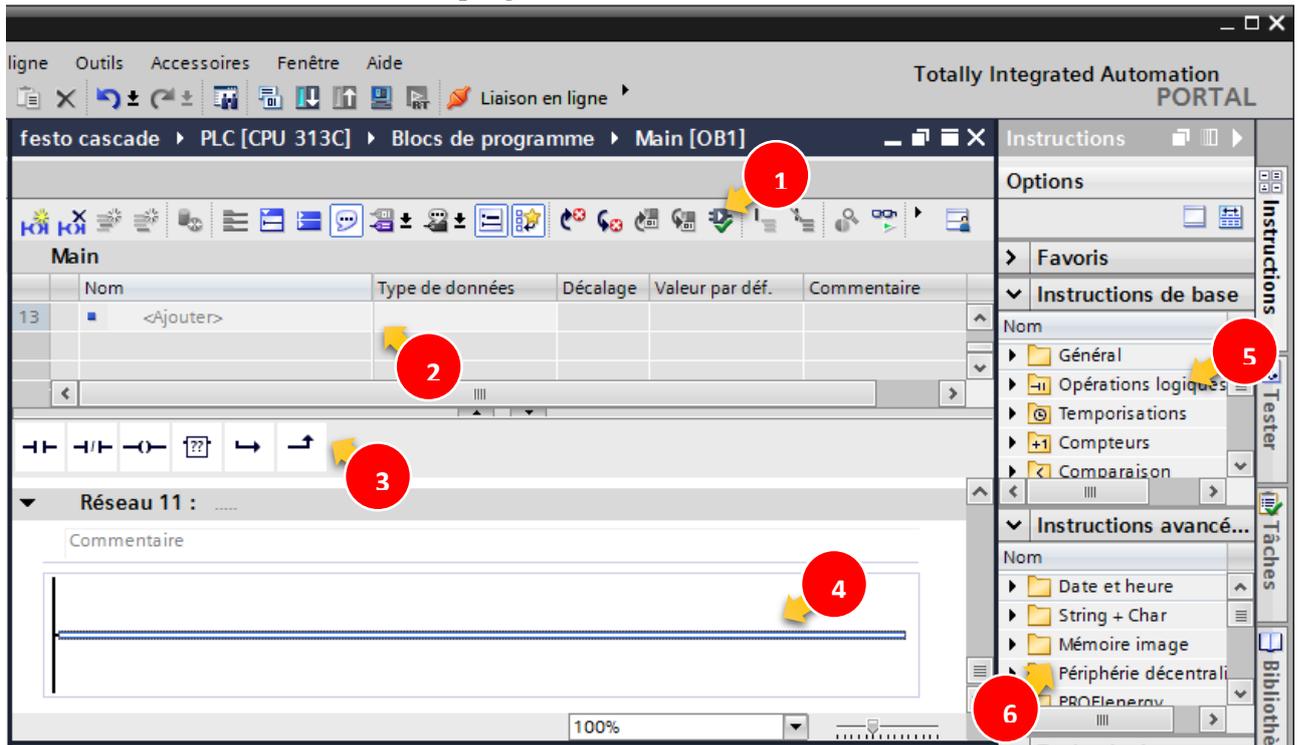


Figure III.21 : Présentation de l'éditeur de programmes

① Barre d'outils :

La barre d'outils permet d'accéder rapidement aux principales fonctions de l'éditeur de programmes, par exemple :

- Ajouter, supprimer, agrandir et réduire des réseaux
- Afficher et masquer les opérandes absolus
- Afficher et masquer les commentaires de réseau
- Afficher et masquer les favoris
- Afficher et masquer l'état du programme

② Interface de bloc :

L'interface de bloc sert à créer et à gérer les variables locales.

③ Palette "Favoris" dans la Task Card "Instructions" et Favoris dans l'éditeur de programmes
Les favoris permettent un accès rapide aux instructions souvent utilisées. Vous pouvez compléter la palette "Favoris" de manière individualisée avec des instructions supplémentaires.

④ Fenêtre d'instructions :

La fenêtre d'instructions est la zone de travail de l'éditeur de programmes. Vous pouvez y exécuter les tâches suivantes :

- Créer et gérer des réseaux
- Saisir les titres et commentaires du bloc et des réseaux
- Insérer des instructions et leur affecter des variables

⑤ Palette "Instructions" dans la Task Card "Instructions"

⑥ Palette "Instructions avancées" dans la Task Card "Instructions"

- Dans la programmation de station FESTO le main contient les différents fonctions nécessaire pour la régulation qui sont les suivants :

- La fonction mise à l'échelle:

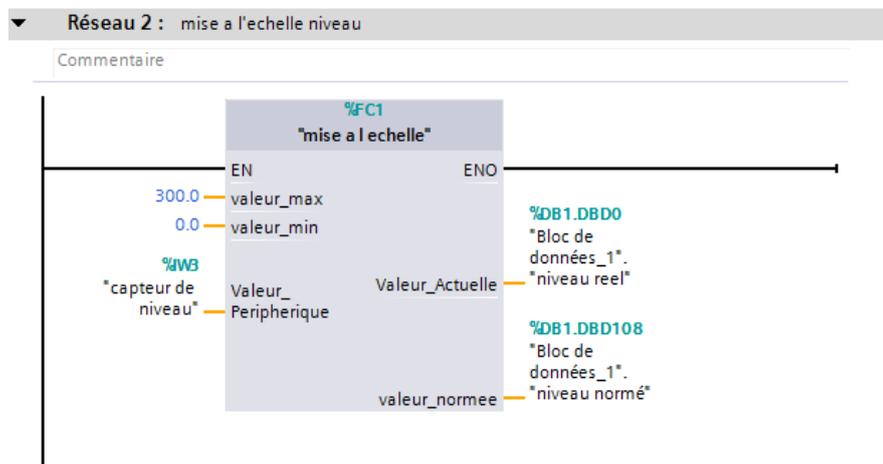


Figure III.22 : La fonction mise à l'échelle

La sortie de cette fonction est utilisée pour la visualisation de niveau, débit, température et la pression dans l'HMI

- La fonction de normalisation de consigne :

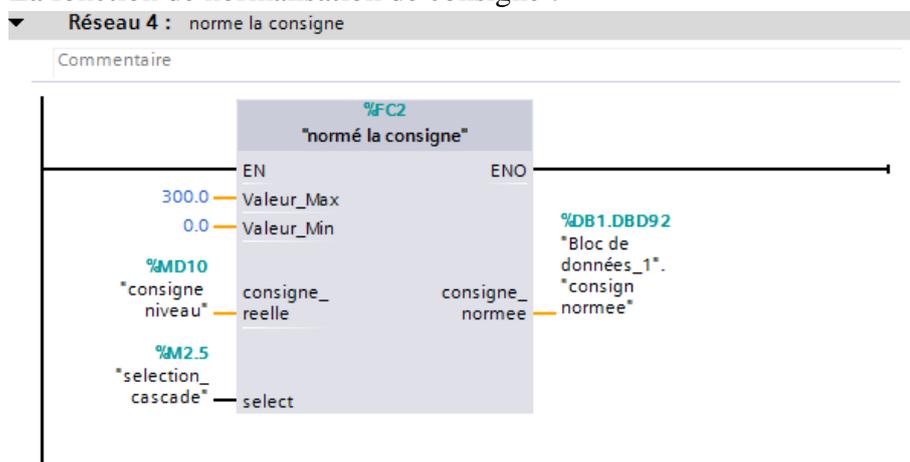


Figure III.23 : La fonction « normalisation de consigne ».

Ce bloc sert à normaliser la consigne qui va être utilisé comme entré dans le bloc PID et dans la visualisation HMI

- La fonction de l'erreur :

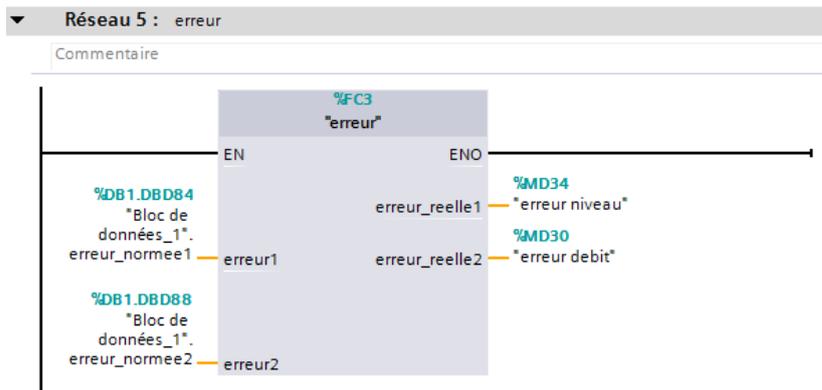


Figure III.24 : la fonction erreur.

Ce bloc sert à calculer l'erreur réelle à partir de l'erreur normée venant du bloc PID

- La fonction régulation :

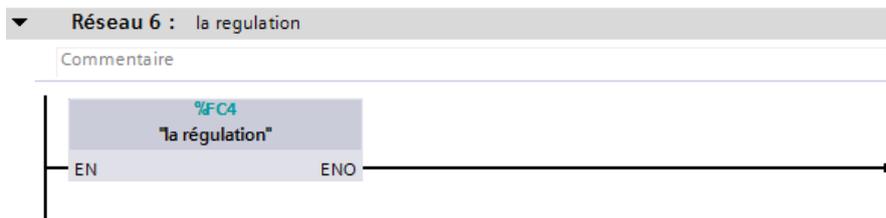


Figure III.25 : la fonction régulation.

Ce bloc sert à affecter les valeurs au paramètre PID et aussi les activé et désactivé

- La fonction start/stop :

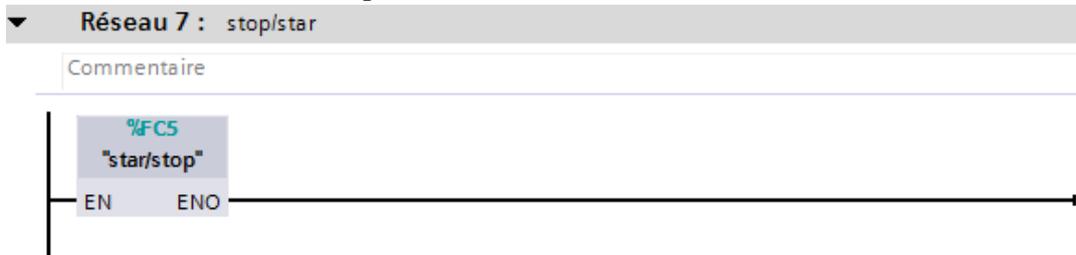


Figure III.26 : la fonction start/stop.

Ce bloc est fait pour programmer le bouton d'arrêt et le bouton de démarrage

- La fonction reset :

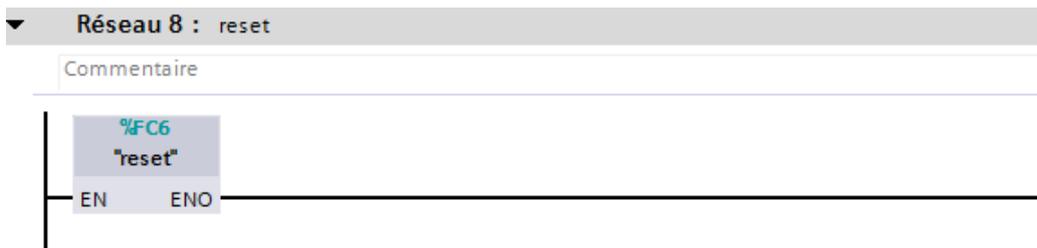


Figure III.27 : La fonction reset.

Ce bloc sert à programmer le bouton « reset »

III.4.7. Création d'une 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.

Pour créer une fonction 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.
 - Sélectionnez le type.
 - Cliquez sur "OK".

