

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIFIQUE**

ECOLE NATIONALE SUPERIEURE POLYTECHNIQUE



**Département Génie Electrique
Spécialité Automatique**

Projet de fin d'études

**Pour l'obtention du diplôme
D'Ingénieur d'Etat en Automatique**

THÈME

**Commande et supervision de la station FESTO
PCS_COMPACT à l'aide de STEP7 et WinCC**

Réalisé par :

**SAHAR Belkacem Amine
SOUCHANE Amine**

Proposé et dirigé par :

**M^r. M. BERKOUK
M^r. R. ILLOUL**

Juin 2009

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

REMERCIEMENTS

Nous tenons tout d'abord à exprimer nos sincères remerciements à nos encadreurs Pr. BERKOUK ainsi que Mr. ILLOUL pour nous avoir encadrés durant ce projet et nous avoir conseillés tout au long de notre travail.

Nous remercions aussi les membres du jury pour l'honneur qu'ils nous ont fait en évaluant notre modeste travail.

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

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

DEDICACES

Je dédie ce modeste travail à ma mère, mon père ainsi qu'à mon frère Salim qui m'ont soutenu et aidé tout au long de mon parcours et sans qui je n'en serais pas la

A ma défunte tante qui j'espère, de la où elle est me regarde avec fierté

A Melissa pour son soutien, son aide et sa présence

A mon cher ami et camarade Belkacem avec qui j'ai travaillé avec grand plaisir

A mes amis de toujours qui m'ont tant supporté

A tous mes camarades de notre chère école avec qui j'ai passé de bons moments et appris beaucoup de choses

A tous ceux qui ont contribué de près ou de loin à notre travail

A tous ceux que je n'ai pas cités et qui sont présents dans mes pensées

Amine SOUCHANE

DEDICACES

*Je dédie ce modeste travail tout d'abord à mon père
qui m'a tout appris, tant donné sans rien demandé en
retour.*

*A ma très chère mère sans laquelle je ne serais pas
l'homme que je suis.*

*A mon grand frère yacine qui a su m'épauler dans les
moments difficiles.*

*A mon ami et binôme de la vie amine sur qui j'ai
toujours pu compter.*

*A mes amis et camarades avec qui j'ai passé de bons
moments.*

*A tous ceux qui ont contribué de loin ou de prêt à notre
travail.*

Belkacem amine SAHAR

Table des matières

INTRODUCTION GENERALE.....	1
CHAPITRE I : Les automates programmables industriels	
Introduction.....	3
Partie A : Généralité sur les automates programmable industriels.....	3
I.A.1 Place des API dans les systèmes automatisés de production	3
I.A.2 Définitions générales.....	4
I.A.3 Architecture générale des API.....	5
I.A.3.1 Module de l'alimentation.....	5
I.A.3.2 Unité centrale.....	6
I.A.3.2.1 Le processeur.....	6
I.A.3.2.2 Les mémoires.....	6
I.A.3.3 Modules d'entrées/sorties.....	7
I.A.3.4 Modules de communication.....	8
I.A.3.5 Auxiliaires.....	8
I.A.5 Langages de programmation.....	8
I.A.5.1 Introduction.....	8
I.A.5.2 Les langages graphiques.	9
I.A.5.2.1 Le GRAFCET.....	9
I.A.5.2.2 Ladder Diagram.....	10
I.A.5.2.3 Blocs de fonctions.....	11
I.A.5.3 Les langages textuels.....	11
I.A.5.3.1 Texte structure.....	11
I.A.5.3.2 Liste d'instructions.....	12
I.A.5.4	
Conclusion.....	13

Partie B : Les automates programmable industriels Siemens	14
I.B.1 Présentation de la gamme SIMATIC Siemens.....	14
I.B.1.1 Historique.....	14
I.B.1.2 Les différentes variantes dans la gamma SIMATIC.....	14
I.B.2 Description de l'automate programmable S7-313.....	18
I.B.2.1 La CPU.....	19
I.B.2.2 Le module E/S digitales intégré.....	24
I.B.2.3 Le module E/S analogiques intégré.....	25
Conclusion.....	30

CHAPITRE II : STEP7, logiciel de programmation des API Siemens

Introduction.....	32
II.1 Présentation du logiciel STEP7.....	32
II.2 Programmation à l'aide du logiciel STEP7.....	33
II.2.1 Création d'un projet SIMATIC Manager.....	34
II.2.1.1 Configuration matérielle.....	35
II.2.1.2 Elaboration des programmes S7.....	40
a. Organisation par blocs des programmes S7.....	40
b. Edition de la table des mnémoniques.....	43
c. Différents langages disponibles dans le STEP7.....	46
II.2.2 Simulation des modules.....	53
II.3 Régulation PID des processus à base d'automates SIMATIC S7.....	56
II.3.1 Principe de la régulation.....	56
II.3.2 Concept de la régulation.....	57
II.3.3 Régulation continue avec le bloc FB41 « CONT_C ».....	64
II.3.4 Paramétrage du régulateur PID FB41 avec STEP7.....	65
Conclusion.....	65

CHAPITRE III : WinCC, logiciel de conception des interfaces HMI

Introduction.....	67
III.1 Présentation de WinCC.....	67
III.1.1 Description générale.....	67
III.1.2 Fonctionnement du WinCC.....	68
a. Logiciel de configuration.....	68
b. Logiciel runtime.....	69
III.2 Etapes de conception d'un projet sous WinCC.....	69
a. Création d'un projet.....	70
b. Ajout d'un pilote API.....	70
c. Définition des variables externes et internes.....	72
d. Création d'une vue de process.....	76
e. Dynamisation de la vue de process.....	78
f. Simulation de la vue de process.....	79
III.3 Affichage des valeurs de process.....	80
III.4 Configuration des alarmes.....	82
III.5 Création de fonctions et d'actions (Global Script)	84
III.5.1 VBScript.....	85
III.5.2 C-Script.....	86
a. Simulation d'un circuit RC.....	86
b. Régulation de la température dans une chaudière.....	87
Conclusion.....	89

CHAPITRE IV : Régulation niveau, débit, pression, et température sur le système didactique de contrôle de processus FESTO

Introduction.	91
Partie A : Description du système didactique du contrôle de processus FESTO.....	91

IV.A.1. Présentation du système didactique du contrôle de processus FESTO.....	91
IV.A.2. Design et fonction des différents systèmes constituant l'installation.....	96
IV.A.2.1. La pompe.	96
IV.A.2.2. Le Processus de vanne à billes.....	97
IV.A.2.3. Eléments de contrôle de niveau tout ou rien.	98
1. Détecteur de proximité tout ou rien.	98
2. Flotteur de sécurité anti débordement.	99
3. Flotteur pour la détection des seuils de remplissage.	99
IV.A.2.4. La vanne proportionnelle.	100
IV.A.2.5. le capteur de niveau.	101
IV.A.2.6. Le capteur de débit.....	101
IV.A.2.7. Le capteur de pression.	102
IV.A.2.8. Le capteur de température.....	102
Partie B : Régulation des systèmes continus.....	105
IV.B.1. Les systèmes de contrôle en boucle fermée.....	105
IV.B.1.1. Le système de régulation de débit.....	106
IV.B.1.1.1 Régulation de débit avec la pompe comme actionneur.....	107
a. Description.....	107
b. Identification du système pompe-débit.....	108
c. Synthèse du régulateur PI.....	109
IV.B.1.1.2 Régulation de débit avec la vanne comme actionneur.....	111
a. Description.....	111
b. Identification du système vanne-débit.....	112
c. Synthèse du régulateur PI.	113
IV.B.1.2. Le système de régulation de niveau.....	115
IV.B.1.2.1 Régulation de niveau analogique.....	116
a. Description.....	116
b. Modélisation du système niveau-pompe.....	117
c. Synthèse du régulateur P.....	118

IV.B.1.3. Le système de régulation de pression.	120
IV.B.1.3.1 Régulation de pression avec la pompe comme actionneur.....	121
a. Description.....	121
b. Régulation PID par la méthode de Ziegler-Nichols.....	122
IV.B.1.3.2 Régulation de pression avec la vanne comme contrôleur.....	124
a. Description.....	124
b. Identification du système vanne-pression.....	125
c. Synthèse du régulateur PI.....	126
IV.B.1.4. Le système de régulation de température.....	127
IV.B.2. Description de l'application.....	130
IV.B.2.1 Description des programmes STEP7.....	130
1. Bloc d'organisation cyclique OB1.....	135
a. Bloc de mise à l'échelle FC10.....	135
b. Bloc fonctionnel pour le choix de régulation FC11.....	136
c. Bloc fonctionnel pour l'arrêt FC12.....	136
2. Bloc d'organisation d'alarmes OB35.....	136
IV.B.2.2 Description des programmes WinCC.....	136
Conclusion	139
CONCLUSION GENERALE.....	140
BIBLIOGRAPHIE.....	141

Liste des figures

Figure I.A.1: Système automatisé de production.....	5
Figure I.A.2: Architecture interne d'un API.....	5
Figure I.A.3 : Le GRAFCET.....	9
Figure I.A.4 : Exemple 1 d'un programme en Ladder.....	10
Figure I.A.5 : Exemple 2 d'un programme en Ladder.....	11
Figure I.A.6 : Exemple d'un programme en Fonction Bloc.....	11
Figure I.A.7 : Exemple d'un programme en Langage Structuré.....	12
Figure I.A.8 : Exemple d'un programme en Liste d'Instructions.....	12
Figure I.B.1 : Présentation de la gamme SIMATIC.....	15
Figure I.B.2: L' API S200.....	15
Figure I.B.3: L' API S300.....	16
Figure I.B.4: L' API S400.....	16
Figure I.B.5 : La gamme SIMATIC C7.....	17
Figure I.B.6 : La gamme SIMATIC M7.....	18
Figure I.B.7 : L' API S7-313C.....	18
Figure II.1 : Création d'un nouveau projet (1).....	34
Figure II.2 : Création d'un nouveau projet (2).....	34
Figure II.3 : Choix de la station SIMATIC de travail.....	35
Figure II.4 : Organigramme de la configuration matérielle.....	36
Figure II.5 : Editeur des configurations matérielles.....	37
Figure II.6 : Catalogue SIMATIC 300.....	37
Figure II.7 : Sélection des modules (1).....	38

Figure II.8 : Sélection des modules (2).....	39
Figure II.9 : Paramétrage de la CPU.....	39
Figure II.10 : Paramétrage du module E/S analogiques.....	40
Figure II.11 : Organisation par blocs des programmes S7 (1).....	41
Figure II.12 : Organisation par blocs des programmes S7 (2).....	42
Figure II.13 : Insertion d'un nouveau bloc d'organisation.....	42
Figure II.14 : Insertion d'un nouveau fichier source.....	43
Figure II.15 : Editeur des mnémoniques.....	43
Figure II.16 : Editeur de programme CONT, LIST, LOG.....	46
Figure II.17 : Choix du langage de programmation.....	47
Figure II.18 : Création d'un fichier source SCL.....	48
Figure II.19 : Choix du type de fichier SCL (1).....	48
Figure II.20 : Choix du type de fichier SCL (2).....	49
Figure II.21 : Le code source de l'exemple de résolution d'un système d'équations.....	50
Figure II.22 : Le synoptique de l'exemple de remplissage.....	51
Figure II.23 : Editeur S7-GRAPH.....	52
Figure II.24 : Le grafcet de l'exemple de remplissage.....	53
Figure II.25 : Interface de simulation de PLCSIM.....	54
Figure II.26 : Circuit RC.....	55
Figure II.27 : Résultat simulation du circuit RC.....	56
Figure II.28 : Schéma de principe de la régulation via un API.....	57
Figure II.29 : Schéma de principe du bloc SFB41.....	58
Figure II.30 : Interface de paramétrage PID.....	64

Figure III.1 : Schéma de communication entre le WinCC et l'API.....	67
Figure III.2 : Structure interne du WinCC.....	68
Figure III.3 : Boite de dialogue de création de projets.....	70
Figure III.4 Explorateur WinCC.....	70
Figure III.5 : Liste des différents types de pilotes de communication.....	71
Figure III.6 : Ajout du pilote d'un API.....	71
Figure III.7 : Création de la liaison MPI.....	72
Figure III.8 : Création d'une variable externe.....	74
Figure III.9 : Adressage de la variable externe.....	74
Figure III.10 : Création d'une variable interne.....	75
Figure III.11: Choix du type et du nom de la variable.....	76
Figure III.12 : création d'une vue grâce au Graphics Designer (GD).....	76
Figure III.13 : Vue détaillée de l'interface de GD.....	77
Figure III.14 : Liste des composants de la bibliothèque.....	77
Figure III.15 : Vue de process obtenue par le GD.....	78
Figure III.16 : Dynamisation d'un objet.....	78
Figure III.17 : Configuration de Graphics-Runtime.....	79
Figure III.18 : Vue de process.....	79
Figure III.19 : Différentes parties du Tag Logging.....	80
Figure III.20 : Etapes de la création de l'archive.....	81
Figure III.21 : Paramétrage de la fenêtre de courbe et de tableau.....	81
Figure III.22 : Courbe et tableau au mode Runtime.....	82
Figure III.23 : L'éditeur Alarm Logging.....	82