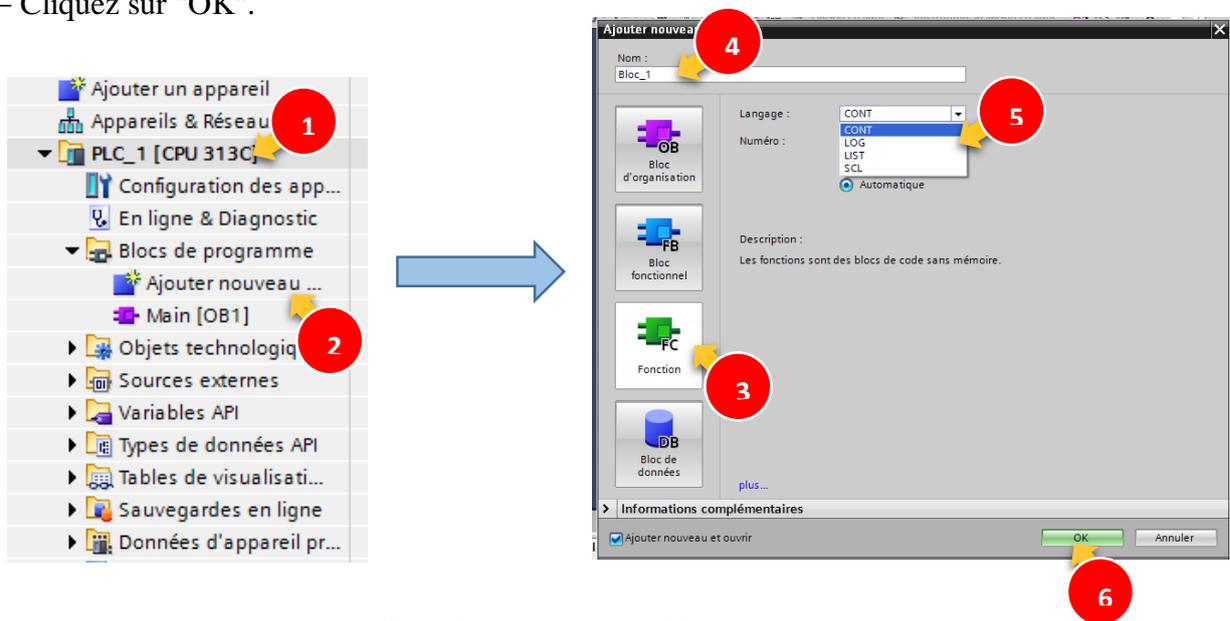


Figure III.28 : création d'une fonction.

La figure suivante montre la structure de l'éditeur :

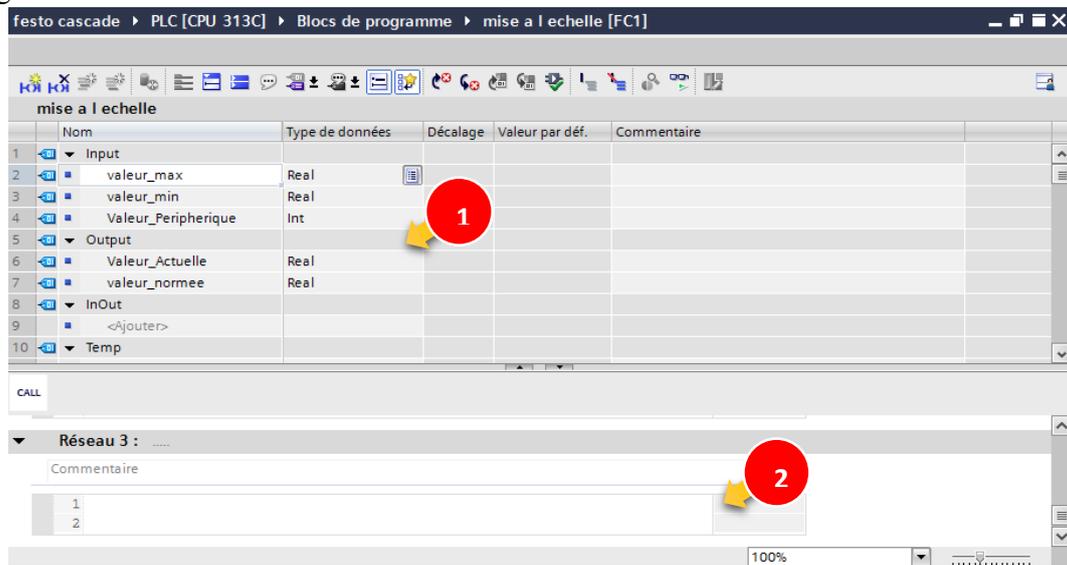


Figure III.29: La structure de l'éditeur.

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.

III.4.8. Création du bloc d'organisation pour le régulateur PID :

Les blocs pour le régulateur PID sont créés dans un nouveau bloc d'organisation. Nous utiliserons comme nouveau bloc d'organisation un OB d'alarme cyclique que nous allons créer par la suite.

Les blocs d'organisation d'alarme cyclique servent à démarrer des programmes à intervalles réguliers indépendamment du traitement cyclique du programme. Le traitement cyclique du programme est alors interrompu par l'OB d'alarme cyclique, puis reprend une fois l'OB traité. La figure suivante montre l'exécution du programme avec un OB d'alarme cyclique :

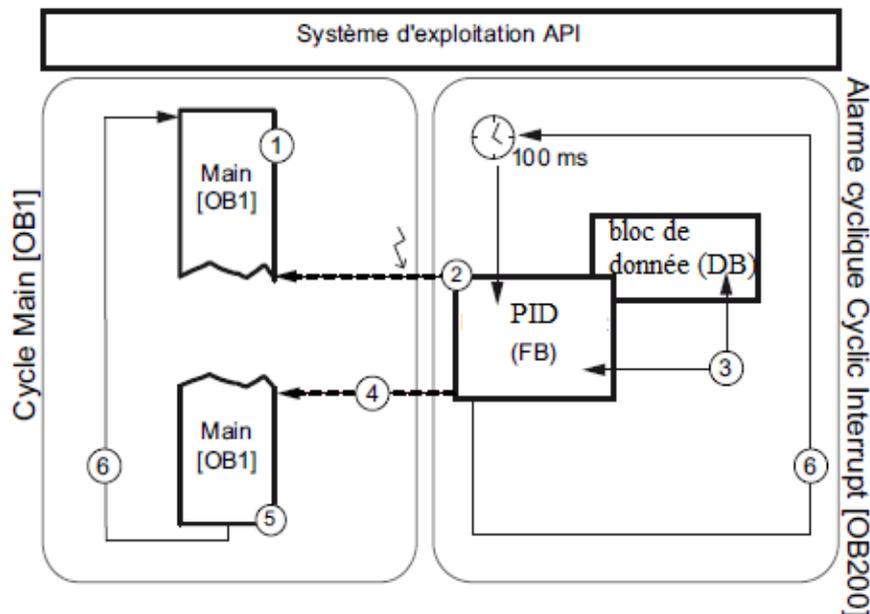


Figure III.30 : l'exécution du programme avec un OB d'alarme cyclique.

- ① Le programme commence avec Main [OB1].
- ② Toutes les 100 ms se déclenche une alarme cyclique qui interrompt le programme en cours à n'importe quel endroit (par exemple, pendant Main [OB1]) et qui exécute le programme figurant dans l'OB35 d'alarme cyclique. Ici, le programme se compose du bloc PID
- ③ PID est exécuté et les valeurs sont écrites dans le bloc de données (DB).
- ④ Une fois l'OB d'alarme cyclique traité, l'exécution de Main [OB1] reprend où elle avait été interrompue. Les valeurs sont conservées.
- ⑤ Main [OB1] est achevé.
- ⑥ Le cycle du programme reprend du début.

Dans le projet FESTO, nous utilisons l'OB d'alarme cyclique pour appeler l'objet technologique "PID". L'objet technologique "PID" est l'image du régulateur PID dans le logiciel. Cet objet technologique nous permet de configurer un régulateur PID, de l'activer et de contrôler l'état d'exécution.

- Dans notre projet on a utilisé deux régulateurs PID, un pour la boucle intérieur et l'autre pour la boucle extérieur.

Pour créer un OB d'alarme cyclique pour le régulateur PID, on Procède de la manière suivante :

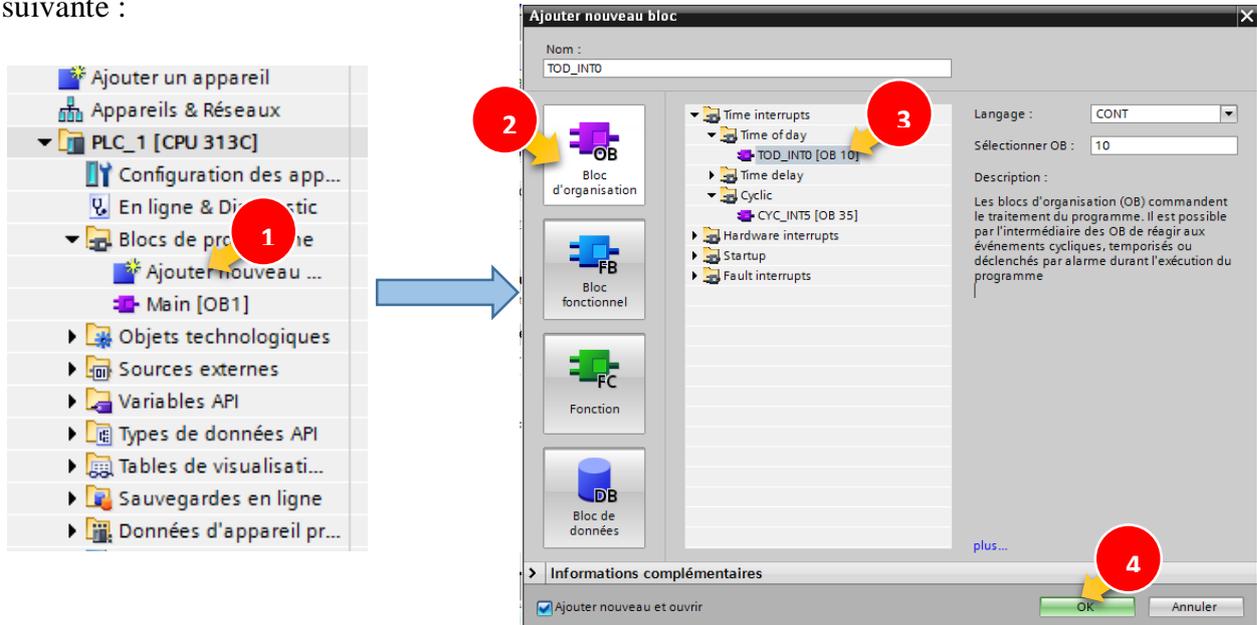


Figure III.31 : création d'un OB d'alarme cyclique.

- Les étapes suivantes décrivent comment appeler l'objet technologique PID dans l'OB35 d'alarme cyclique.

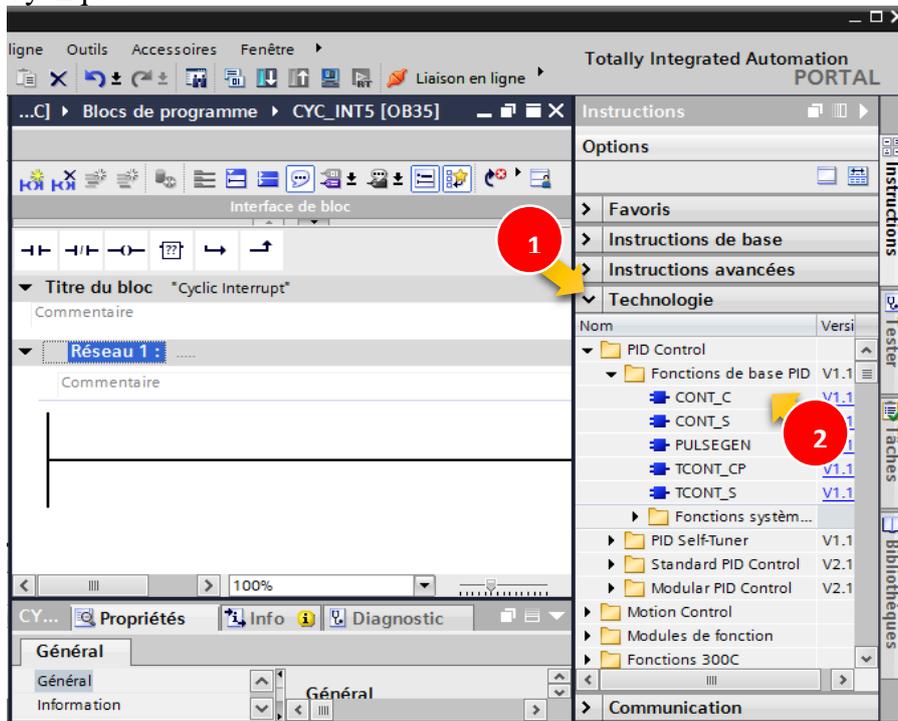


Figure III.32 : l'insertion d'un bloc PID.