Figure III.24 : Configuration des propriétés des alarmes.....	83
Figure III.25 : Configuration des alarmes de seuils.....	84
Figure III.26 : Fenêtre d’alarmes au runtime.....	85
Figure III.27 : Global Script.....	87
Figure III.28 : Ajout d’une action en C.....	87
Figure III.29 : Simulation d’un circuit RC.....	87
Figure III.30 : Schéma bloc du système de régulation.....	88
Figure III.31 : Programme de la régulation de température.....	88
Figure III.32 : Régulation de température.....	89
Figure IV.A.1 : Maquette didactique FESTO.....	91
Figure IV.A.3 : Diagramme synoptique de l’installation didactique FESTO.....	94
Figure IV.A.4 : Schéma de connexion API/Panneau de contrôle/Terminal analogique.....	96
Figure IV.A.5 : Pompe centrifuge.....	97
Figure IV.A.6 : Processus de la vanne à bille.....	98
Figure IV.A.7 : Détecteurs de proximité capacitifs.....	98
Figure IV.A.8 : Flotteur de sécurité anti débordement.....	99
Figure IV.A.9 : Flotteur pour la détection des seuils de remplissage.....	100
Figure IV.A.10 : Vanne proportionnelle.....	100
Figure IV.A.11 : Capteur de niveau ultrasonique.....	101
Figure IV.A.12 : capteur de débit.....	102
Figure IV.A.13 : Capteur de pression.....	102
Figure IV.A.14 : Capteur de température.....	103
Figure IV.B.1 : Circuit relatif à la régulation de niveau.....	106

Figure IV.B.2 : Diagramme électrique de principe de la régulation de niveau analogique...	107
Figure IV.B.3 : Réponse indicielle théorique en boucle fermée de niveau.....	109
Figure IV.B.4 : Réponse indicielle réelle en boucle fermée de niveau.....	110
Figure IV.B.5 : Réponse indicielle réelle en boucle fermée de niveau avec action I.....	110
Figure IV.B.6 : Circuit relatif à la régulation de débit par le biais de la pompe.....	111
Figure IV.B.7 : Circuit relatif à la régulation de débit par le biais de la vanne proportionnelle.....	111
Figure IV.B.8 : Diagramme électrique de principe de la régulation de débit avec la pompe comme contrôleur.....	112
Figure IV.B.9 : Réponse indicielle du débit (pompe) en boucle ouverte.....	113
Figure IV.B.10 : Calcul de la constante de temps graphiquement.....	114
Figure IV.B.11 : Méthode du lieu d'Evans pour le calcul des paramètres du régulateur PI...	114
Figure IV.B.12 : Simulation de la boucle de régulation de débit (pompe).....	115
Figure IV.B.13 : Réponse indicielle du débit réelle (pompe) en boucle fermée.....	115
Figure IV.B.14 : Diagramme électrique de principe de la régulation de débit avec la vanne proportionnelle comme contrôleur.....	116
Figure IV.B.15 : Réponse indicielle du débit (vanne analogique).....	117
Figure IV.B.16 : Calcul de la constante de temps graphiquement.....	117
Figure IV.B.17 : Méthode du lieu d'Evans pour le calcul des paramètres du régulateur PI...	118
Figure IV.B.18 : Réponse indicielle théorique en boucle fermée (débit-vanne).....	119
Figure IV.B.19 : Réponse indicielle réelle en boucle fermée (débit vanne).....	119
Figure IV.B.20 : Circuit relatif à la régulation de pression avec la pompe comme actionneur.....	120

Figure IV.B.21 : Circuit relatif à la régulation de pression avec la pompe comme actionneur.....	120
Figure IV.22 : Diagramme électrique de principe de la régulation de pression avec la pompe comme contrôleur.....	121
Figure IV.B.23 : Réponse en régime de pompage de la pression (pompe).....	122
Figure IV.B.24 : Réponse indicielle en boucle ouverte de la pression (pompe).....	123
Figure IV.B.25 : Réponse indicielle en boucle fermée de la pression (pompe).....	123
Figure IV.B.26 : Diagramme électrique de principe de la régulation de pression avec la vanne proportionnelle comme contrôleur.....	124
Figure IV.B.27 : Caractéristique du gain statique.....	125
Figure IV.B.28 : Réponse indicielle en boucle ouverte de la pression (vanne analogique)..	125
Figure IV.B.29 : Lieu d'Evans pour le calcul du régulateur PI.....	126
Figure IV.B.30 : Réponse indicielle en boucle fermée de la pression (vanne).....	126
Figure IV.B.31 : Circuit relatif à la régulation de température.....	127
Figure IV.B.32 : Diagramme électrique de principe de la régulation de température.....	128
Figure IV.B.32 : Organigramme du choix des boucles de régulation.....	129
Figure IV.B.33 : Organigramme de la régulation de niveau.....	130
Figure IV.B.34 : Organigramme de la régulation de débit/pression.....	131
Figure IV.B.35 : Organigramme de la régulation de température.....	132
Figure IV.B.36 : Organigramme de la régulation cascade débit/niveau.....	133
Figure IV.B.38 : Vue d'accueil de l'interface.....	136
Figure IV.B.39 : Vue relative à la régulation de niveau.....	137
Figure IV.B.40 : Vue relatif à la visualisation des courbes de niveau.....	138

Liste des tableaux

Tableau I.B.1 : Position du commutateur du mode de fonctionnement.....	19
Tableau I.B.2 : Les LEDs de visualisations.....	20
Tableau I.B.3 : Caractéristiques techniques de la CPU-313C.....	23
Tableau I.B.4 : Caractéristiques techniques du module E/S digitales.....	25
Tableau I.B.5 : Caractéristiques techniques du module E/S analogique.....	28
Tableau I.B.6 : Correspondance valeurs analogiques tension.....	29
Tableau I.7 : Correspondance valeurs analogiques courant.....	29
Tableau II.1 : Les zones mémoires.....	44
Tableau II.2 : Accès aux différentes zones mémoires.....	45
Tableau II.3 : Les types d'actions du S7-GRAPH.....	53
Tableau II.4 : Entrées du bloc SFB41.....	62
Tableau II.5 : Sorties du bloc SFB41.....	63
Tableau III.1 : différents types de données.....	73
Tableau IV.A.1 : Caractéristiques techniques de l'installation.....	93
Tableau IV.A.2 : Tableau d'adressage des entrées binaires.....	103
Tableau IV.A.3 : Tableau d'adressage des sorties binaires.....	104
Tableau IV.A.4 : Tableau d'adressage des entrées/sorties analogiques.....	104
Figure IV.B.1 : Tableau des configurations relatives aux différentes boucles de régulation.....	106

Introduction générale :

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

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

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

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

Le but de notre travail est donc l'étude des différents composants (capteurs, actionneurs et convertisseurs...) de cette station afin de proposer une solution, à base d'automates programmable Siemens, permettant la gestion de la station d'un point de vue d'identification des processus, contrôle des électrovannes, commande de pompe, régulation de niveau, débit, pression, et température. La communication entre la station et le pupitre de visualisation se fait via un réseau MPI. Il va sans dire que ce travail a nécessité l'étude préalable du logiciel de programmation des API Siemens STEP7 et le logiciel de conception des interfaces homme-machine, WinCC.

CHAPITRE I

Les automates programmables industriels

Introduction :

Tous les secteurs de l'industrie, de la production d'électricité à la peinture pour automobiles, l'emballage alimentaire utilisent les automates programmables industriels pour améliorer la production. Dans ce chapitre, nous présenterons les différents aspects de ces outils puissants et polyvalents. Nous nous intéresserons, également à la gamme de produits SIMATIC proposée par SIEMENS dans le cadre de l'automatisation de l'industrie en général, et à l'automate programmable industriel S7-313C en particulier.

Partie A : Généralité sur les automates programmable industriels (API)

I.A.1 Place des API dans les systèmes automatisés de production :

Le système automatisé de production (SAP) est une notion assez large qui inclut des systèmes de contrôle, de conditionnement, et d'analyse. Ce dernier reçoit un flux de matières ou de produits et génère un flux de produits plus élaborés, de plus il doit gérer l'alimentation en énergie ainsi que des flux auxiliaires. Tout cela, ajouté à des exigences sans cesse accrues de qualité, sécurité, flexibilité, entraîne un accroissement des besoins, en particulier la manipulation d'un grand nombre de variables et la gestion de véritables flux de communication.

Ainsi, les systèmes câblés devenant trop volumineux et trop rigides pour de telles applications, explique que l'on se tourne donc vers des solutions utilisant les techniques de traitement de l'information par processeurs programmables.

La solution reposant sur un processeur central unique s'étant vite révélée peu économique du point de vue du câblage et complexe quant à la maintenance voire dangereuse en cas d'incident, l'utilisation de **processeurs spécialisés et interconnectés** s'est aujourd'hui largement imposée.

L'architecture décentralisée qui en résulte facilite la conception et l'installation en permettant de fractionner les études, la mise en place, les tests ; elle améliore aussi la maintenance (modification aisée des programmes, de parties du système automatisé) et se traduit par plus de flexibilité et de disponibilité. Elle entraîne toutefois, du fait des multiples sous-ensembles fonctionnels, un fort accroissement des besoins de communication et de gestion.

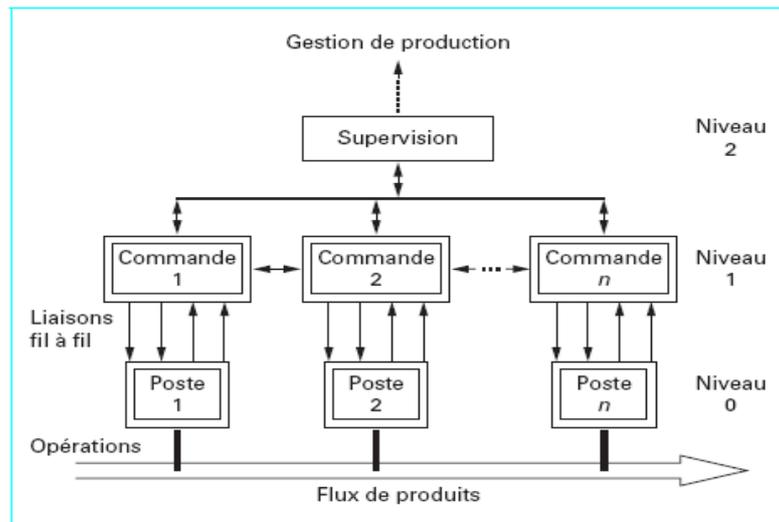


Figure I.A.1: Système automatisé de production [2].

Dans ces systèmes de traitement de l'information, les API occupent une place de choix. Les équipements notés « commande » sont souvent des automates. Ils constituent de plus en plus un maillon fiable et efficace entre le calculateur et l'appareillage de terrain.

I.A.3 Définitions générales :

Il nous faut définir l'API de manière très précise tout en sachant que dans ce domaine et dans beaucoup d'autres touchant à l'informatique, les frontières sont floues et changeantes.

Un système est considéré comme automate programmable si :

- Il est construit autour d'un processeur numérique ;
- Il peut être relié à plusieurs signaux physiques ;
- Il fonctionne grâce à une protection adaptée aux milieux industriels ;
- Il est doté d'un logiciel de programmation ;
- Il est doté de possibilité d'échanges avec d'autres processeurs.

Ceci amène à la définition assez lourde de la norme française EN 61131-1 :

« Système électronique fonctionnant de manière numérique, destiné à être utilisé dans un environnement industriel, qui utilise une mémoire programmable pour le stockage interne des instructions orientées utilisateur aux fins de mise en œuvre de fonctions spécifiques, telles que des fonctions de logique, de mise en séquence, de temporisation, de comptage et de calcul arithmétique, pour commander au moyen d'entrées et de sorties Tout-ou-Rien ou analogiques divers types de machines ou de processus. L'automate programmable et ses périphériques

associés sont conçus pour pouvoir facilement s'intégrer à un système d'automatisme industriel et être facilement utilisés dans toutes leurs fonctions prévues. »

I.A.4 Architecture générale des API :

En général un automate programmable se constitue essentiellement d'une unité centrale, un module d'entrées/sorties, un module d'alimentation, un module de stockage et de liaisons et des auxiliaires [1].

Cette architecture est représentée dans la figure suivante :

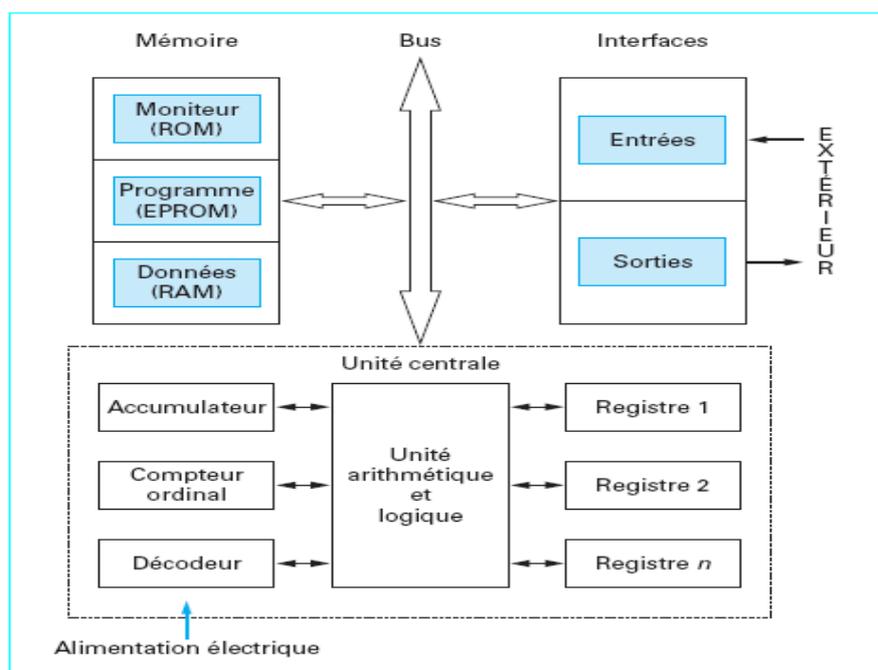


Figure I.A.2: Architecture interne d'un API [2].