- 1- On ouvre le bloc d'instruction
 - 2- On ouvre le bloc technologique >> « PID control » >> « fonction de base PID »
 - 3- Puit on glisse le bloc « CONT_C »
- La figure suivante montre la structure du bloc de régulation et ses paramètre d'entrée/sortie qu'on va les expliquer par la suite:

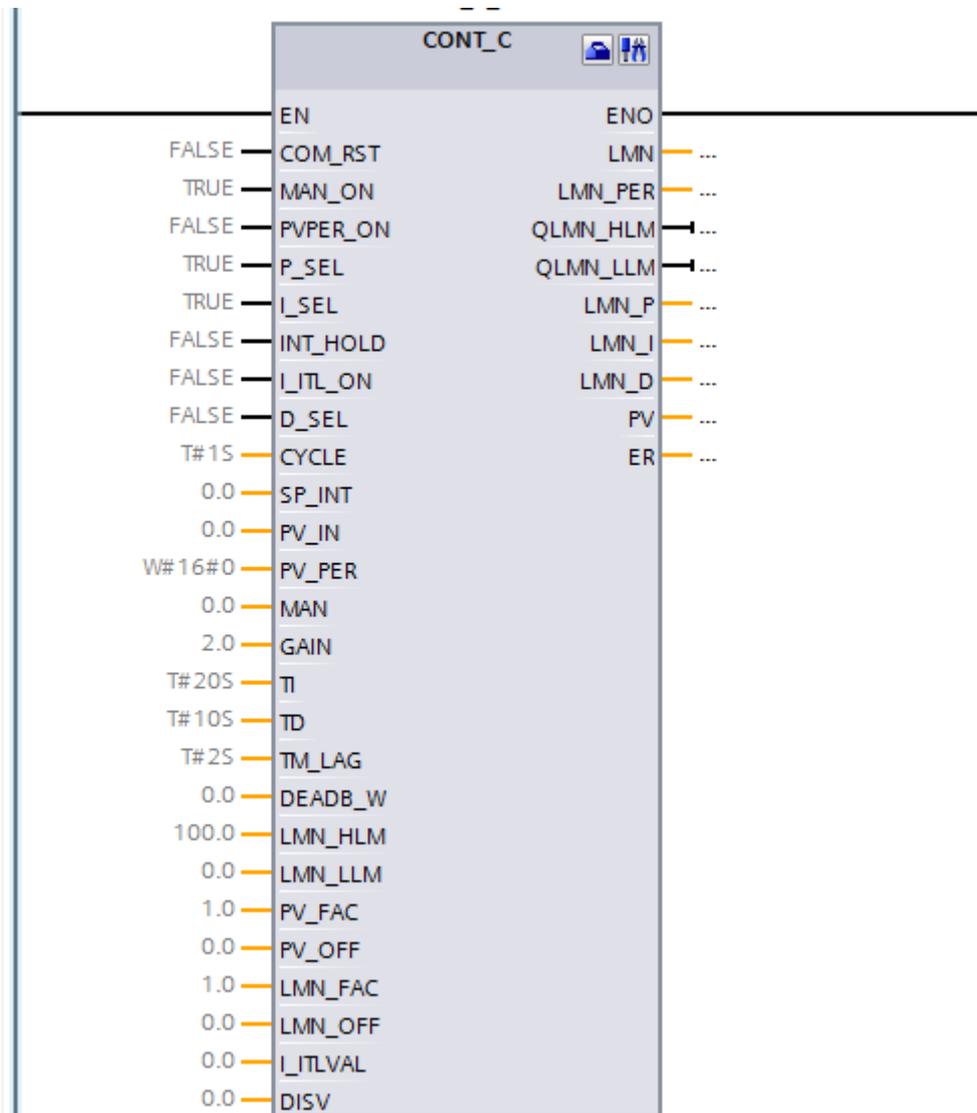


Figure III.33 : bloc PID.

III.4.8.1. 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.

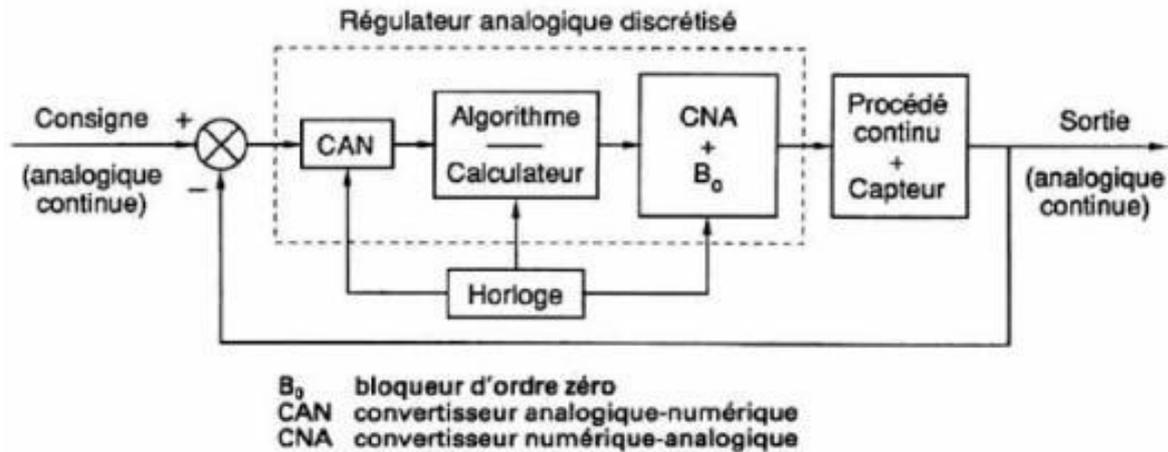


Figure III.34 : Schéma de principe de la régulation via un API.

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.

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.

III.4.8.2. Régulation continue avec le bloc « CONT_C »

Le bloc « 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 suivante:

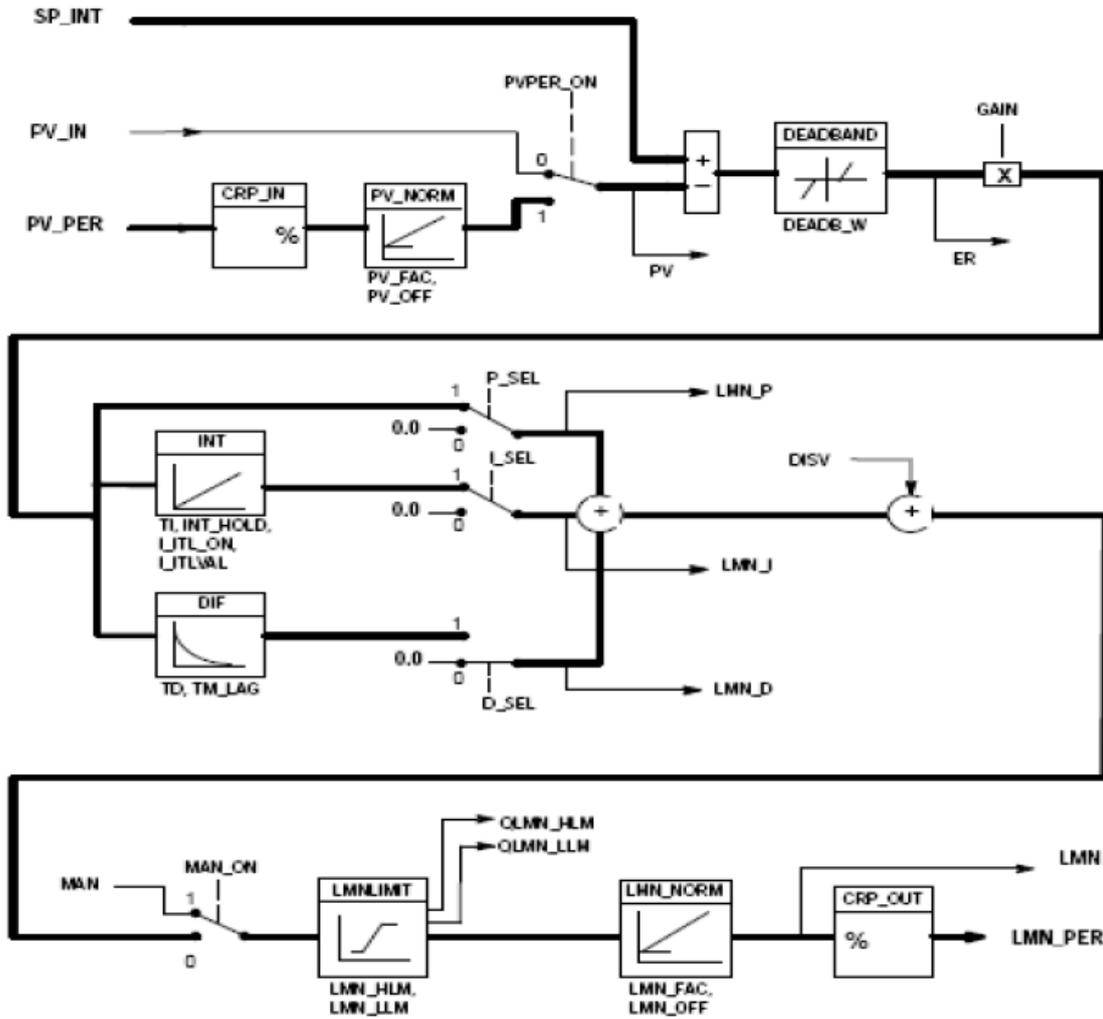


Figure III.35: Schéma de principe du bloc PID.

Les tableaux suivants 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			PROCESS VARIABLE PERIPHERIE / Mesure de périphérie La mesure en format de périphérie est appliquée au régulateur par cette entrée.
MAN	REAL	-100 à 100%		MANUAL VALUE / Valeur de réglage manuelle Cette entrée sert à introduire une valeur de réglage manuelle moyennant des fonctions de contrôle-commande
GAIN	REAL			PROPORTIONAL GAIN / Coefficient d'action proportionnelle Cette entrée indique le gain du régulateur.

TI	TIME	\geq cycle		RESET TIME / Temps d'intégration Cette entrée détermine la réponse temporelle de l'intégrateur
TD	TIME	\geq cycle		DERIVATIVE TIME / Temps de dérivation Cette entrée détermine la réponse temporelle de l'unité de dérivation.
TM_LAG	TIME	\geq cycle/2		TIME LAG OF THE DERIVATE ACTION / Retard de l'action par dérivation L'algorithme de l'action D contient un retard qui peut être paramétré à cette entrée
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			MANIPULATED VALUE FACTOR / Facteur de valeur de réglage Cette entrée est multipliée par la valeur de réglage. Elle sert à adapter l'étendue de réglage.
LMN_OFF	REAL			MANIPULATED VALUE OFFSET / Décalage de valeur de réglage Cette entrée est ajoutée à la valeur de réglage. Elle sert à adapter l'étendue de réglage.
I_ITLVAL	REAL	-100 à 100%		INITIALIZATION VALUE OF THE INTEGRAL ACTION / Valeur d'initialisation pour l'action par intégration 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.
DISV	REAL	-100 à 100%		DISTURBANCE VARIABLE / Grandeur perturbatrice La grandeur perturbatrice est appliquée à cette entrée pour l'action anticipatrice.

Tableau III.2: Entrées du bloc PID.

- Paramètre 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 III.3: Sorties du bloc PID.

III.4.9. Création d'une interface IHM

Introduction

Un système IHM constitue l'interface entre l'utilisateur et le processus. Le processus est pour l'essentiel piloté par l'automate. L'utilisateur peut visualiser le processus ou intervenir dans le processus en cours par le biais d'un pupitre opérateur.

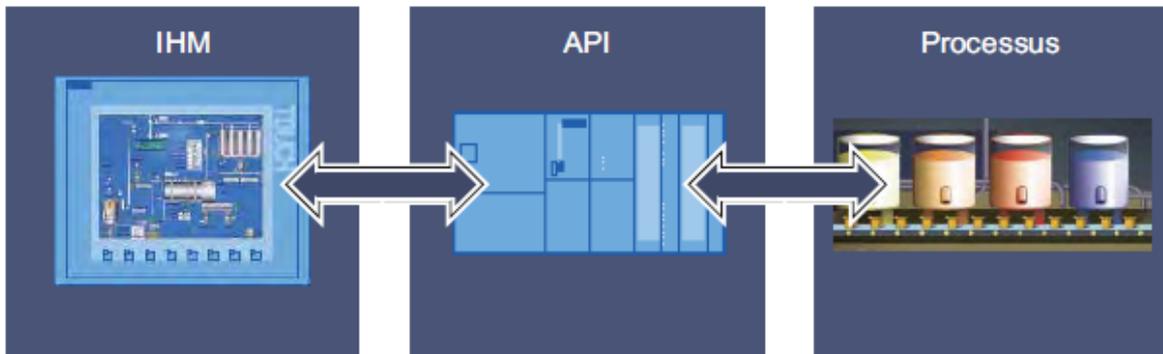


Figure III.36 : « IHM » l'interface entre l'utilisateur et le processus.