I.A.4.1 Module d'alimentation :

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

I.A.4.2 Unité centrale :

L'unité centrale (CPU) est l'élément le plus important dans l'automate programmable, elle peut être considérée comme le cerveau du système. Elle est constituée de deux composants principaux (voir figure 3) :

- Le Processeur ;
- Les mémoires.

I.A.4.2.1 Le processeur :

La principale fonction du processeur est de commander et gouverner les différentes activités du système. Il effectue cette tâche en interprétant et en exécutant un ensemble de programmes système. Ces derniers forment un groupe de programmes superviseurs stockés de façon permanente dans le processeur. Grâce à ces programmes superviseurs le processeur peut ainsi exécuter toutes ses tâches de contrôle, ainsi que diverses fonctions domestiques.

Ces programmes appelés aussi « **le pouvoir exécutif** » assurent la communication entre l'API et l'utilisateur par le biais de dispositifs de programmation. Ils supportent aussi d'autres périphériques de communication tels que la surveillance des appareils de terrain, la lecture des données de diagnostic, l'alimentation, les modules d'entrées/sortie, les mémoires, et la communication avec les interfaces opérateurs.

I.A.4.2.2 Les mémoires :

Tout système bâti autour d'un processeur possède un ou plusieurs types de mémoires. La mémoire système dans un API est composée de deux majeures parties :

- La mémoire exécutive : assure le stockage des programmes superviseurs.
- La mémoire d'application : est une zone de stockage dédiée aux programmes d'utilisateur.

Les exigences de stockage et de récupération pour les programmes superviseurs et les programmes d'application ne sont pas les mêmes, par conséquent ils ne sont pas toujours stockés dans le même type mémoire. Ainsi on aura l'organisation suivante :

- ROM ou PROM : Ce sont des mémoires mortes dont l'utilisateur ne peut que lire le contenu (ROM) et éventuellement les programmer à l'aide d'outils spéciaux (PROM). On y retrouve dans notre cas les programmes superviseurs.
- EPROM : C'est une mémoire reprogrammable qui permet de stocker les programmes mis au point et utilisables.
- RAM : C'est une mémoire vive (volatile) secourue en général par une batterie, elle stocke les données système lors du fonctionnement.

I.A.4.3 Modules d'entrées/sorties (E/S) :

Les modules d'E/S assurent le rôle d'interface de la partie commande, ils se situent entre la CPU et le processus.

Pour ce faire ils doivent :

- Regrouper les variables de même nature pour diminuer la complexité et le coût ;
- Assurer le dialogue avec la CPU ;
- Traduire les signaux industriels en information API et inversement.

Plusieurs types de modules sont disponibles sur les marchés comme :

- Modules d'E/S tout ou rien (TOR) : Ces modules traitent une information qui ne peut prendre que deux états (vrai ou faux, 0 ou 1), ils constituent l'interface entre l'API et les différents capteurs et pré-actionneurs présents.
- Modules d'E/S analogique : Dans ce cas, le signal traité est analogique et prend des valeurs comprises dans une plage bien déterminée. Ces modules sont munis de convertisseur analogique/numérique pour les entrées et respectivement de convertisseur numérique/analogique.
- Module spécialisés : l'information traitée est contenue dans des mots codés sous forme binaire ou bien hexadécimale. C'est le type d'information délivrée par un ordinateur ou un module intelligent.

I.A.4.4 Modules de communication :

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

- Les consoles : Les consoles permettent la programmation, le paramétrage et les relevés d'informations, ils peuvent également afficher le résultat de l'autotest comprenant l'état des modules d'entrées et de sorties, l'état de la mémoire, de la batterie, etc. Ils sont équipés (pour la plupart) d'un écran à cristaux liquides.
- Les boîtiers tests : Les boîtiers de tests quand a eux sont destinés aux personnels d'entretien ; ils permettent de visualiser le programme ou les valeurs des paramètres tes que l'affichage de la ligne de programme à contrôler, la visualisation de l'état des entrées et des sorties...).

I.A.4.5 Auxiliaires :

Il s'agit principalement :

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

I.A.5 Langages de programmation :

I.A.5.1 Introduction :

La norme IEC 1131-3 (Commission Électrotechnique Internationale) définit cinq langages qui peuvent être utilisés pour la programmation des automates programmables industriels. Ces langages peuvent être divisés en deux catégories [3]:

- **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.

I.A.5.2 Les langages graphiques :

I.A.5.2.1 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 automate. 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 :

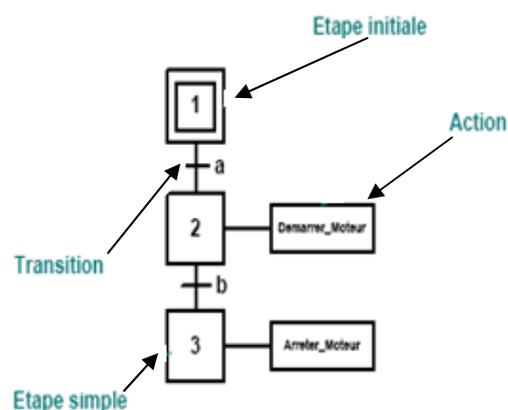


Figure I.A.3 : Le GRAFCET.

Chapitre I : Les automates programmables industriels

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.

I.A.5.2.2 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:

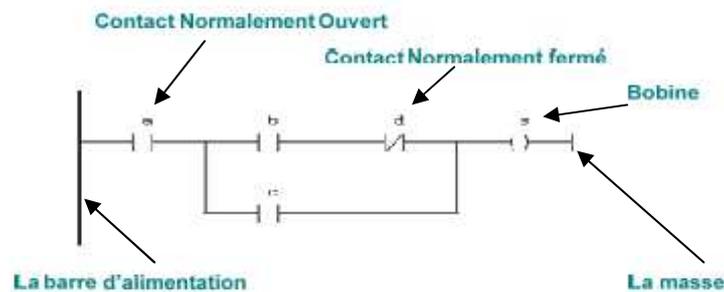


Figure I.A.4 : Exemple 1 d'un programme en Ladder.

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.

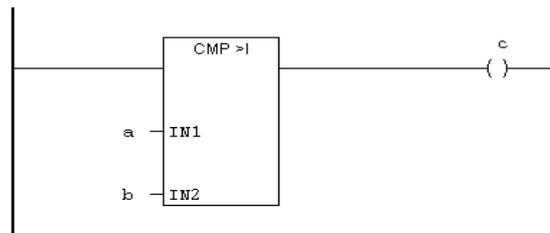


Figure I.A.5 : Exemple 2 d'un programme en Ladder.

I.A.5.2.3 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 I.A.6 : Exemple d'un programme en Fonction Bloc.

I.A.5.3 Les langages textuels :

I.A.5.3.1 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.

Chapitre I : Les automates programmables industriels

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 INPLIT
X1, X2, Y1, Y2 : REAL;
END VAR
BEGIN
RESULT := SQRT((X1-X2)^2 + (Y1-Y2)^2);
END_FUNCTION
```

Figure I.A.7 : Exemple d'un programme en Langage Structuré.

I.A.5.3.2 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 I.A.8 : Exemple d'un programme en Liste d'Instructions.

I.A.5.4 Conclusion :

La plus part des grands constructeurs d'automates programmables, fournissent des logiciels de configuration et de programmation munis des langages SFC, LD, FBD, ST et IL.

Le choix d'un langage s'appuie sur la complexité de l'application et de la tâche de commande. Il est préférable d'utiliser les langages graphiques (SFC, LD et FBD) pour la réalisation des programmes de commande séquentielle. Le SFC est la réalisation direct d'un GRAFCET de commande, les langages LD et FBD sont plus utiles pour les opérations combinatoires sur bits ou mots.

Les langages textuels sont beaucoup plus performants pour le traitement de variables continues ou analogiques ainsi que pour la commande des systèmes continus. Les programmes en IL sont un peu fastidieux à mettre en œuvre, mais connaissent une optimisation optimale pour le temps de traitement et l'occupation de la mémoire. Le ST est le langage par excellence, très utile pour des utilisateurs ayant des connaissances en langages évolués tel que PASCAL.

Partie B : Les automates programmable industriels Siemens

I.B.1 Présentation de la gamme SIMATIC Siemens :

I.B.1.1 Historique :

Les années soixante-dix ont connu une explosion des besoins industriels dans le domaine de l'automatique, de la flexibilité et de l'évolutivité des systèmes automatisés de production (SAP).

Siemens AG est un groupe allemand. Fondé en 1847 par Werner Von Siemens ; il réalise des équipements électroniques et électrotechniques. Son siège est à Munich, et c'est l'une des plus grandes entreprises européennes.

Dans le cadre de son expansion, Siemens crée le 28 janvier 1972, le consortium UNIDATA. Ce projet européen a permis, dans les années 1970, l'émergence d'une grande industrie informatique européenne. Les compétences de trois participants ont été mises en commun. La maîtrise d'œuvre, l'architecture des machines et le logiciel ont été attribués à la Compagnie Internationale pour l'Informatique (CII), la technologie électronique est revenue à Philips, tandis que Siemens se chargeait des périphériques mécaniques.

En 1975, la France abandonne unilatéralement l'accord UNIDATA, CII fusionne avec Honeywell-Bull, Philips délaisse l'informatique et Siemens rejoint Fujitsu pour devenir, aujourd'hui, un des plus grands constructeurs mondiaux.

I.B.1.2 Les différentes variantes dans la gamme SIMATIC :

De nos jours SIEMENS propose une gamme complète de produits pour l'automatisation industrielle. En effet, par le biais de sa gamme SIMATIC, elle intègre tout l'environnement d'automatisation. Ceci est effectué par :

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

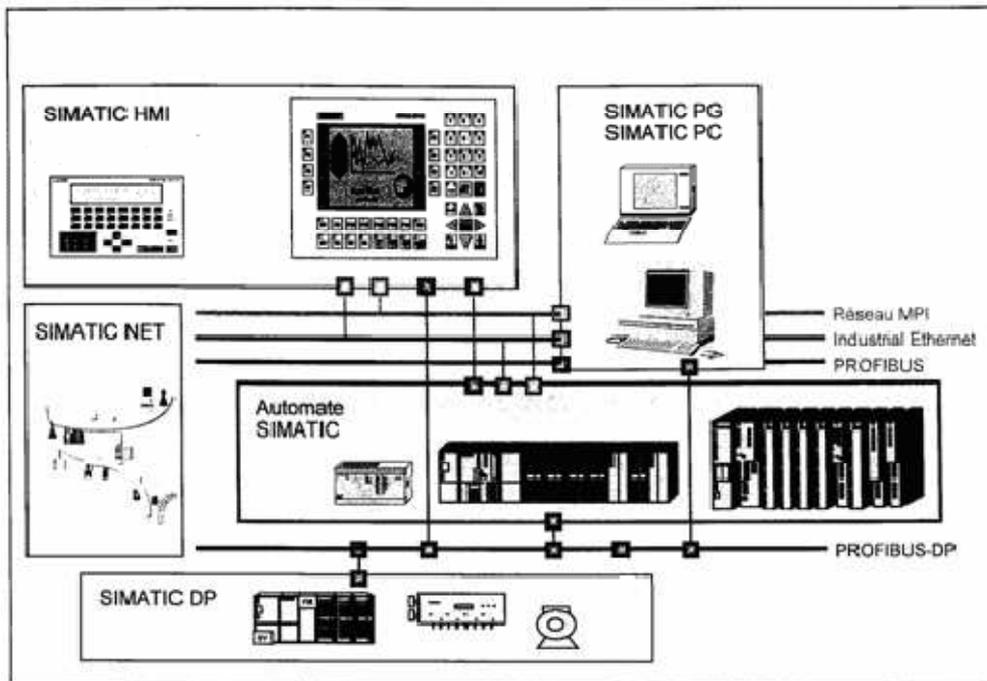


Figure I.B.1 : Présentation de la gamme SIMATIC.

La gamme SIMATIC comporte trois grandes familles :

a. SIMATIC S7 :

La gamme SIMATIC S7 contient différentes familles d'automates :

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

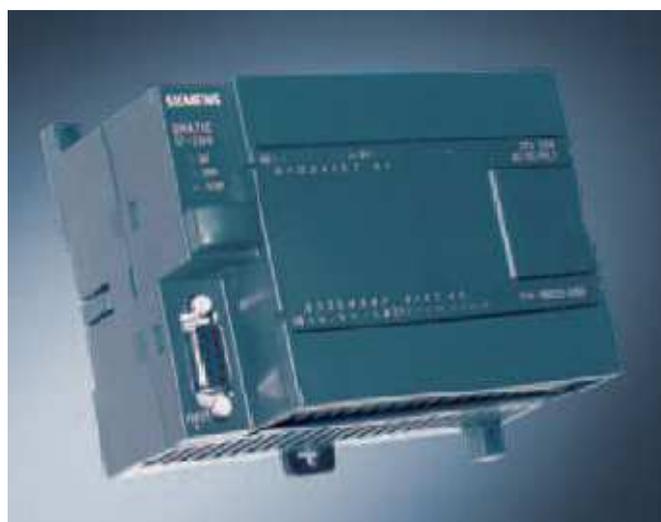


Figure I.B.2 : L'API S200.

Chapitre I : Les automates programmables industriels

- S7 300 : est un Mini-automate modulaire pour les applications d'entrées et de milieu de gamme, avec possibilités d'extension jusqu'à 32 modules, et une mise en réseau par l'interface multipoint (MPI), PROFIBUS et Industrial Ethernet.



Figure I.B.3 : L'API S300.

- S7400 : 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 Industrial Ethernet.



Figure I.B.4 : L'API S400.

b. 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 utilise comme une interface Homme/Machine HMI.

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



Figure I.B.5 : La gamme SIMATIC C7.

c. SIMATIC M7 :

Les SIMATIC M7 sont des calculateurs industriels compatibles PC. Il s'agit d'un système modulaire sous boitier, 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 I.B.6 : La gamme SIMATIC M7.

I.B.2 Description de l'automate programmable S7-313 :



Figure I.B.7 : 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.

Chapitre I : Les automates programmables industriels

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 :

I.B.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 I.B.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

Chapitre I : Les automates programmables industriels

STOP (jaune)	CPU en STOP ou ATTENTE ou en démarrage ;la LED clignote en cas d'effacement général
--------------	---

Tableau I.B.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 (μ s)	
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

Chapitre I : Les automates programmables industriels

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 :	

Chapitre I : Les automates programmables industriels

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	

Chapitre I : Les automates programmables industriels

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 I.B.3 : Caractéristiques techniques de la CPU-313C [4].

I.B.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 :	24
Nombre	12
Entrées utilisables pour les fonctions technologiques	
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

Chapitre I : Les automates programmables industriels

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 I.B.4 : Caractéristiques techniques du module E/S digitales [5].

I.B.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

Chapitre I : Les automates programmables industriels

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 successives)
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

Chapitre I : Les automates programmables industriels

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 V/100 k Ω
Courant	$\pm 20 \text{ mA}/50 \Omega$; 0 mA à 20 mA/50 Ω ; 4 mA à 20 mA/50 Ω
Résistance	0 Ω à 600 Ω /10 M Ω
Thermomètre de résistance	Pt 100/10 M Ω
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

Chapitre I : Les automates programmables industriels

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 I.B.5 : Caractéristiques techniques du module E/S analogique [5].

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		

Chapitre I : Les automates programmables industriels

100 %	27648	6C00	10 V	Etendue nominale
75 %	20736	5100	7.5 V	