Les possibilités suivantes nous sont offertes pour le contrôle-commande de machines et d'installations :

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

III.4.9.1. Création d'un pupitre opérateur avec vue IHM

On procède comme suit pour ajouter un pupitre opérateur :

1. On insère un nouvel appareil par le biais du navigateur du projet. et On ajoute une station PC

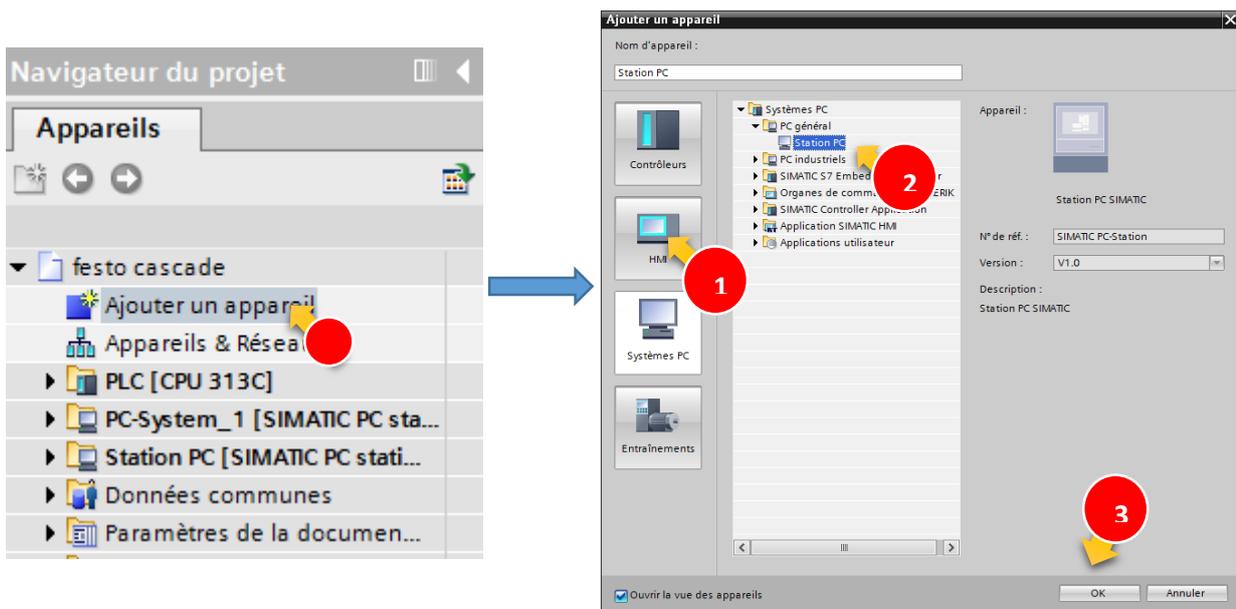


Figure III.37 : création d'un pupitre(1).

2. Et on ajoute « WinCC RT Advanced » dans la station PC

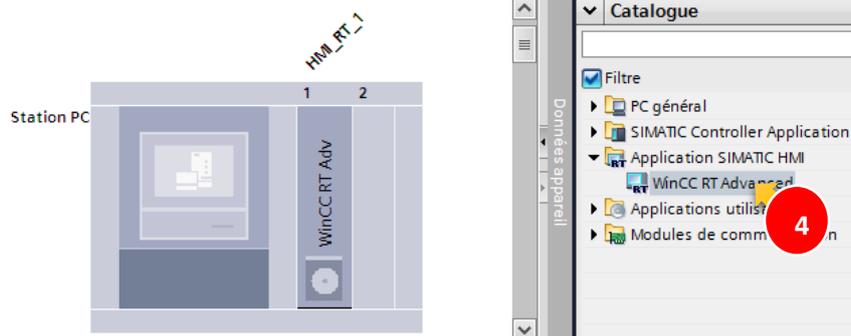


Figure III.38 : création d'un pupitre(2).

3. Après avoir créé le pupitre on crée maintenant les vues, on clique sur station pc → HMI_RT → ajouter une vue

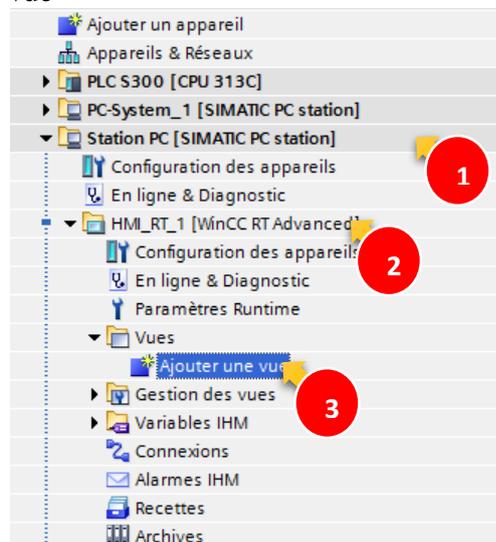


Figure III.39 : création d'un pupitre(3).

III.4.9.2. Création d'une connexion entre la station PC et l'automate S7-300

Pour établir une connexion entre la station Pc qui inclut le « HMI_RT » et l'automate, on procède comme suite:

On ouvert l'éditeur de configuration de connexions en appuyant sur « connexion » :

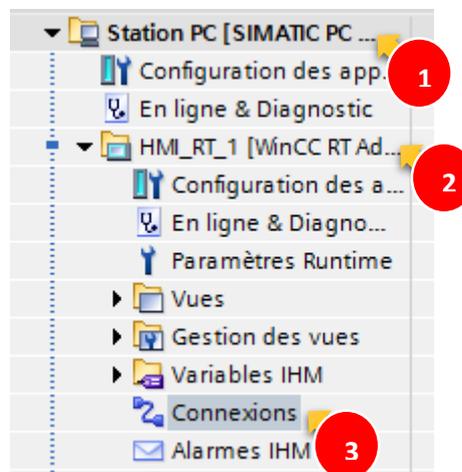


Figure III.40 : La connexion entre la station PC et l'automate S7-300.

Et puisque la connexion entre l'automate de la station FESTO et le PC de type « MPI/DP », on configure la connexion dans le TIA PORTAL de type « MPI/DP » :

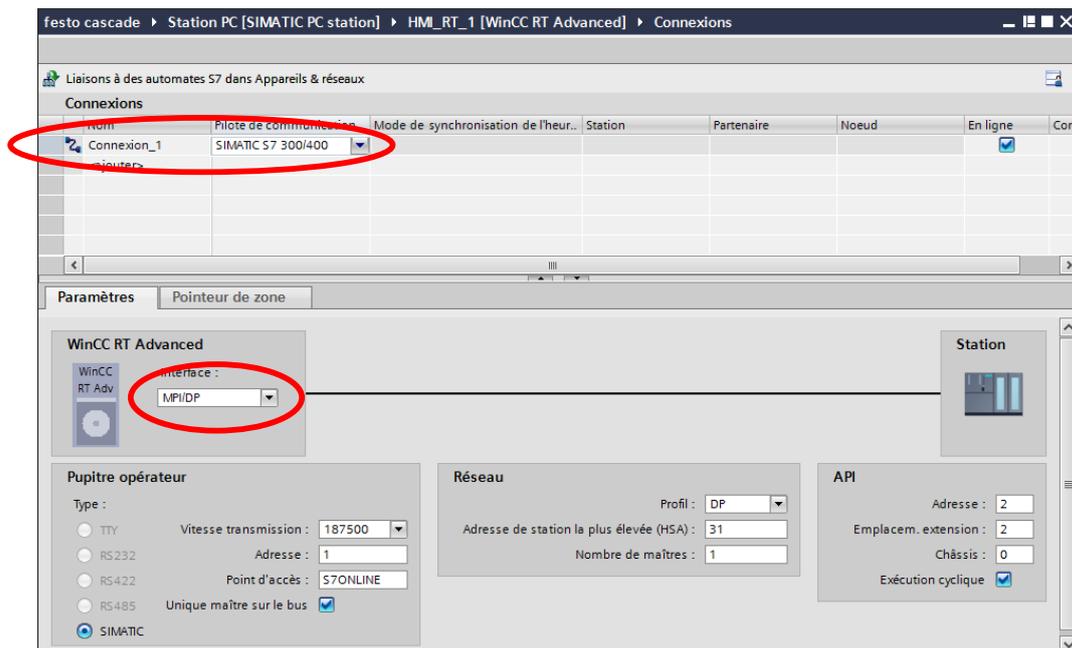


Figure III.41 : la configuration de la connexion Pc station/API (1)

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

Nom	Type de données	Connexion	Variable API	Adresse	Mod
Bloc de données_1_command...	Real	Connexion_1	<indéfini>	%DB1.DBD0	<acc
Bloc de données_1_command...	Real	Connexion_1	<indéfini>	%DB1.DBD100	<acc
Bloc de données_1_consign no...	Real	Connexion_1	<indéfini>	%DB1.DBD92	<acc
Bloc de données_1_débit normé	Real	Connexion_1	<indéfini>	%DB1.DBD112	<acc
Bloc de données_1_debit reel	Real	Connexion_1	<indéfini>	%DB1.DBD4	<acc
Bloc de données_1_erreur 1	Real	Connexion_1	<indéfini>	%DB1.DBD44	<acc
Bloc de données_1_erreur_nor...	Real	Connexion_1	<indéfini>	%DB1.DBD84	<acc
Bloc de données_1_ko1	Real	Connexion_1	<indéfini>	%DB1.DBD8	<acc

Figure III.42 : la configuration de la connexion Pc station/API (2).

III.4.10. Les vues de la supervision

Dans cette partie on présentera les huit vues créées pour la visualisation et la commande de la station FESTO.

1. La vue de la synoptique générale de la station

Dans cette vue on présente le schéma générale de la station FESTO avec les indicateurs et les boutons

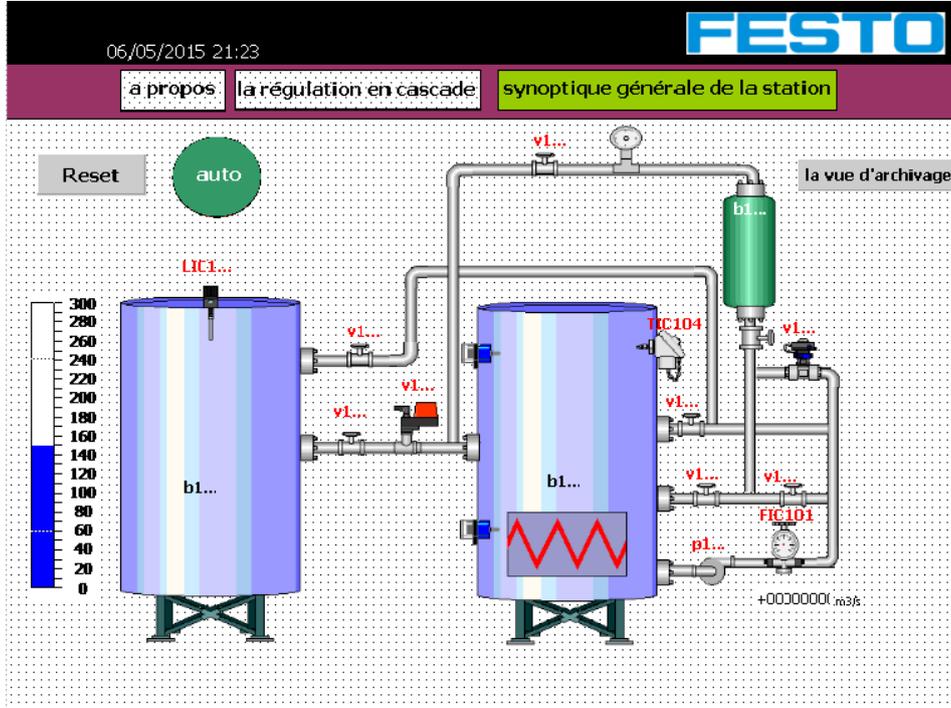


Figure III.43 : La vue de la synoptique générale de la station.

2. La vue « à propos »

Dans cette vue on a présenté la station FESTO.

Le système didactique du contrôle de processus FESTO

Le projet "Station Festo" est réalisé en tant qu'installation industrielle de régulations de processus composé de quatre boucles de régulations.

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.

Figure III.44 : La vue « à propos »

3. **La vue « régulation en cascade »** cette vue présente la synoptique de la régulation en cascade et présente aussi la vue PID pour le mode automatique et la vue d'identification pour le mode manuel.

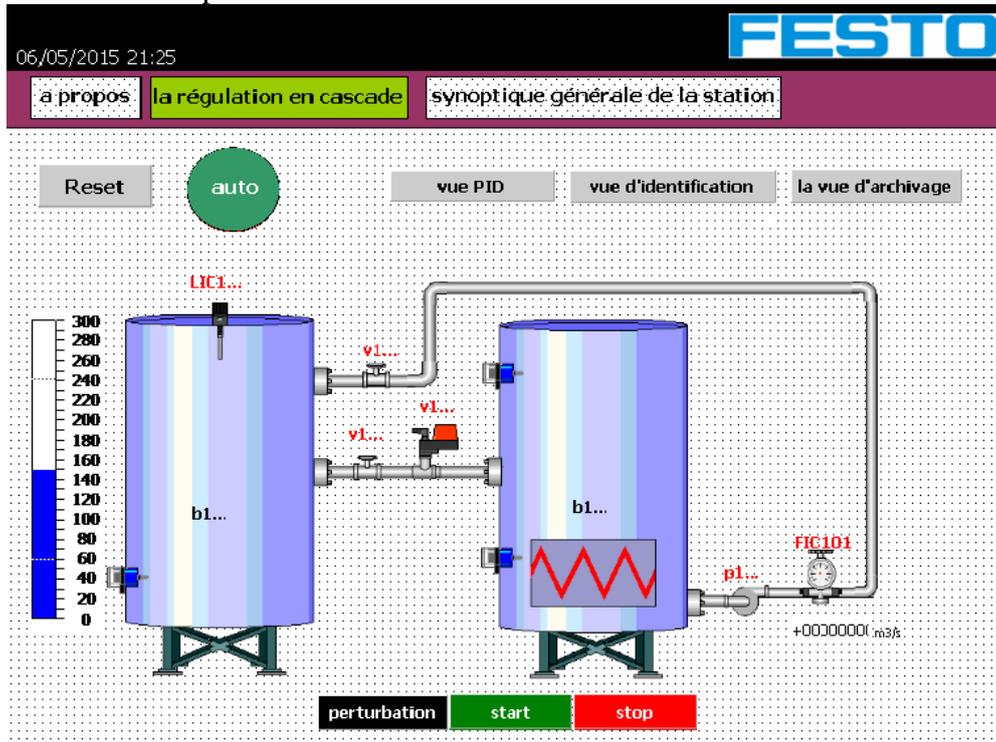


Figure III.45 : La vue « régulation en cascade ».

4. **La vue PID** cette vue présente la courbe et les indicateurs (de niveau, de débit et la valeur de réglage) et les champs d'entrée des paramètres de réglage.

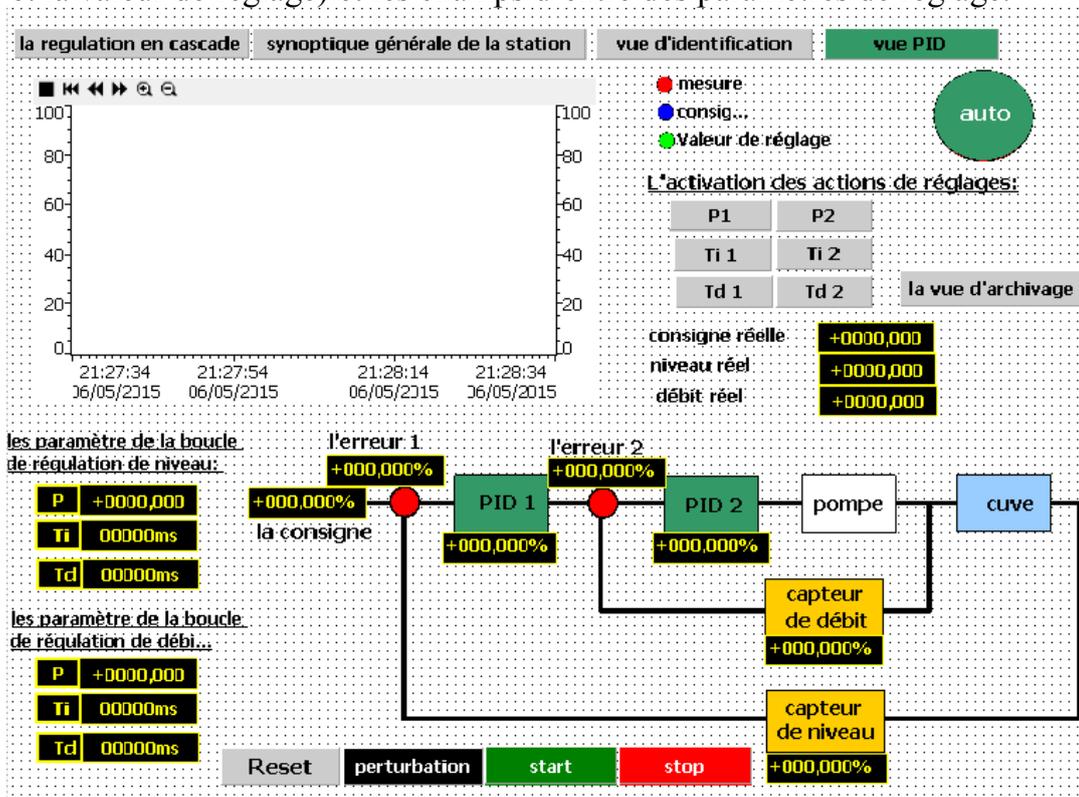


Figure III.46 : La vue PID.

5. La vue d'identification cette vue sert à choisir l'identification des deux boucles.

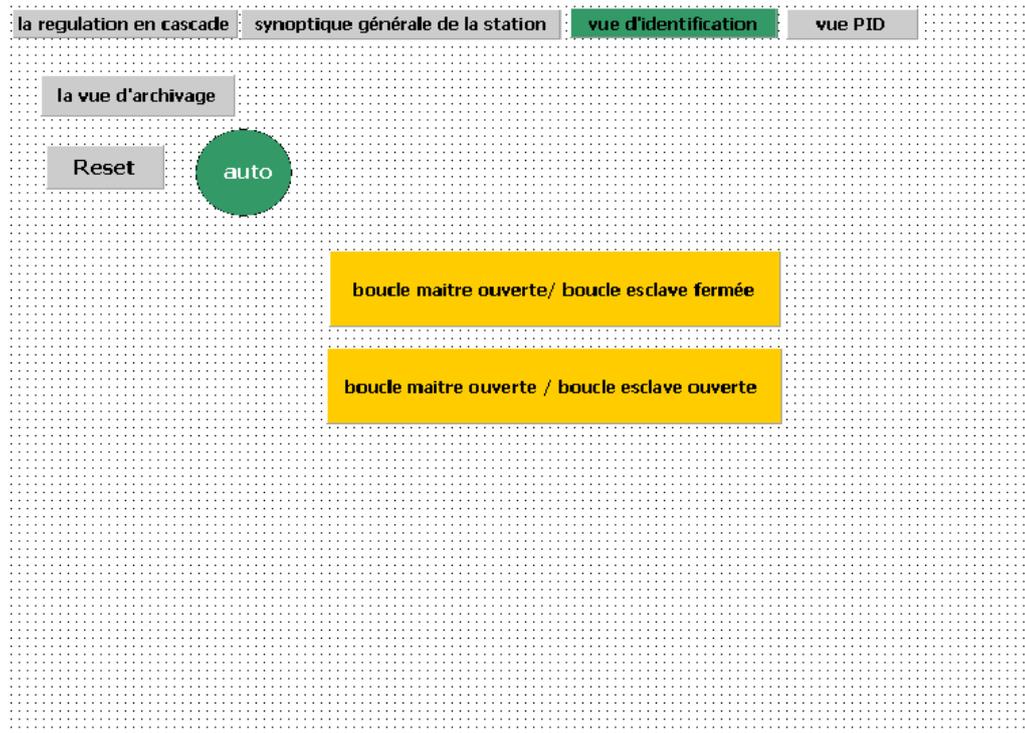


Figure III.47 : La vue d'identification.

6. Vue d'identification1

Cette vue sert à faire l'identification de la boucle intérieure.

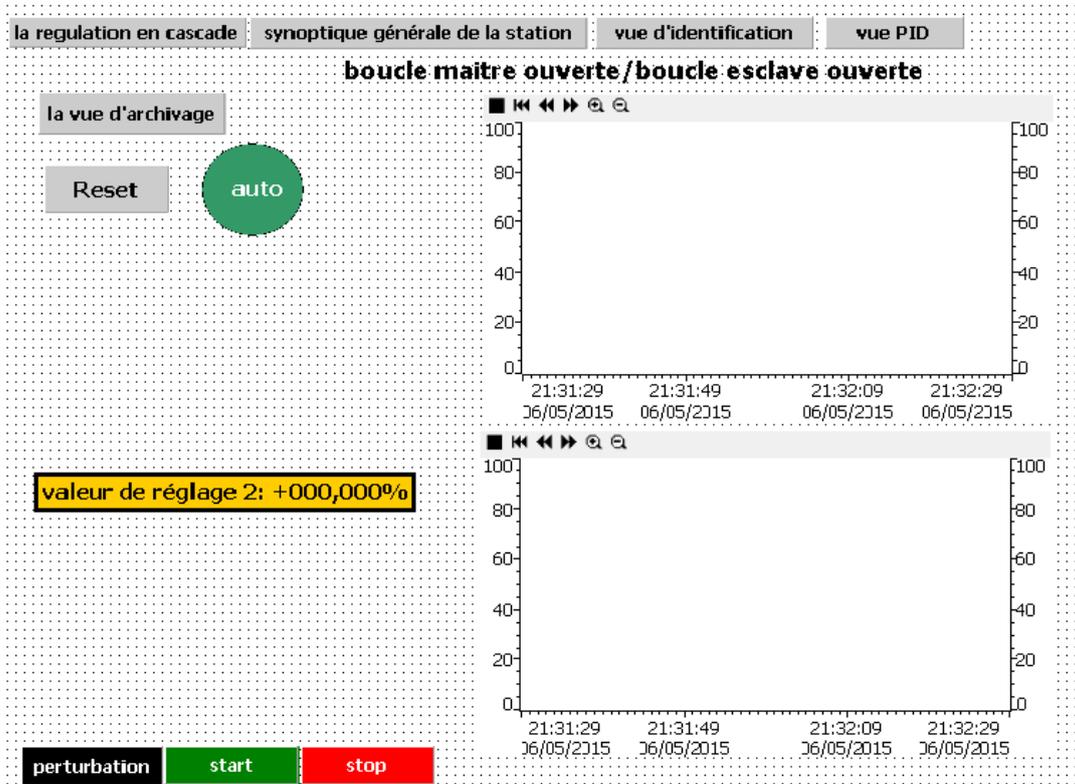


Figure III.48 : La vue de l'identification de la boucle esclave.

7. Vue d'identification 2

Cette vue sert à faire l'identification de la boucle extérieure en fermant la boucle intérieure.