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 I.B.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 I.7 : Correspondance valeurs analogiques courant [5].

Conclusion :

De part la mobilité, la flexibilité de son architecture, la facilité de sa programmation, de sa connexion et de son adaptation dans les milieux industriels, l'automate programmable est devenu un produit incontournable dans les systèmes automatisés de production. 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 II

**STEP7, logiciel de programmation des API
Siemens**

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.

Dans ce qui va suivre nous allons étudier le logiciel de programmation et de communication step7 ; ce dernier permet la programmation et la simulation de procédures automatisées avec différents langages normalisés.

II.1. Présentation du logiciel STEP7 :

Le STEP7 est un logiciel de base pour la programmation et la configuration des systèmes d'automatisation S300/S400. En effet il permet l'accès de base à un automate de la gamme SIMATIC, ainsi que sa programmation en différents langages. Il assure également la fonction de moyen de communication en prenant en compte le réseau des automates. Ce logiciel assiste l'opérateur dans toutes les phases de création de la solution d'automatisation.

On peut diviser le step7 en six sous-logiciels interdépendants :

➤ Gestionnaire de projets SIMATIC Manager :

Ce gestionnaire apparaît dès le lancement du step7. C'est l'interface d'accès à la programmation et à la configuration matérielle. En effet il permet de gérer toutes les données relatives au projet d'automatisation.

➤ Editeur de la configuration matérielle :

L'éditeur HWConfig permet la configuration et le paramétrage du matériel d'un projet d'automatisation, d'un point de vue comportement à la mise en route, surveillance du temps de cycle pour ce qui est de la CPU, et d'un point de vue adressage pour ce qui est des modules d'entrées/sorties.

➤ Editeurs de programmes des Différents langages de programmation :

Trois langages de programmation font partie intégrante du logiciel de base :

- Editeur de programme CONT, LIST, LOG qui correspondent respectivement aux langages normalisés LD Ladder Diagram, IL Instruction List, FBD Fonction Bloc Diagram).

D'autres langages de programmation peuvent être inclus sous forme de logiciels additionnels :

- Editeur de programmes SCL qui correspond au langage normalisé ST Structured Text.
 - Editeur de programmes Graph qui correspond au GRAFCET.
- Interface de paramétrage du régulateur PID :
- Cette interface permet de paramétrer le régulateur PID du projet d'automatisation, il permet aussi une visualisation graphique du comportement de la régulation.
- Paramétrage de l'interface PG/PC :
- Cette interface a pour but de configurer la communication avec l'automate en termes de vitesse de transmission dans le réseau Profibus ou MPI en vue d'une transmission de projets.
- Le simulateur de programmes PLCSIM :
- Ce logiciel permet d'exécuter et de simuler un programme S7, la simulation étant réalisée complètement au sein du logiciel step7, il n'est pas nécessaire qu'une liaison PC/automate soit établie. Cette interface simple permet de visualiser et de forcer les différents paramètres présents dans le programme.

II.2. Programmation a l'aide du logiciel STEP7 :

Dans ce qui va suivre nous allons essayer de traiter plusieurs exemples d'automatisation afin de pouvoir présenter toutes les possibilités de programmation qu'offre step7.

La stratégie pour la conception d'une structure de programmes complète et optimisée nécessite de réaliser les deux tâches suivantes :

- La configuration matérielle ;
- Edition des programmes utilisateurs.

II.2.1. Création d'un projet SIMATIC Manager :

Le SIMATIC Manager apparaît de manière automatique dès le lancement du step7. Un double clic sur « nouveau projet » permet de créer un projet d'automatisation.

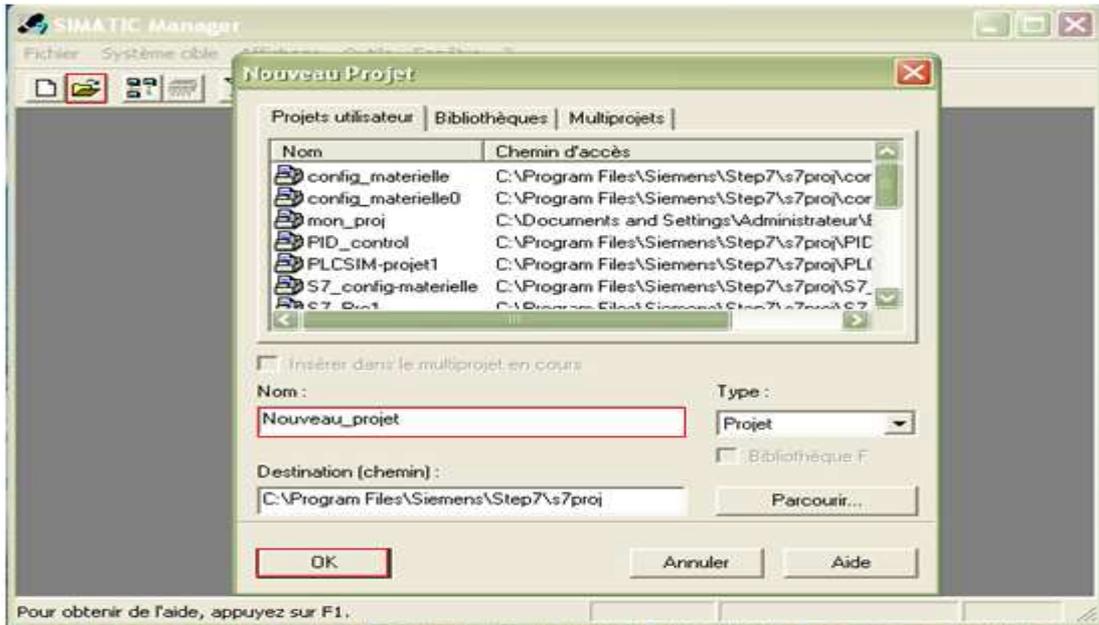


Figure II.1 : Création d'un nouveau projet (1).

Une fois le projet nommé et sauvegardé, la fenêtre suivante apparaît :

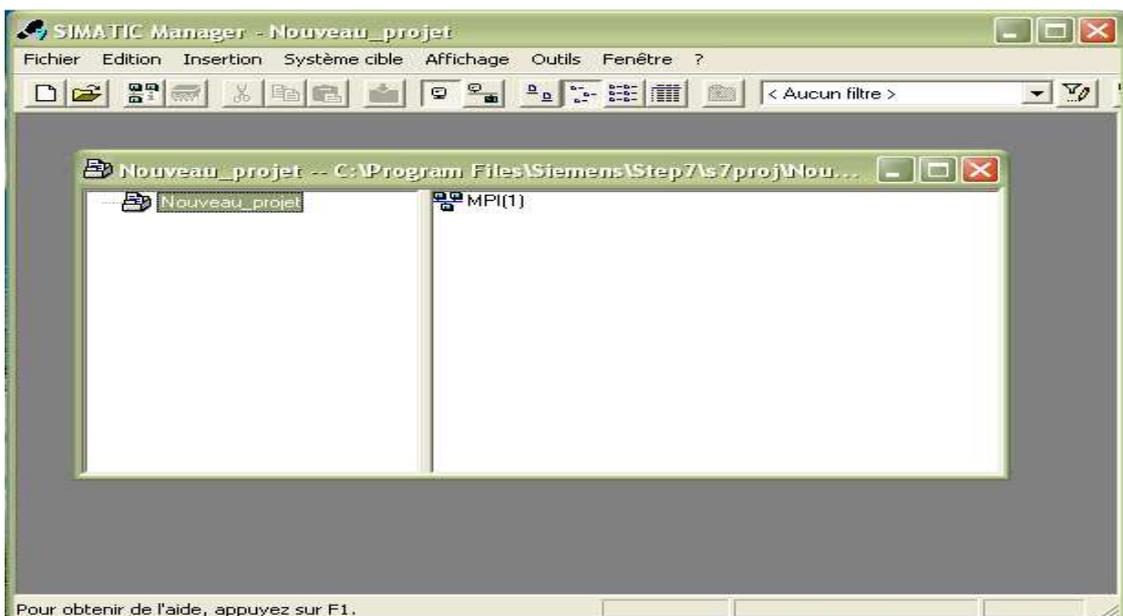


Figure II.2 : Création d'un nouveau projet (2).

Un clic droit sur le projet en cours nous permet de choisir la station SIMATIC adéquate. Le choix d'une station correspond à une configuration matérielle bien déterminée. Cette station peut inclure un ou plusieurs modules programmables.

Les stations disponibles sont :

- SIMATIC 400 ;
- SIMATIC 300 ;
- SIMATIC H ;
- SIMATIC PC ;
- SIMATIC S5.

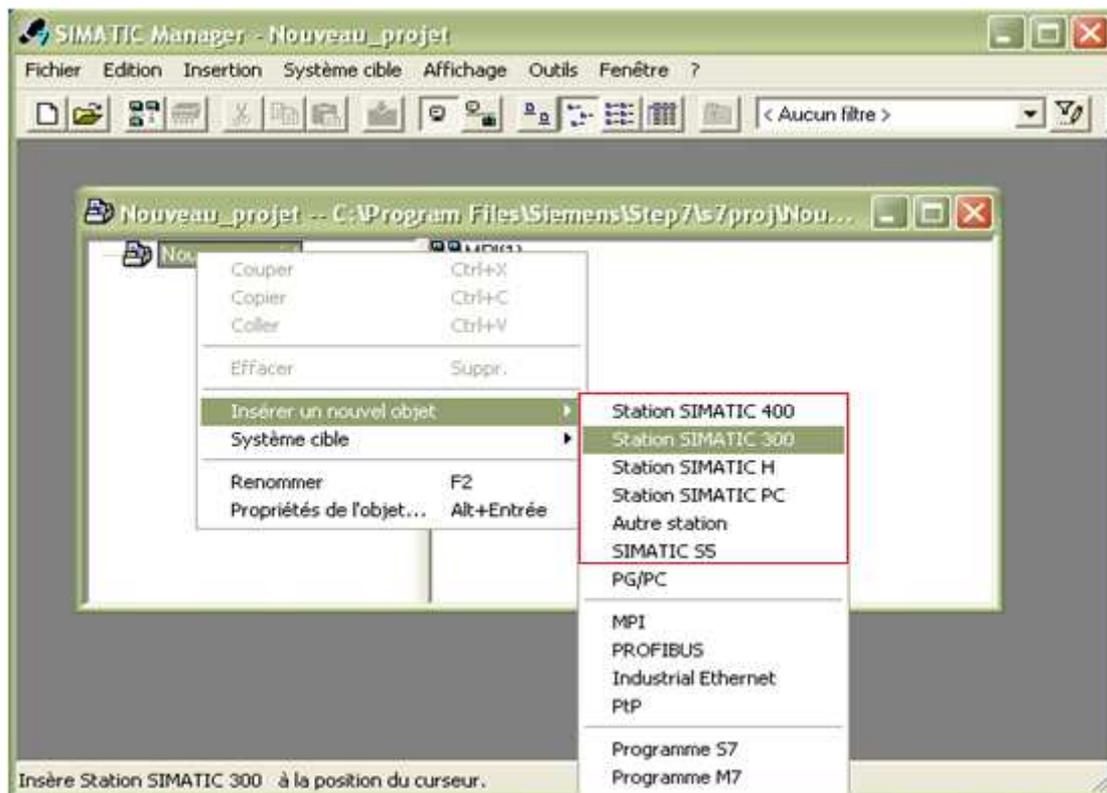


Figure II.3 : Choix de la station SIMATIC de travail.

Notre choix s'est arrêté sur la station SIMATIC 300.

II.2.1.1. Configuration matérielle :

Par configuration on entend la disposition de châssis, de modules, d'appareils de la périphérie décentralisée ou de cartouches interface dans une fenêtre de station. Les châssis ou

profilés support sont représentés par une table de configuration dans laquelle on peut disposer, dans un ordre bien défini, un nombre précis de modules, tout comme dans la réalité.

Par configuration on entend aussi « paramétrage », c'est-à-dire :

- Le réglage des paramètres des modules paramétrables pour la configuration centralisée et pour un réseau. Exemple : une CPU est un module paramétrable. La surveillance du temps de cycle est un paramètre que vous pouvez définir ;
- La définition des paramètres de bus, des maîtres et d'esclaves pour un réseau maître (PROFIBUS) ou d'autres définitions pour l'échange de données entre des composants.

La configuration matérielle n'est pas toujours nécessaire. Elle est effectuée dans des cas bien précis, par exemple pour la configuration des liaisons de communication ou bien pour les stations S7-400, ou il y a plusieurs CPU et châssis d'extension.

Dans step7 la configuration matérielle est effectuée à l'aide de l'interface HWconfig et la marche à suivre est la suivante :

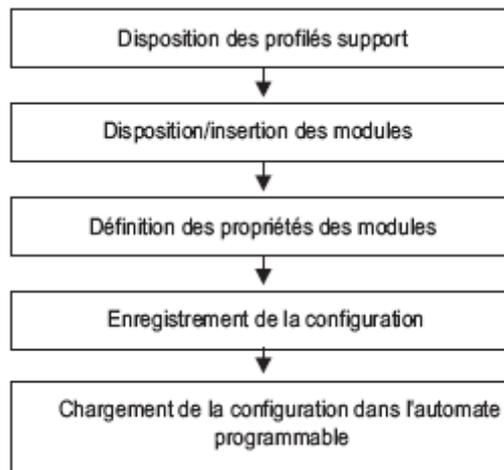


Figure II.4 : Organigramme de la configuration matérielle.

Une fois la station choisie vient alors la partie configuration matérielle. Un double clic sur « matériel » ouvre l'interface HWconfig.

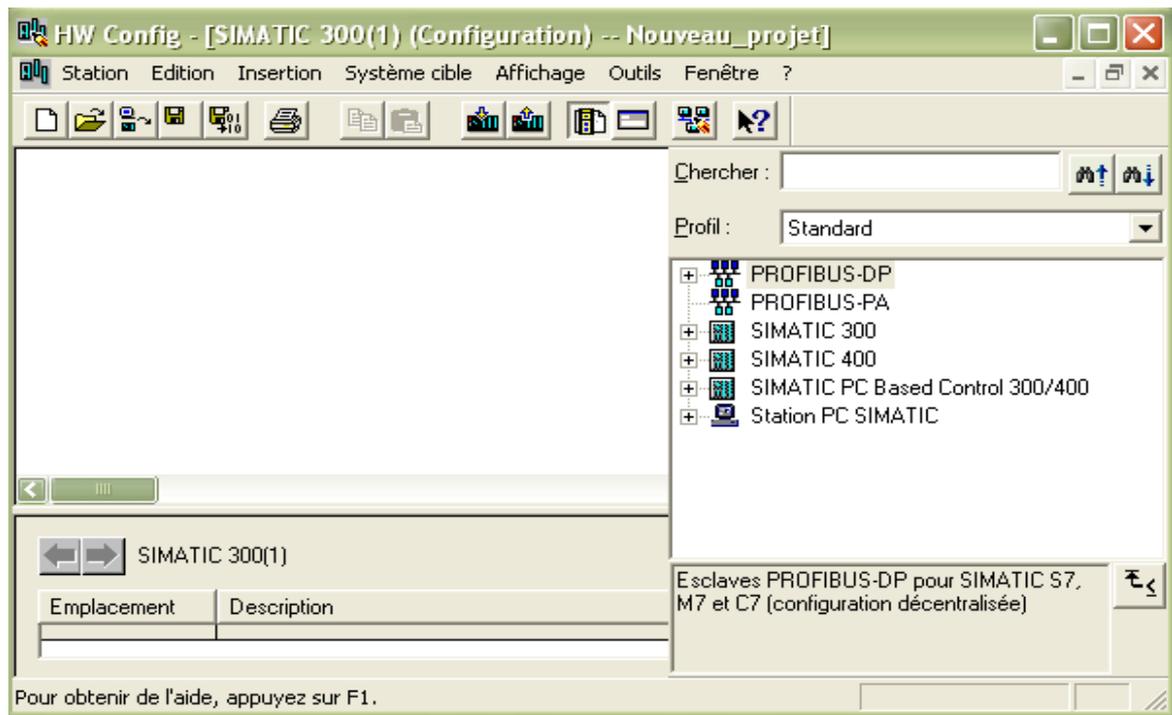


Figure II.5 : Editeur des configurations matérielles.

Après le choix de SIMATIC 300 dans catalogue, une liste d'éléments s'offre à nous :

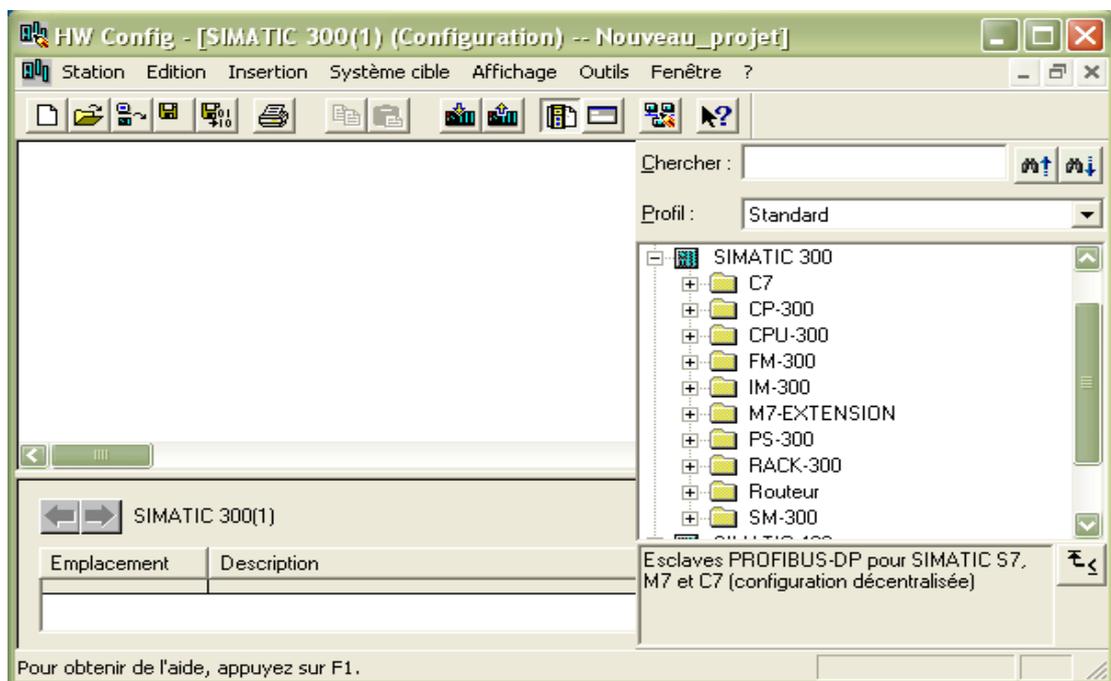


Figure II.6 : Catalogue SIMATIC 300.

Un double clic sur Rack-300 permet d'insérer le profilé support qui contiendra tous les différents modules par la suite.