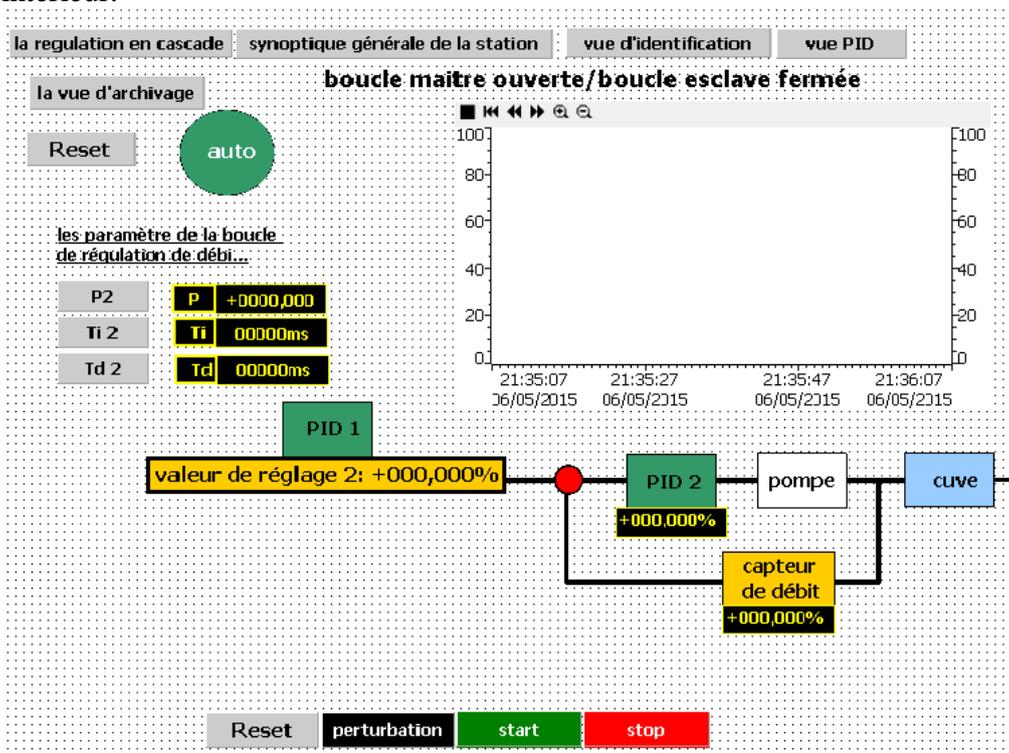


Figure III.49 : La vue de l'identification de la boucle extérieure.

8. Vue d'archivage

Cette vue sert à lancer, stopper et effacer les fichiers d'archivage



Figure III.50 : la vue d'archivage.

III.4.11. La simulation « PLCSIM » et « WINCC RT »

Pour lancer la simulation avec le PLCSim en procède comme suite :

1. On compile le matériel et le logiciel

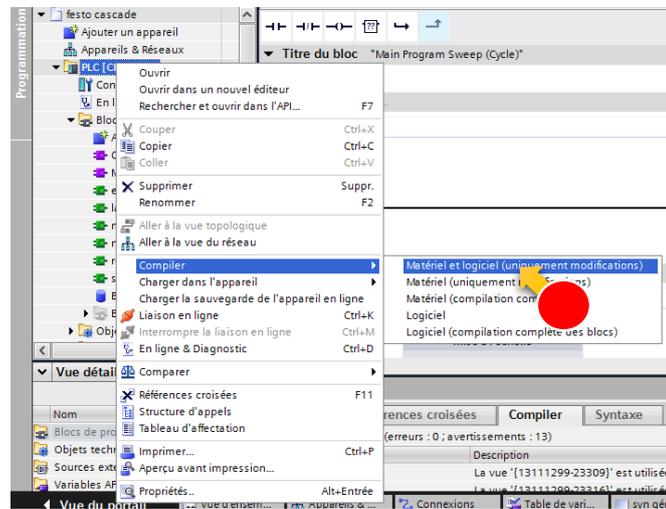


Figure III.51 : compilation du programme.

2. Après avoir compilé le programme on lance la simulation en cliquant sur l'icône « démarré la simulation » sur la barre d'outils :

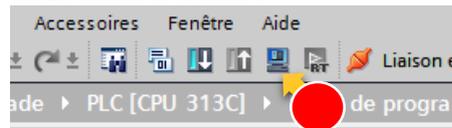


Figure III.52 : la simulation du programme.

3. On charge toute les fonctions :

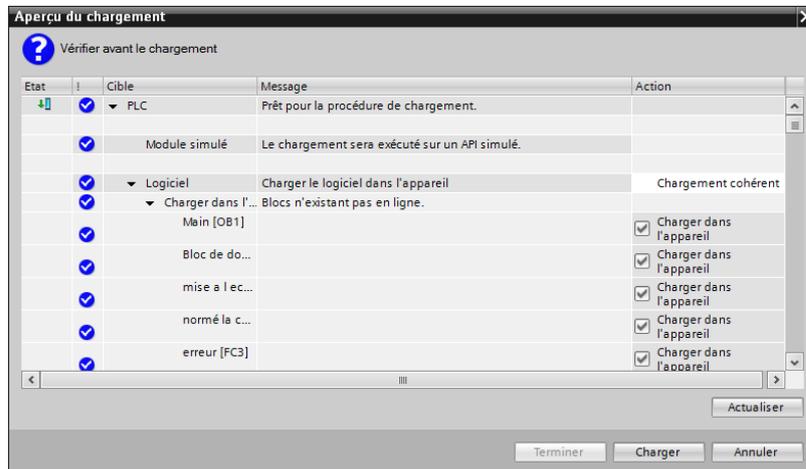


Figure III.53 : chargement du programme.

4. Et le simulateur PLCsim se lance

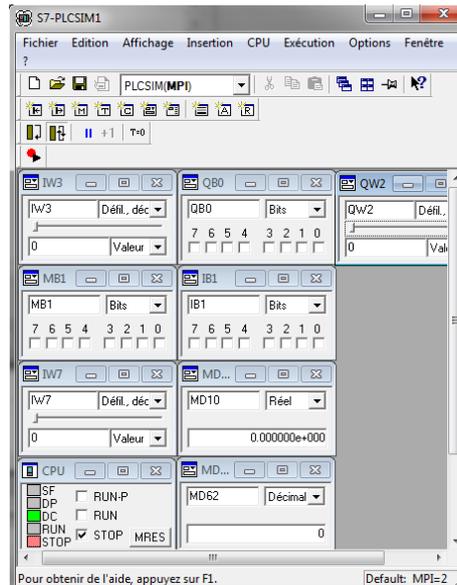


Figure III.54: lancement de la simulation.

- Et pour lancer la simulation de la supervision on clique sur l'icône de la barre d'outils suivante :



Figure III.55: La simulation wincc RT

Et le simulateur « wincc RT » se lance :

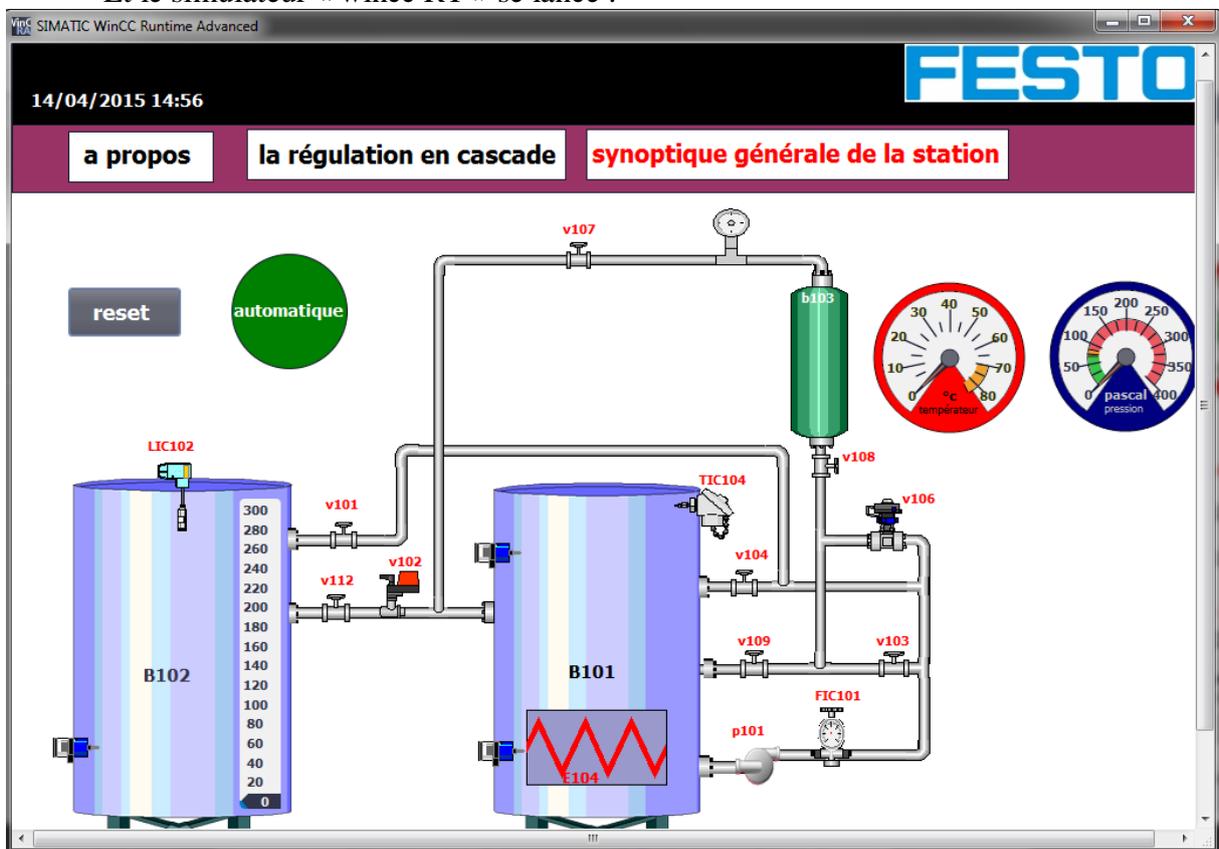


Figure III.56: lancement de la supervision.

III.4.12. Chargement de programme dans l'automate

Pour charger le programme précédent vers l'automate on procède comme suite :

1. en clique sur l'icône « chargé le programme » sur la barre d'outils :

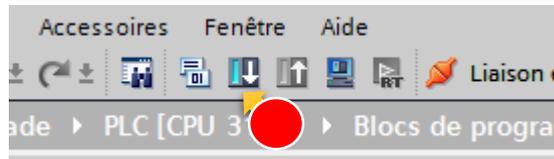


Figure III.57 : Chargement de programme dans l'automate (1).

2. on choisit le mode de liaison entre le Pc et l'automate dans notre projet on a une liaison MPI par un câble USB « PC adapter »

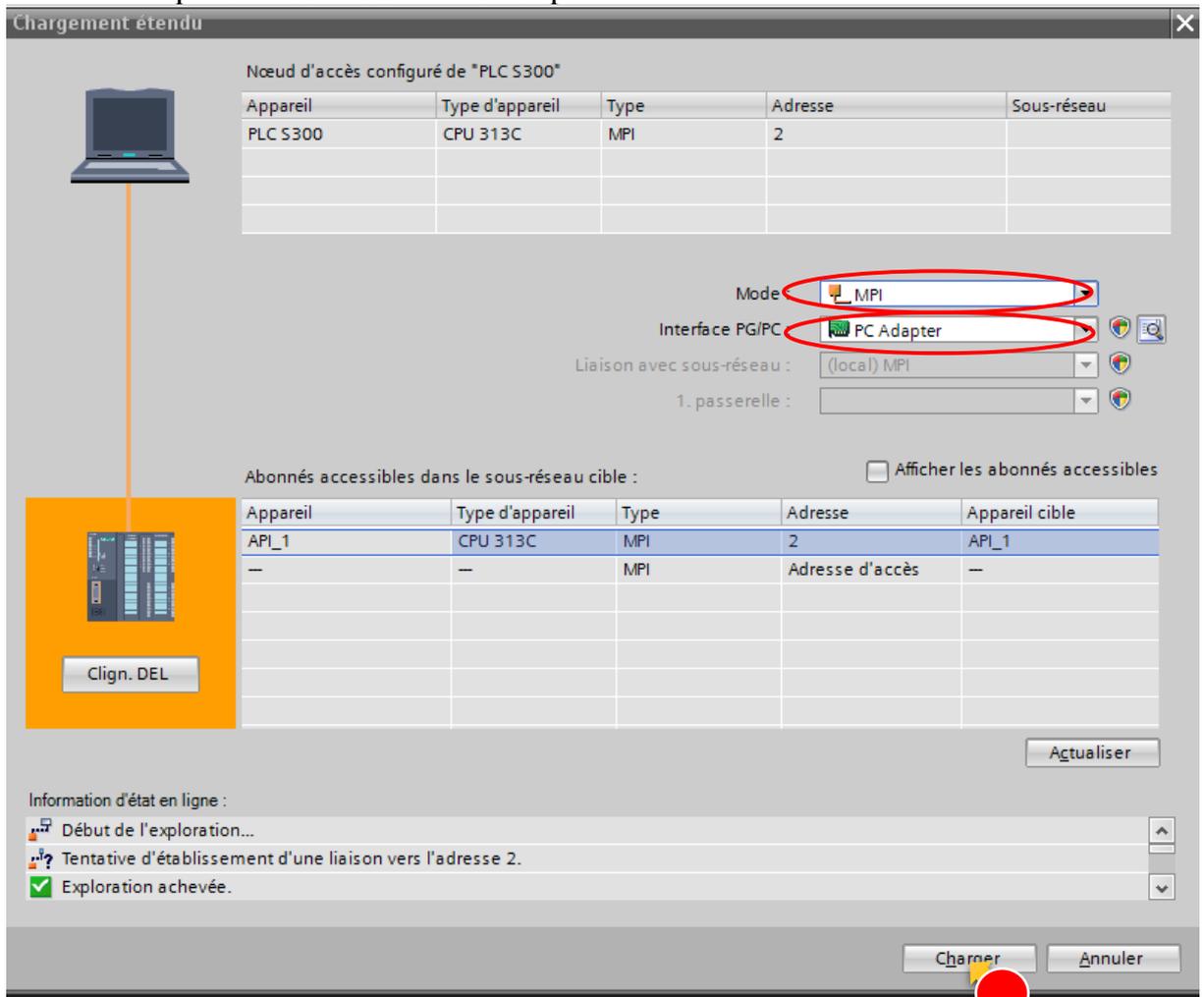


Figure III.58: chargement de programme dans l'automate (2).