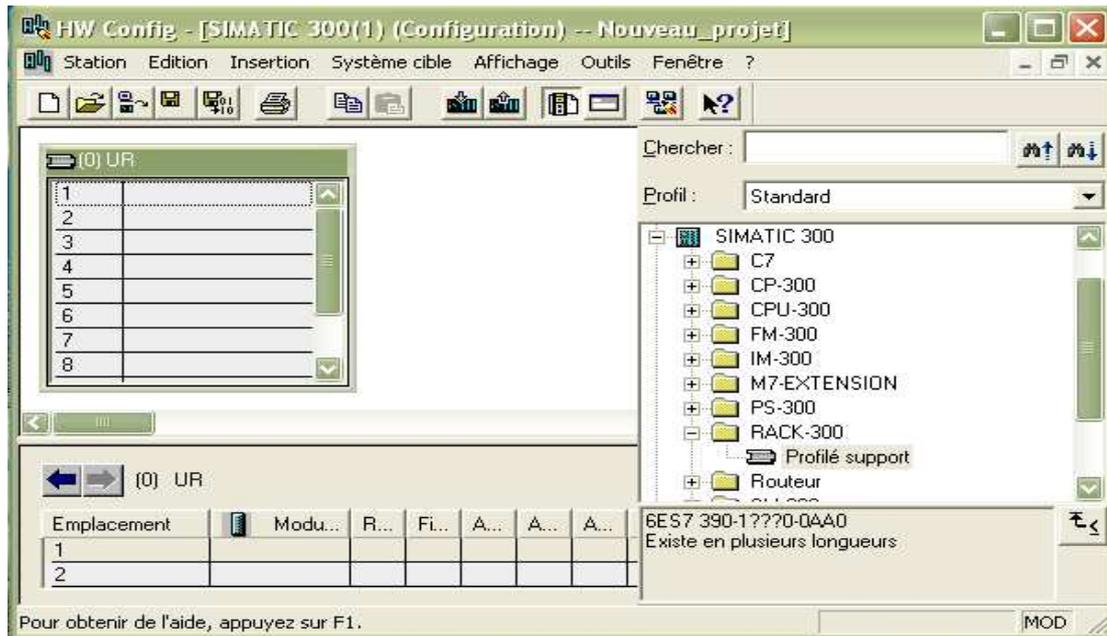


Figure II.7 : Sélection des modules (1).

Le rack possède onze emplacements, le premier emplacement est réservé exclusivement au module d'alimentation, le deuxième, lui, est réservé à la CPU, le troisième est quant à lui réservé à des modules spéciaux. Ainsi, il ne reste que huit emplacements pour les type de cartes standards (module entrées/sorties.....).

En sélectionnant les différents emplacements et en double cliquant sur les éléments correspondant dans le catalogue à gauche, ces derniers s'insèrent automatiquement dans le châssis.

Ainsi :

- les différents types de modules d'alimentation sont dans le dossier PS-300 (relatif à la station SIMATIC 300 choisie) classés en fonction du courant et de la tension d'alimentation.
- Les différentes CPU disponibles sont classées dans le dossier CPU-300. (Voir annexe-I)
- Les modules d'entrées/sorties se trouvent dans le dossier SM-300.
- Les coupleurs d'extension dans le dossier IM-300.
- Les modules de fonction (régulation, comptage...) se trouvent dans FM-300.
- Les modules d'extension et cartouches interface pour SIMATIC M7 dans M7-EXTENSION.

- Les modules de communication (Profibus, Industriel Ethernet,...) dans CP-300.

Après il ne restera qu'à charger la configuration dans le projet. Pour un exemple de configuration bien précis, la table de configuration devient :

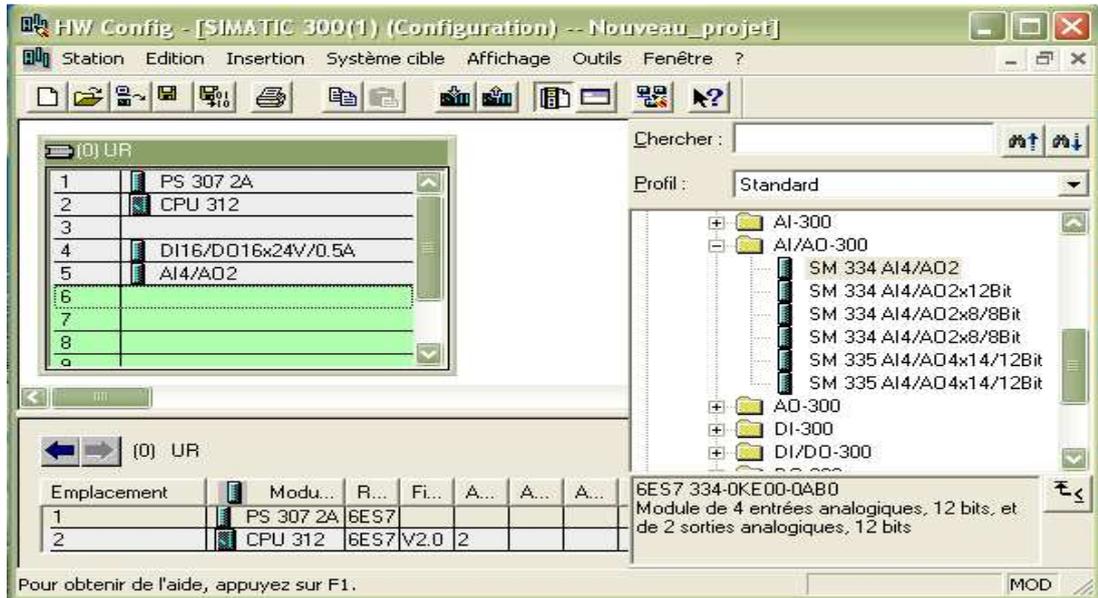


Figure II.8 : Sélection des modules (2).

Après l'insertion de tous les éléments requis pour la configuration, un double clic sur chacun d'eux permet leurs paramètres.

On peut par exemple fixer le temps de la CPU (si elle le permet) :

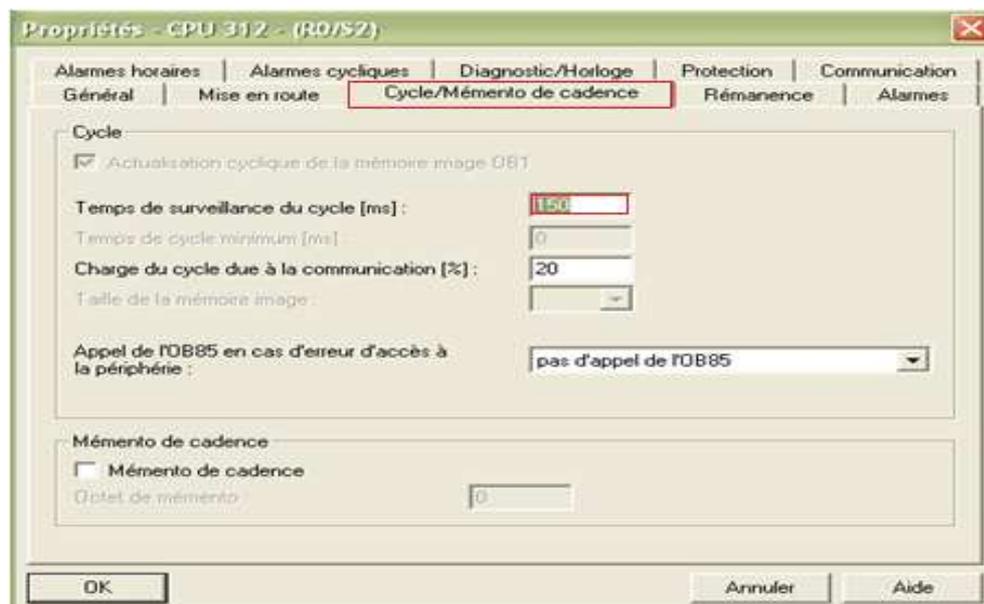


Figure II.9 : Paramétrage de la CPU.

On peut aussi changer l'adresse des modules d'entrées/sorties :

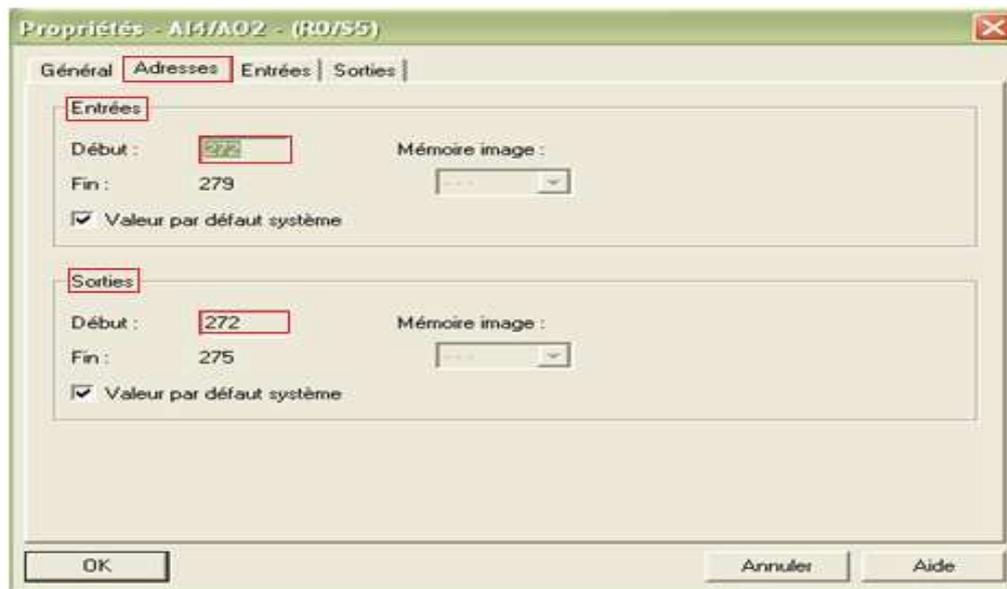


Figure II.10 : Paramétrage du module E/S analogiques.

II.2.1.2. Elaboration des programmes S7 :

a. Organisation par blocs des programmes S7 :

Le STEP 7 permet de structurer le programme utilisateur en le subdivisant en différentes parties (blocs) autonomes ou dépendantes. Cette organisation permet une vision globale claire et facile à tester.

➤ Blocs d'organisation :

Les blocs d'organisation (OB) constituent l'interface entre le système d'exploitation et le programme utilisateur. Ils sont appelés par le système d'exploitation et gèrent non seulement le traitement du programme cyclique mais aussi celui des programmes déclenchés par alarmes, ainsi que le comportement à la mise en route de l'automate programmable et le traitement des erreurs.

Les blocs d'organisation définissent l'ordre dans lequel les différentes parties du programme sont traitées. Les différents types de blocs d'organisation ainsi que leurs priorités sont développés dans l'annexe-II-.

➤ Fonctions et blocs fonctionnels :

On peut programmer chaque bloc d'organisation en tant que programme structuré en créant des fonctions (FC) et des blocs fonctionnels (FB).

- Les blocs fonctionnels (FB) sont des blocs de code associés à des blocs de données d'instance, dans lesquels sont sauvegardés les paramètres effectifs et les données statiques des blocs fonctionnels.
- Les fonctions (FC) sont des blocs de code sans rémanence, c'est-à-dire qu'ils ne sont pas associés à des blocs de données. Les paramètres effectifs ne sont pas sauvegardés automatiquement.

De plus, il existe les blocs fonctionnels système (SFB) et les fonctions système (SFC), qui sont des fonctions préprogrammés. Ces dernières peuvent être appelées par le programme utilisateur.

➤ Blocs de données :

Les blocs de données (DB) servent à l'enregistrement de données utilisateur. Ils se divisent à leur tour en :

- Blocs de données globaux : ils servent en enregistrer des données accessible à tous les autre blocs.
- Blocs de données d'instance : ils sont affectés aux blocs fonctionnels (FB).

Comme le montre l'arborescence suivante, un répertoire « programmes » est associé à la CPU choisie (ici 312) ; ce dernier contient à son tour deux sous répertoires « blocs », « sources » et « mnémoniques ».

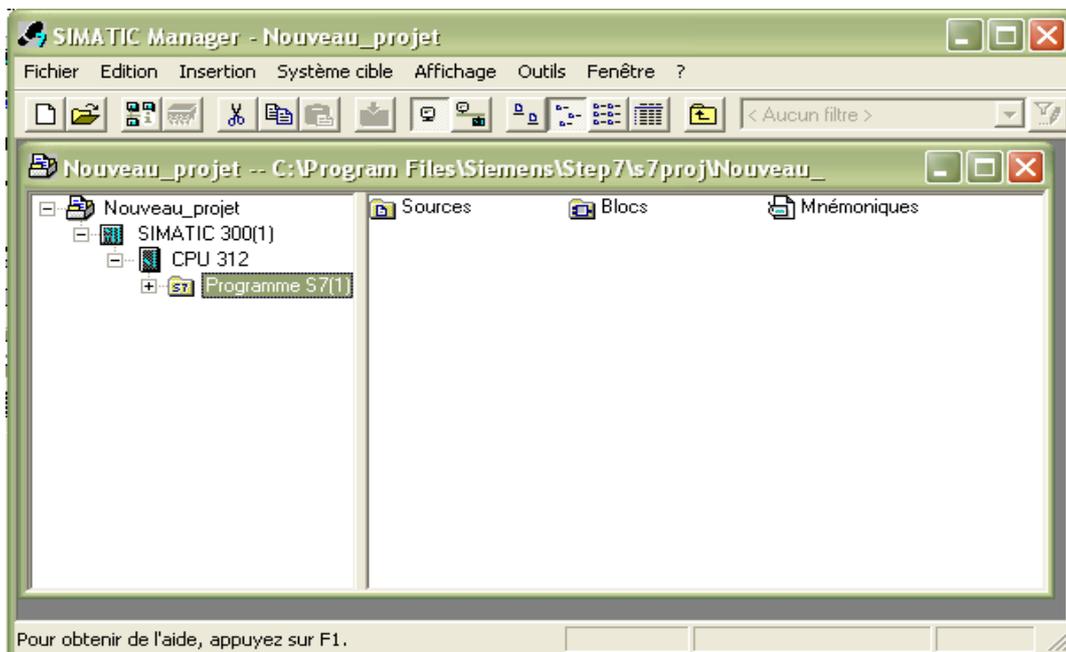


Figure II.11 : Organisation par blocs des programmes S7 (1).

Si on clique sur le dossier « blocs » ou trouvera le bloc d'organisation (OB1), ce dernier est associé par défaut à la CPU choisie, il permet l'exécution continue du programme.

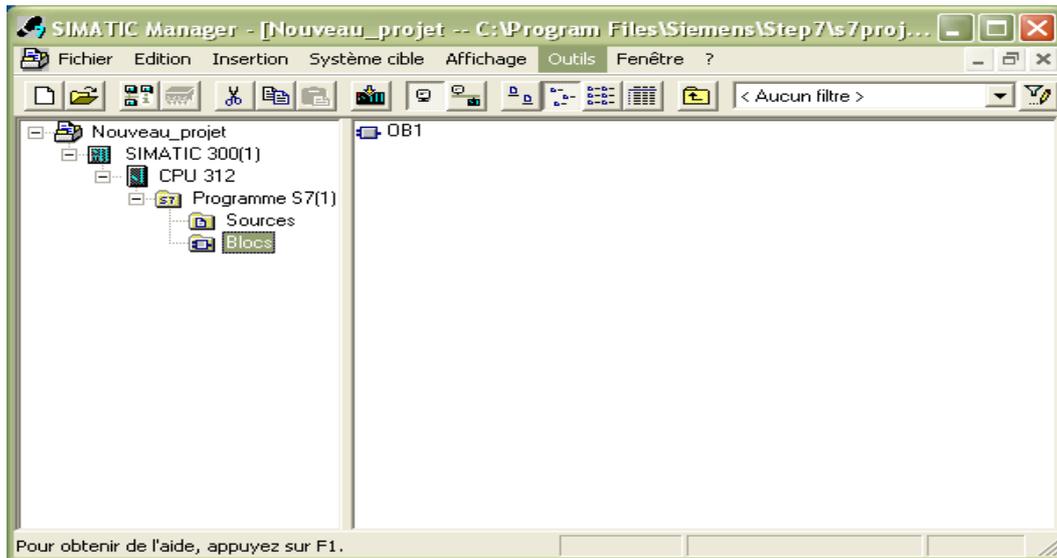


Figure II.12 : Organisation par blocs des programmes S7 (2).

Un clic droit sur « blocs » permet d'insérer de nouveaux blocs (OB, FB, DB....).

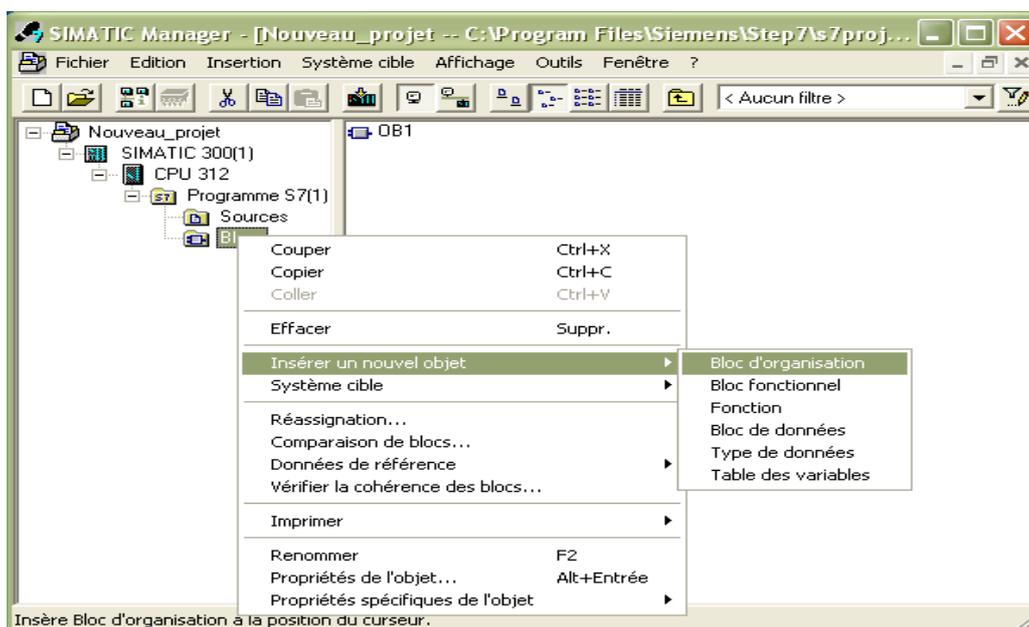


Figure II.13 : Insertion d'un nouveau bloc d'organisation.

Le répertoire « source » contiendra des fichiers source. Ces derniers seront complémentaires aux blocs déjà présents. Un clic droit sur « source » permettra d'insérer un nouveau fichier, ainsi que le choix du langage de programmation.

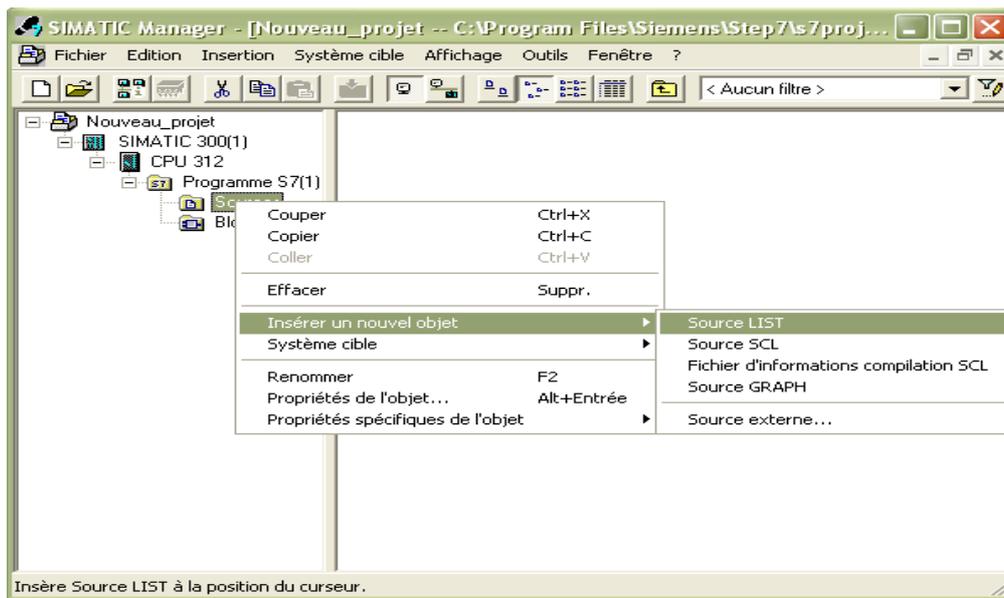


Figure II.14 : Insertion d'un nouveau fichier source.

b. Edition de la table des mnémoniques :

Toujours dans le SIMATIC Manager si on double clic sur le dossier « mnémoniques », on ouvre l'éditeur des mnémoniques.

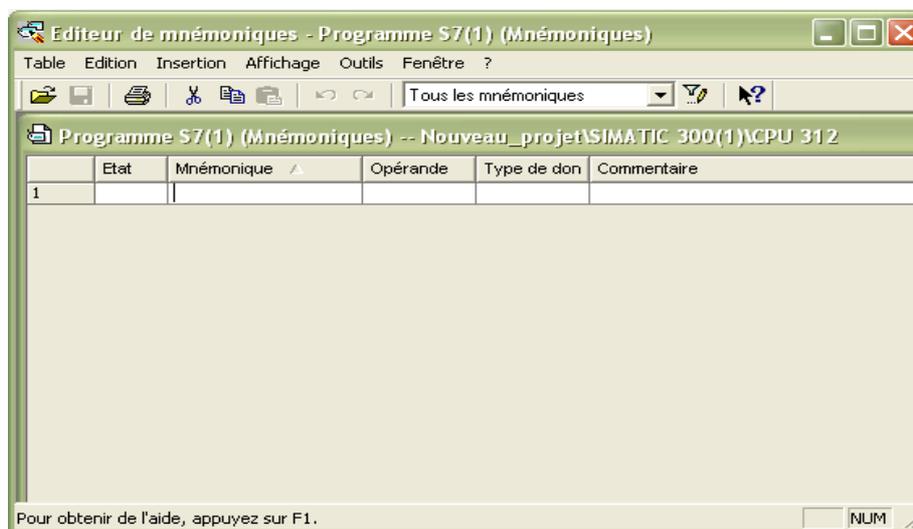


Figure II.15 : Editeur des mnémoniques.

Les mnémoniques sont des variables globales accessibles à tous les blocs et sous-programmes. Il existe plusieurs types de variables ; nous les résumons dans le tableau suivant :

Zone mémoire	Type de la variable	Valeurs possibles
Pour le bit	Booléenne	Vrai ou faux (1, 0)
Pour l'octet (8 bits)	Deux de données possibles : -BYTE : Nombre hexadécimale. -CHAR : Caractères ASCII.	-De B#16#0 à B#16#FF. -'A', 'B'.....
Pour le mot (16 bits)	Plusieurs possibilités : -WORD : Nombre hexadécimale. -INT : Nombre entier. - S5TIME : durée s7 en pas de 10ms (valeur par défaut). - DATE : Date en incréments de 1 jour.	- De W#16#0 à W#16#FFFF. - De -32768 à 32767. - De S5T#0H_0M_0S_10MS à S5T#2H_46M_30S_0MS. - De D#1990-1-1 à D#2168-12-31.
Pour le double mot (32 bits)	Plusieurs possibilités : - DWORD : nombre hexadécimal. - DINT : nombre entier. - REAL : nombre à virgule flottante. - TIME : Durée en incréments de 1 ms. - TIME_OF_DAY : Heure en pas de 1 ms.	- De DW#16#0000_0000 à DW#16#FFFF_FFFF. - DE L#-2147483648 à L#2147483647. - Limite supérieure : 3.402823e+38 Limite inférieure : 1.175 495e-38. - DE -T#24D_20H_31M_23S_647MS. à T#24D_20H_31M_23S_647MS. - De TOD#0:0:0.0 à TOD#23:59:59.999.

Tableau II.1 : Les zones mémoires.

Tout type de variable peut être associé à une mnémonique dont l'accès est défini comme suit :

Nom de la zone	Description	Accès à la zone par :
Mémoire Image des Entrées (MIE)	Au début du cycle le système d'exploitation lit les entrées provenant du processus et enregistre ces valeurs dans la MIE. Le programme utilise ces valeurs pendant son traitement normal.	Bit : E Octet : EB Mot : EW Double mot : ED
Mémoire Image des Sorties (MIS)	Pendant le cycle le programme calcule les valeurs de sortie et les dépose dans la MIS. A la fin du cycle le système d'exploitation lit les valeurs de sorties figurées dans la MIS et les transmet aux sorties du processus.	Bit : A Octet : AB Mot : AW Double mot : AD
Mémentos	Se sont des zones mémoires qui permettent de sauvegarder les résultats intermédiaires calculés dans le programme.	Bit : M Octet : MB Mot : MW Double mot : MD
Périphérie d'entrée et de sortie	Cette zone permet d'accéder directement aux modules d'entrées et de sorties.	Octet : PEB, PAB Mot : PEW, PAW Double mot : PED ou PAD
Temporisations	Cette zone sert d'espace mémoire pour les cellules de temporisation, l'horloge accède à ces cellules afin de les mettre à jour en décrémentant la valeur de temps.	T
Compteurs	Cette zone mémoire, sert d'espace mémoire pour les opérations de comptage.	Z
données locales	Cette zone contient les données temporaires, elle est utilisée dans les blocs de code (OB, FB ou FC). Ces données sont rangées dans la pile des données locales, elles seront perdues une fois le bloc de code achevé.	Bit : L Octet : LE Mot : LW Double mot : LD

Tableau II.2 : Accès aux différentes zones mémoires.

L'éditeur des mnémoniques permet ainsi d'affecter un nom et une adresse absolue à chaque variable déclarée. Ainsi, la correction et la mise à jour des programmes seront aisées.

c. Différents langages disponibles dans le STEP7 :

La programmation des blocs de codes peut se faire à l'aide de trois applications :

➤ **CONT, LIST, LOG** :

Ce langage permet de programmer les blocs d'organisation (OB), les blocs fonctionnels (FB), ainsi que les fonctions (FC).

Pour accéder à l'éditeur des programmes CONT, LIST, LOG il suffit de double cliquer sur l'objet précédemment créé (OB, FB, FC). Ainsi on obtiendra la fenêtre suivante :

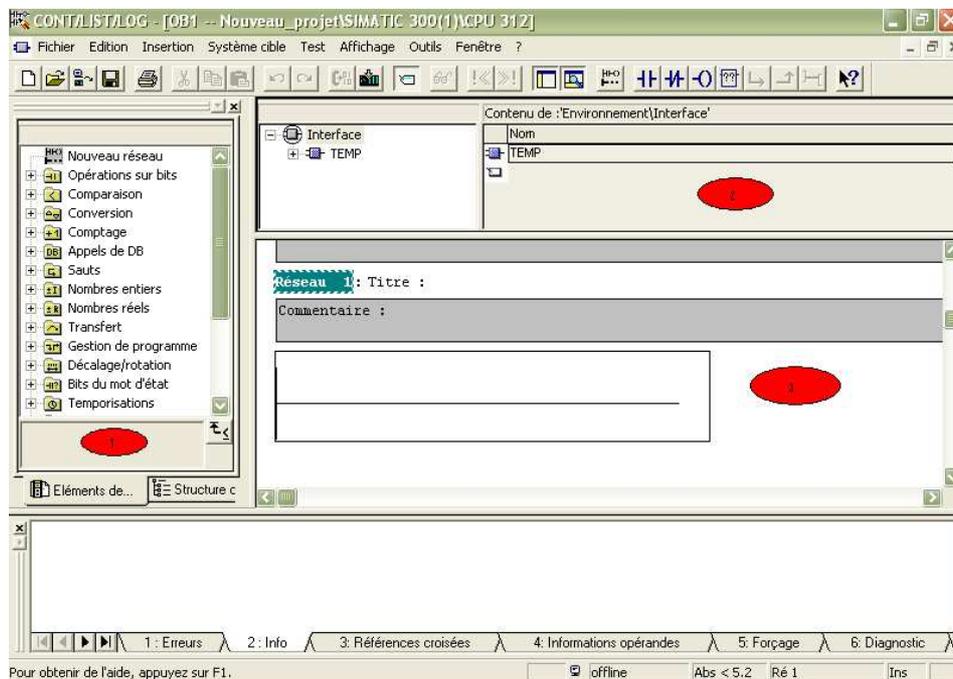


Figure II.16 : Editeur de programme CONT, LIST, LOG.

On peut distinguer trois Zones principales :

1. Le catalogue : c'est la bibliothèque des opérations et instructions du langage choisit (ici CONT), ces différentes opérations sont regroupées par famille : opération sur les bits, sur les mots, sur les entiers... .Pour utiliser une opération, il suffit de faire glisser l'élément vers la zone souhaitée.

2. La zone de déclarations : dans cette zone on définit les variables locales, les variables d'entrées et sorties ou arguments pour les fonctions et les blocs fonctionnels.
3. La zone d'édition : elle est structurée en réseaux ; chaque réseau se termine par une affectation ou un saut.

On remarque que les programmes sont structurés sous forme de réseaux avec la possibilité de choisir un langage de programmation et cela en cliquant sur 'affichage' comme nous le montre l'image ci-dessous.

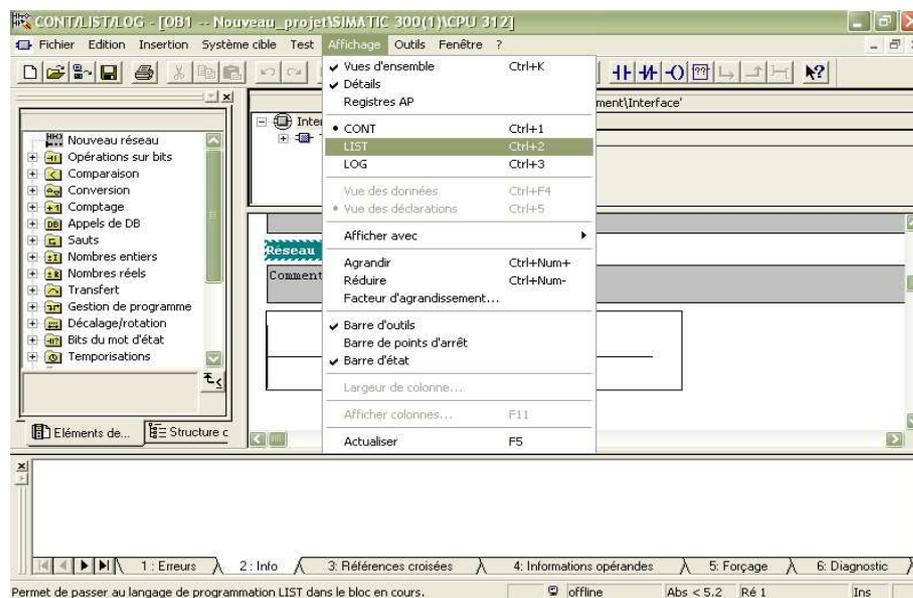


Figure II.17 : Choix du langage de programmation.

➤ **Texte structuré (SCL) :**

SCL est un langage de programmation textuel de haut niveau dédié aux applications d'automatisations. Il est utilisé principalement pour décrire des procédures complexes et difficilement modélisables, avec les langages graphiques. Il peut aussi être utilisé de façon complémentaire avec d'autres langages de programmation. Ainsi un bloc programmé en SCL peut appeler un autre programme en LIST, CONT ou LOG.

Le SCL possède un jeu d'instructions très riche ; il utilise les mêmes énoncés que les langages de programmation évolués (PASCAL, C..), comme les assignations, les appels de fonctions, les énoncés de contrôle (IF, THES, ELSE, CASE...), les énoncés d'itération (FOR, WHILE, REPEAT...).

Pour créer un fichier source SCL, il suffit de revenir au SIMATIC Manager, un clic droit sur le répertoire source permet l'insertion de ce fichier source.

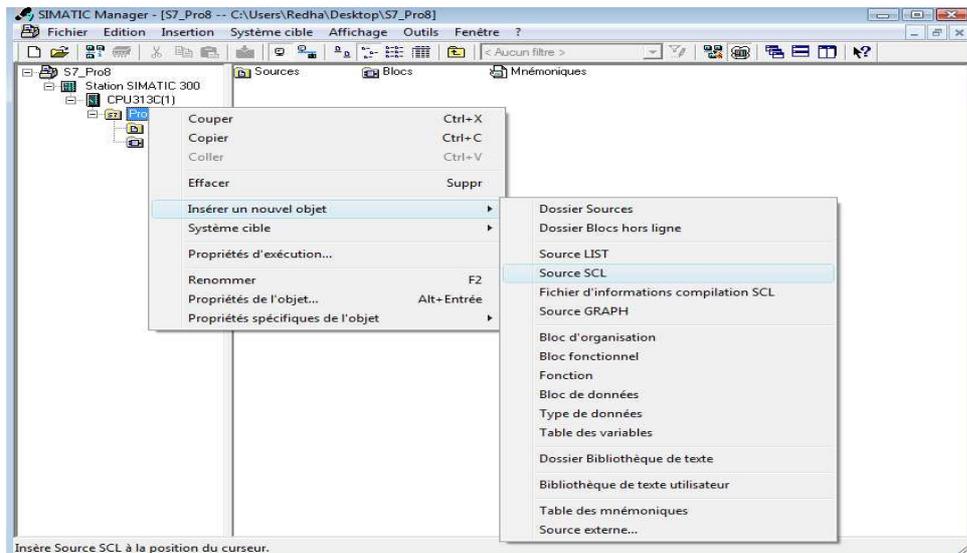


Figure II.18 : Création d'un fichier source SCL.

Un double clic sur l'objet crée ouvre l'application d'édition de programme SCL. De la, plusieurs choix relatifs au type de fichier que l'on veut créer, sont possibles. En cliquant sur insertion blocs, on pourra choisir entre :

- Bloc d'organisation.
- Bloc de fonction.
- Fonction.

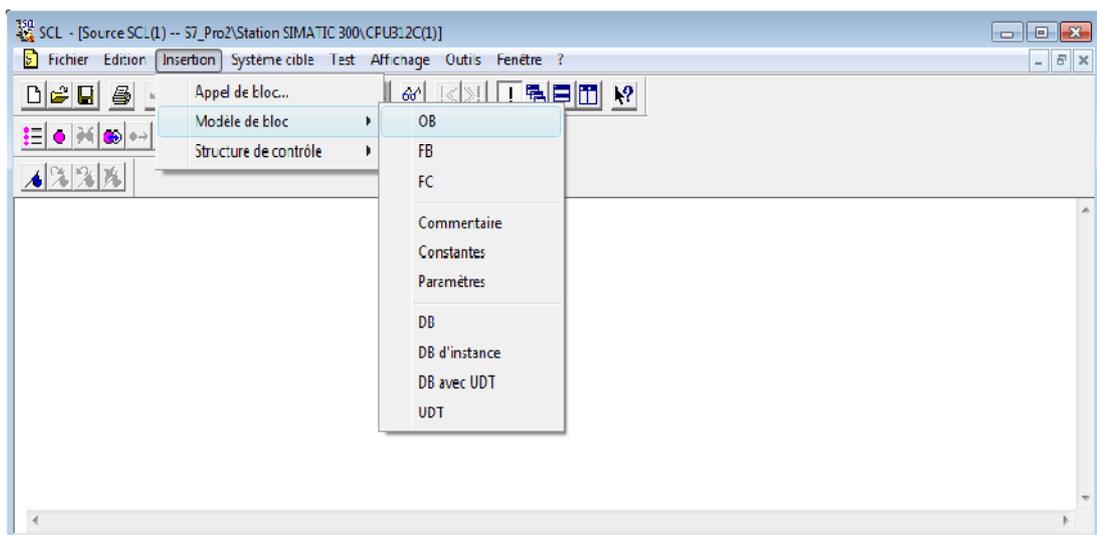


Figure II.19 : Choix du type de fichier SCL (1).

Suivant ce choix, un fichier de type OB'xxx' (respectivement FB'xxx' et FC'xxx') se créera automatiquement dans le projet courant.

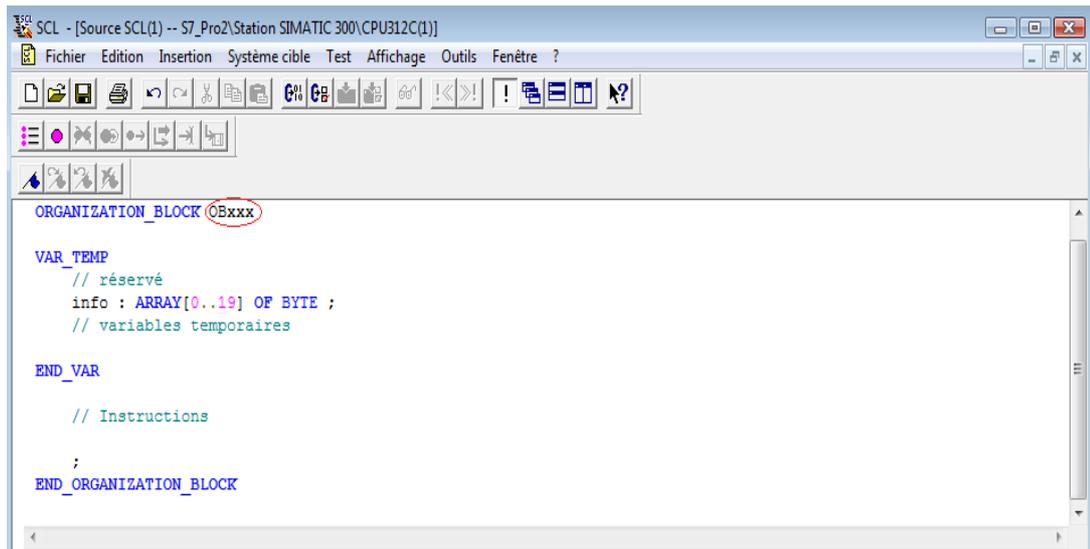


Figure II.20 : Choix du type de fichier SCL (2).

Exemple de l'illustration d'édition avec S7-SCL :

Dans ce qui va suivre nous allons montrer l'éditeur S7-SCL pour faire une simulation numérique qui a pour but :

- De trouver des solutions numériques aux équations différentielles difficiles à résoudre analytiquement.
- De prédire le comportement d'un système dans le temps.

Pour se faire nous allons traiter le problème de résolution d'un système d'équation.

Rappel :

Soit un système d'équation du type : $AX + B = 0$.

Pour résoudre ce type de système il suffit de trouver le vecteur :

$D\dot{X} = AX + B \Leftrightarrow \dot{X} = D^{-1}AX + D^{-1}B$. De manière à ce que la matrice $D^{-1}A$ soit éfinie négative, telle que $D^{-1}A$ est une matrice diagonale.

La définie négativité de $D^{-1}A$ entraîne la stabilisation du système ainsi constitué. En d'autres termes : $\lim_{t \rightarrow \infty} \dot{X} = 0 \Rightarrow D^{-1}AX + D^{-1}B$. De cette manière la solution du système d'origine est la valeur du vecteur d'état en régime permanent.

Pour notre application nous avons pris le système suivant :

$$\begin{cases} x + y = 3; \\ x + 2y = 2. \end{cases}$$

Ce système admet comme solution unique le couple $(x, y) = (4, -1)$.

Après calcul on obtient la représentation d'état suivante :

$$\begin{cases} \dot{x} = -x - y + 3; \\ \dot{y} = -x - 2y + 2. \end{cases}$$

Le problème devient maintenant un problème de résolution d'équations différentielles de premier ordre, pour se faire nous avons utilisé la méthode d'intégration de RANGE-KUTTA.

Le programme a été élaboré, comme suit, en utilisant :

- Un bloc fonction FC qui contient le programme principal.
- Un bloc OB1 qui permet d'appeler le bloc fonction FC.