3. puis on clique sur le bouton « charger »

III.4.13. Acquisition des données

Dans notre projet on s'intéresse à enregistrer les données venant de l'automate et spécialement les mesures de capteur de niveau et les mesures de capteur de débit, et on les archive dans un fichier texte afin de les utiliser pour la modélisation de système

- Pour archiver les données sur le TIA PORTAL on procède comme suite :
On ajoute un nouveau fichier d'archivage par le biais du navigateur du projet.

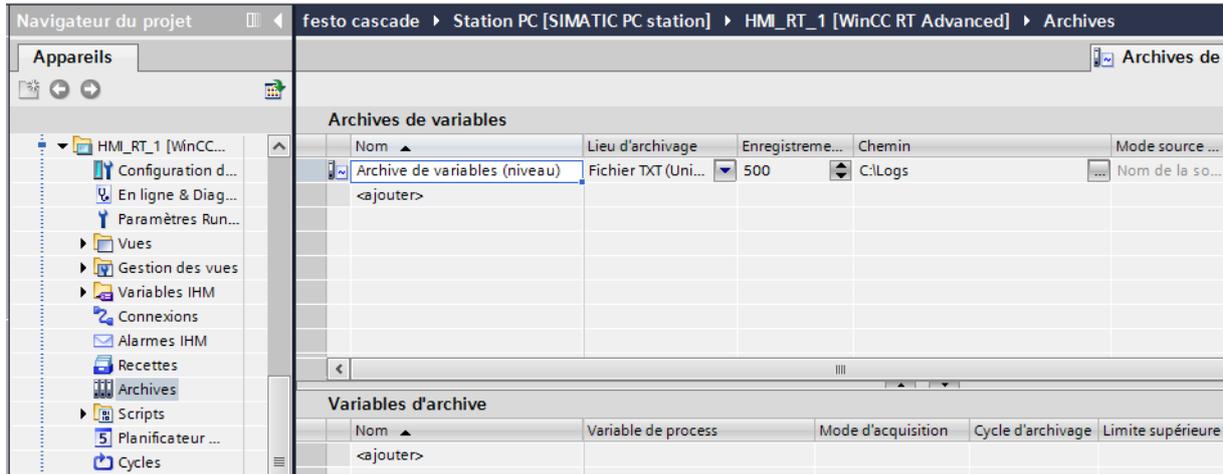


Figure III.59: l'archivage des mesures (1).

- On introduit le nom du fichier « archive de variables (niveau) » et on choisit le type de fichier et son emplacement et le nombre d'enregistrement
- Puis on affecte le variable IHM dans l'archive comme suite :

On sélectionne le variable a affecté et on clique sur « variable d'archive » et on introduit les paramètre nécessaire

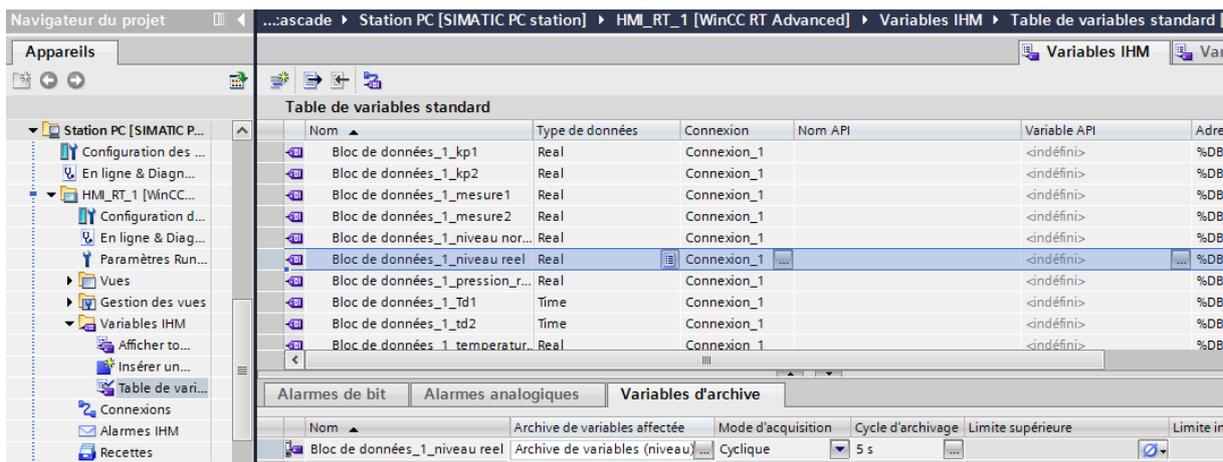
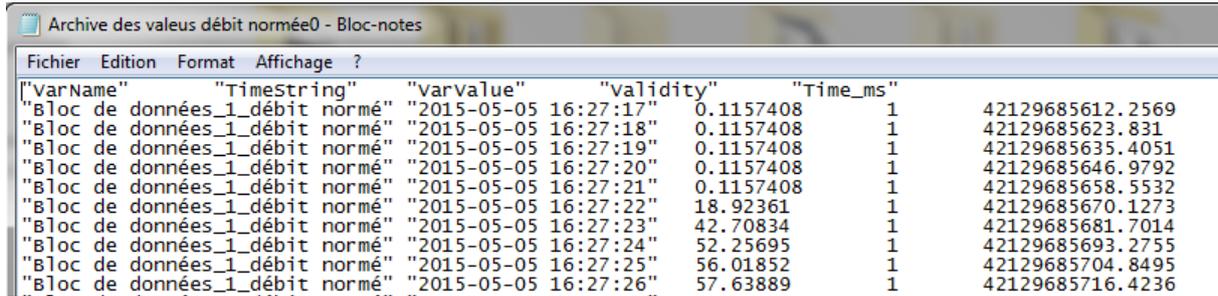


Figure III.60: l'archivage des mesures (2).

- Puis un fichier va être créé dans l'emplacement que nous avons choisi

Archive de variables (niveau) 21/04/2015 17:46 Document texte 77 Ko

- Et on trouve les mesures de capteur de niveau dans ce fichier sous cette forme



The screenshot shows a Notepad window titled "Archive des valeurs débit normée0 - Bloc-notes". The window contains a CSV file with the following data:

"varName"	"timeString"	"varValue"	"validity"	"Time_ms"
"Bloc de données_1_débit normé"	"2015-05-05 16:27:17"	0.1157408	1	42129685612.2569
"Bloc de données_1_débit normé"	"2015-05-05 16:27:18"	0.1157408	1	42129685623.831
"Bloc de données_1_débit normé"	"2015-05-05 16:27:19"	0.1157408	1	42129685635.4051
"Bloc de données_1_débit normé"	"2015-05-05 16:27:20"	0.1157408	1	42129685646.9792
"Bloc de données_1_débit normé"	"2015-05-05 16:27:21"	0.1157408	1	42129685658.5532
"Bloc de données_1_débit normé"	"2015-05-05 16:27:22"	18.92361	1	42129685670.1273
"Bloc de données_1_débit normé"	"2015-05-05 16:27:23"	42.70834	1	42129685681.7014
"Bloc de données_1_débit normé"	"2015-05-05 16:27:24"	52.25695	1	42129685693.2755
"Bloc de données_1_débit normé"	"2015-05-05 16:27:25"	56.01852	1	42129685704.8495
"Bloc de données_1_débit normé"	"2015-05-05 16:27:26"	57.63889	1	42129685716.4236

Figure III.61: les mesures de capteur de niveau.

On fait la même chose pour le mesure de débit

III.4.14. Conclusion

Nous avons pu voir à travers ce chapitre l'application de logiciel TIA PORTAL pour l'élaboration d'une commande avancée ainsi que sa supervision par l'interface HMI, nous avons su aussi la configuration utilisée pour l'acquisition des données sous forme de fichier text, qui peut être utilisé pour l'identification de système.

Conclusion générale

Dans notre projet nous avons programmé une commande avancée (régulation en cascade), par le logiciel 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, cet outil puissant nous a permis aussi d'enregistrer les données venant depuis l'automate dans une fichier text, afin de pouvoir les utiliser pour l'identification.

L'étude de la station didactique de contrôle des processus, FESTO nous a permis aussi de comprendre les différents systèmes de régulation et les instruments utilisés dans l'industrie, ainsi que leurs étalonnages, et de voir les différentes possibilités offertes par les automates programmables :simulation des systèmes continus, calcul numérique, identification des processus, commande et régulation des systèmes continus.

Notre contribution s'est portée sur l'implémentation d'un programme de régulation en cascade et la mise au point d'un système de supervision et de contrôle avec le logiciel TIA PORTAL, ainsi elle s'est portée sur l'acquisition des données depuis l'automate pour servir de base de départ à d'éventuels travaux pratiques et projets de fin d'études.

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

- L'ajout d'un clapet anti-retour à la sortie de la pompe pour éliminer la chute de niveau dans le réservoir.
- L'utilisation d'une vanne analogique (contrôlable) à la place de la vanne manuelle V109 pour qu'on puisse réguler le niveau en cascade avec le débit, par une vanne.

Et pour les perspectives, nous proposons les points suivants :

- La possibilité d'utilisé une identification par les méthodes récursive en utilisant MATLAB ou le logiciel PIM.
- La configuration d'une liaison entre MATLAB et l'automate programmable par le logiciel SIMATIC NET qui peut établir une liaison entre « l'opc server » de TIA PORTAL et « l'opc server » de MATLAB, ce qui nous permis d'appliquer des commande avancée (comme la commande adaptative, ou bien la commande par la logique flou...) sur le système à régler, et qui nous permis aussi d'implémenter une SBPA ou d'autre type des signaux (rampe, sinusoïdale.....), qui sont difficiles à générer par le TIA PORTAL

Bibliographie :

[1] www.siemens.com

[2] www.wikipedia.com

[3] J.HELMICH, «Compact workstation manual », FESTO, 2004

[4] Slim BEN SAOUD, «LES AUTOMATES PROGRAMMABLES INDUSTRIELS (API) »

[5] Frank D.Petruzella, « Programmable Logic Controllers », McGraw-Hill, 2005

[6] P.JARGOT, « Langages de programmation pour API, Norme IEC 61131-3 », Technique de l'ingénieur, Vol. S 8 030

[7] CHERGUI Younes, « Commande et supervision de la station FESTO PCS_COMPACT avec le TIA PORTAL », projet de fin d'étude, Ecole Nationale Polytechnique, 2012.

Liste des figures

Figure I.1: Maquette didactique FESTO.	2
Figure I.2 : Diagramme synoptique de l'installation didactique FESTO.	5
Figure I.3: Schéma de connexion API/Panneau de contrôle/Terminal analogique.....	6
Figure I.4: Pompe centrifuge.	7
Figure I.5: Processus de la vanne à bille.....	7
Figure I.6: Détecteurs de proximité capacitifs.	8
Figure I.7: Flotteur de sécurité anti débordement.	9
Figure I.8: Flotteur pour la détection des seuils de remplissage.	9
Figure I.9: Vanne proportionnelle.	10
Figure I.10: Capteur de niveau ultrasonique.....	10
Figure I.11: capteur de débit.	11
Figure I.12: Capteur de pression.	11
Figure I.13: Capteur de température.....	12
Figure I.14: Circuit relatif à la régulation de niveau en cascade avec le débit	15
Figure II.1: Automate compact.....	19
Figure II.2: Automate modulaire (Modicon)	19
Figure II.3:Automate modulaire (Siemens).....	19
Figure II.4: La structure interne d'un automate [1].....	20
Figure II.5: langages de programmation [5]	21
Figure II.6: Le GRAFCET	22
Figure II.7: Exemple 1 d'un programme en Ladder.....	23
Figure II.8: Exemple 2 d'un programme en Ladder.....	23
Figure II.9: Exemple d'un programme en Fonction Bloc.....	23
Figure II.10: Exemple d'un programme en Langage Structuré.	24
Figure II.11: Exemple d'un programme en Liste d'Instructions.[7]	24
Figure II.12: Présentation de la gamme de SIMATIC	25
Figure II.13:L'API S200.....	25
Figure II.14: L'API S300.....	26
Figure II.15: L'API S400.....	26
Figure II.16: La gamme SIMATIC C7.	27

Figure II.17: La gamme SIMATIC M7.	27
Figure II.18: L'API S7-313C.	28
Figure III.1: TIA PORTAL	41
Figure III.2: Système d'ingénierie.....	42
Figure III.3:Gestion des données	43
Figure III.4 : la vue du portail	44
Figure III.5 : la vue du projet.....	45
Figure III.6 : démarrage de TIA PRTAL V11	47
Figure III.7 : création de projet.....	48
Figure III.8: Insertion et configuration d'un automate (1).....	48
Figure III.9 : Insertion et configuration d'un automate (2).....	49
Figure III.10 : Insertion et configuration d'un automate (3).....	49
<i>Figure III.11 : la vue des appareils de l'éditeur Appareils et réseaux.....</i>	<i>50</i>
Figure III.12 : de la vue du réseau.	51
<i>Figure III.13 : Vue des appareils.....</i>	<i>52</i>
Figure III.14 : la création de la de variables	54
Figure III.15 : Tableau de référence	55
Figure III.16 : création d'un bloc de données globale.....	56
Figure III.17 : les variables du bloc de données.	56
Figure III.18 : l'exécution d'un OB de cycle de programme.....	57
Figure III.19 : L'ouverture d'un bloc d'organisation.....	58
Figure III.20 : Main OB1.	58
Figure III.21 : Présentation de l'éditeur de programmes	59
Figure III.22 : La fonction mise à l'échelle.....	60
Figure III.23 : La fonction « normalisation de consigne ».....	60
Figure III.24 : la fonction erreur.....	61
Figure III.25 : la fonction régulation.....	61
Figure III.26 : la fonction start/stop.....	61
Figure III.27 : La fonction reset.....	61
Figure III.28 : création d'une fonction	62
Figure III.29:La structure de l'éditeur.....	62
Figure III.30 : l'exécution du programme avec un OB d'alarme cyclique.....	63
Figure III.31 : création d'un OB d'alarme cyclique.....	64
Figure III.32 : l'insertion d'un bloc PID.	64

Figure III.33 : bloc PID.....	65
Figure III.34 : Schéma de principe de la régulation via un API.	66
Figure III.35: Schéma de principe du bloc PID.	67
Figure III.36 : « IHM » l'interface entre l'utilisateur et le processus.....	74
Figure III.37 : création d'un pupitre(1).	74
Figure III.38 : création d'un pupitre(2).	75
Figure III.39 : création d'un pupitre(3).	75
Figure III.40 : La connexion entre la station PC et l'automate S7-300.....	75
Figure III.41 : la configuration de la connexion Pc station/API (1).....	76
Figure III.42 : la configuration de la connexion Pc station/API (2).....	76
Figure III.43 : La vue de la synoptique générale de la station.	77
Figure III.44 : La vue « à propos »	77
Figure III.45 : La vue « régulation en cascade ».	78
Figure III.46 : La vue PID.	78
Figure III.47 : La vue d'identification.....	79
Figure III.48 : La vue de l'identification de la boucle esclave.	79
Figure III.49 : La vue de l'identification de la boucle extérieure.	80
Figure III.50 : la vue d'archivage.	80
Figure III.51 : compilation du programme.	81
Figure III.52 : la simulation du programme.	81
Figure III.53 : chargement du programme.	81
Figure III.54: lancement de la simulation.	82
Figure III.55: La simulation wincc RT	82
Figure III.56:lancement de la supervision.	82
Figure III.57 : Chargement de programme dans l'automate (1).	83
Figure III.58: chargement de programme dans l'automate (2).....	83
Figure III.59: l'archivage des mesures (1).	84
Figure III.60:l'archivage des mesures (2).	84
Figure III.61: les mesures de capteur de niveau.	85

Liste des tableaux

Tableau I.1: Caractéristiques techniques de l'installation.	4
Tableau I.2: Tableau d'adressage des entrées binaires.	12
Tableau I.3: Tableau d'adressage des sorties binaires.	13
Tableau I.4: Tableau d'adressage des entrées/sorties analogiques.	13
Tableau I.5: Tableau des configurations relatives aux différentes boucles de régulation.	15
Tableau I.6: Tableau des configurations relatives à la boucle de régulation en cascade.....	15
Tableau II.1 : Position du commutateur du mode de fonctionnement.	28
Tableau II.2: Les LEDs de visualisations.....	29
Tableau II.3: Caractéristiques techniques de la CPU-313C [1].....	33
Tableau II.4: Caractéristiques techniques du module E/S digitales [1].	35
Tableau II.5: Caractéristiques techniques du module E/S analogique [1].	37
Tableau II.6: Correspondance valeurs analogiques tension.	38
Tableau II.7: Correspondance valeurs analogiques courant [1].....	39
Tableau III.1 : la signification des différentes colonnes de l'onglet "Variables".	55
Tableau III.2: Entrées du bloc PID.	71
Tableau III.3: Sorties du bloc PID.	73

TABLES DES MATIERES

<i>Introduction général</i>	1
-----------------------------------	---

Chapitre I: Présentation du système didactique du contrôle de processus « FESTO » [3]

<i>I.1. INTRODUCTION</i>	2
<i>I.2. PRESENTATION DU SYSTEME DIDACTIQUE DU CONTROLE DE PROCESSUS FESTO</i>	2
<i>I.3. DESIGN ET FONCTION DES DIFFERENTS SYSTEMES CONSTITUANT L'INSTALLATION</i>	7
<i>I.3.1. LA POMPE</i>	7
<i>I.3.2. LE PROCESSUS DE LA VANNE A BILLE</i>	7
<i>I.3.3. ELEMENTS DE CONTROLE DE NIVEAU TOUT OU RIEN</i>	8
<i>I.3.4. LA VANNE PROPORTIONNELLE</i>	10
<i>I.3.5. LE CAPTEUR DE NIVEAU ULTRASONIQUE</i>	10
<i>I.3.6. LE CAPTEUR DE DEBIT</i>	11
<i>I.3.7. LE CAPTEUR DE PRESSION</i>	11
<i>I.3.8. LE CAPTEUR DE TEMPERATURE</i>	12
<i>I.4. LES SYSTEMES DE REGULATION EN BOUCLE FERMEE</i>	14
<i>I.5. SYSTEME DE REGULATION DE NIVEAU EN CASCADE AVEC LE DEBIT</i>	15
<i>I.6. CONCLUSION</i>	16

Chapitre II : Généralité sur les automates programmables

<i>II.1. INTRODUCTION</i>	18
<i>II.2. GENERALITE SUR LES AUTOMATES</i>	18
<i>II.2.1. HISTORIQUE</i>	18
<i>II.2.2. API – DEFINITION</i>	18
<i>II.3. ARCHITECTURE DES AUTOMATES [1]</i>	18
<i>II.3.1. ASPECT EXTERIEUR :</i>	18
<i>II.3.2. STRUCTURE INTERNE</i>	20
<i>II.4. FONCTIONS REALISEES</i>	20
<i>II.5. LANGAGES DE PROGRAMMATION</i>	21
<i>II.5.1. LES LANGAGES GRAPHIQUES</i>	22
<i>1) LE GRAFCET</i>	22
<i>2) LADDER DIAGRAM</i>	22
<i>3) BLOC DE FONCTION</i>	23

II.5.2. LES LANGAGES TEXTUELS.....	24
1) TEXTE STRUCTURE.....	24
2) LISTE D'INSTRUCTIONS.....	24
II.6. PRESENTATION DE LA GAMME SIMATIC DE SIEMENS [1].....	25
II.6.1. LES DIFFERENTES VARIANTES DANS LA GAMME SIMATIC	25
1. SIMATIC S7.....	25
2. SIMATIC C7	26
3. SIMATIC M7.....	27
II.6.2. DESCRIPTION DE L'AUTOMATE PROGRAMMABLE S7-313	28
II.6.2.1. LA CPU.....	28
II.6.2.2. LE MODULE E/S DIGITALES INTEGRE (DI24/DO16).....	33
II.6.2.3. LE MODULE E/S ANALOGIQUES INTEGRE (AI5/AO2) :	35
II.7. CONCLUSION	39

Chapitre III : La programmation et la supervision sur le TIA Portal

III.1. INTRODUCTION.....	41
III.2.2. CONCEPT D'INGENIERIE :	42
III.2.3. GESTION DES DONNEES	42
III.2.4. LES AVANTAGES DE TRAVAILLER AVEC LE PORTAIL TIA :	43
III.3. VUES DU PORTAIL TIA.....	44
III.3.1. NAVIGATION DANS LE PORTAIL TIA.....	44
III.3.2. VUE DU PORTAIL.....	44
III.3.3. VUE DU PROJET	45
III.4. LA PROGRAMMATION DE STATION FESTO	47
III.4.1. CREATION DU PROJET	47
III.4.2. INSERTION ET CONFIGURATION D'UN AUTOMATE :	48
III.4.3. PRESENTATION DE L'EDITEUR D'APPAREILS ET DE RESEAUX :	50
III.4.3.1. VUE DU RESEAU	50
III.4.3.2. VUE DES APPAREILS.....	52
III.4.3.3. LES MODULES DE SIGNAUX.....	53
III.4.4. CREATION DE LA TABLE DE VARIABLES	53
III.4.5. CREATION D'UN BLOC DE DONNEE GLOBALE	55
III.4.6. CREATION D'UN BLOC D'ORGANISATION « MAIN OBI »	57

<i>III.4.6.1. PROGRAMME UTILISATEUR</i>	57
<i>III.4.3.1. BLOCS D'ORGANISATION</i>	57
<i>III.4.7. CREATION D'UNE FONCTION</i>	62
<i>III.4.8. CREATION DU BLOC D'ORGANISATION POUR LE REGULATEUR PID</i> :.....	63
<i>III.4.8.1. PRINCIPE DE LA REGULATION</i>	66
<i>III.4.8.2. REGULATION CONTINUE AVEC LE BLOC « CONT_C »</i>	66
<i>III.4.9. CREATION D'UNE INTERFACE IHM</i>	74
<i>III.4.9.1. CREATION D'UN PUPITRE OPERATEUR AVEC VUE IHM</i>	75
<i>III.4.9.2. CREATION D'UNE CONNEXION ENTRE LA STATION PC ET L'AUTOMATE S7-300</i>	75
<i>III.4.10. LES VUES DE LA SUPERVISION</i>	77
1. <i>LA VUE DE LA SYNOPTIQUE GENERALE DE LA STATION</i>	77
2. <i>LA VUE « A PROPOS »</i>	77
3. <i>LA VUE « REGULATION EN CASCADE »</i>	78
4. <i>LA VUE PID</i>	78
5. <i>LA VUE D'IDENTIFICATION</i>	79
6. <i>VUE D'IDENTIFICATION 1</i>	79
7. <i>VUE D'IDENTIFICATION 2</i>	80
8. <i>VUE D'ARCHIVAGE</i>	80
<i>III.4.11. LA SIMULATION « PLCSIM » ET « WINCC RT »</i>	81
<i>III.4.12. CHARGEMENT DE PROGRAMME DANS L'AUTOMATE</i>	83
<i>III.4.13. ACQUISITION DES DONNEES</i>	84
<i>III.4.14. CONCLUSION</i>	85
<i>Conclusion générale</i>	86
<i>Bibliographie</i>	87