```
FUNCTION FC2 : REAL
VAR_INPUT
a,b : INT;
x,y : REAL;
END_VAR
VAR
x1,y1,k1,k2,q1,q2 : REAL;
n,i : INT;
END_VAR
BEGIN
n:=1000*(b-a);
x1:=x;
y1:=y;
FOR i:= 1 TO n BY 1 DO
k1:=0.001*(-x1-y1+3);
q1:=0.001*(-x1-2*y1+2);
k2:=0.001*(-x1-k1/2-y1-q1/2+3);
q2:=0.001*(-x1-k1/2-2*y1-q1+2);
x1:=x1+0.5*(k1+k2);
y1:=y1+0.5*(q1+q2);
END_FOR;
FC2 := y1;
END_FUNCTION
```

Figure II.21 : le code SCL de l'exemple de résolution d'un système d'équation.

➤ **S7-GRAPH :**

Cette application permet l'édition des blocs fonctionnels. Cependant, elle ne permet pas d'éditer des fonctions, car elles n'ont pas des blocs de données associés pour mémoriser l'état des différentes transitions et étapes.

Exemple de l'illustration d'édition avec S7-GRAPH :

On dispose d'un réservoir ayant trois vannes (une de remplissage 'R' et deux vannes de vidange 'V1' et 'V2') et trois capteurs de niveau (L : niveau bas, N : niveau nominal, H : niveau haut). Ce système fonctionne suivant le cahier de charges suivant :

On appuyant sur le bouton poussoir 'm', on ouvre la vanne de remplissage 'R' jusqu'au point 'N', ensuite on ouvre la vanne 'V1'. Si le niveau atteint le point 'H' on ouvre la deuxième vanne 'V2' jusqu'à ce que le niveau atteigne le point 'L'.

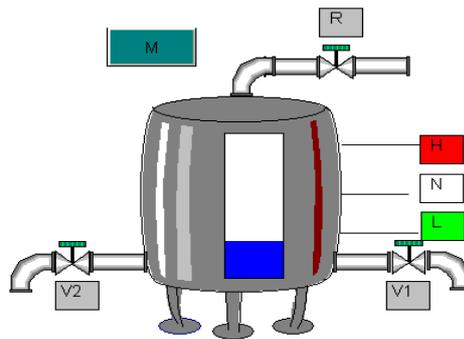


Figure II.22 : Le synoptique de l'exemple de remplissage.

Pour satisfaire ce cahier de charge, on commence par créer un bloc fonctionnel et un autre de données d'instance dans le dossier Blocs du SIMATIC Manager. On choisit le GRAPH comme langage de programmation et on lance l'application.

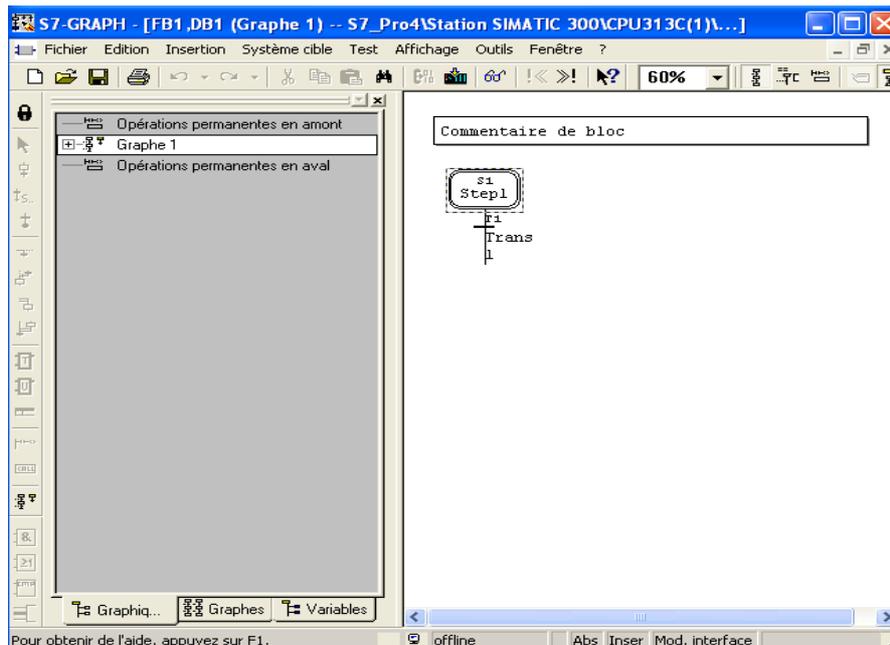


Figure II.23 : Editeur S7-GRAPH.

On remarque deux parties, la première est la zone d'édition, la deuxième inclut les éléments à ajouter (des transitions, des branchements, des étapes...).

Pour ajouter un élément à la zone d'édition, il suffit de cliquer sur l'emplacement souhaité puis sur l'élément.

Notre exemple comprend quatre étapes et quatre transitions. On commence par créer ces dernières, ensuite pour l'édition des actions et des transitions on change l'affichage de 'Graph' à 'Étape'. Pour ajouter une action, cliquer par le bouton droit de la souris sur la colonne à droite et choisir 'insérer un nouvel élément' puis sur 'action'. Il existe différents types d'actions. Le tableau suivant résume les plus utilisées :

Opération	Opérande	Signification
N	A, E, M, ou D	Tant que l'étape est active l'opérande est à 1
S	A, E, M, ou D	SET : dès que l'étape est active, l'opérande est mis à 1
R	A, E, M, ou D	RESET : dès que l'étape est active, l'opérande est mis à 0
D	A, E, M, ou D	Delay : n secondes après l'activation de l'étape, l'opérande se met à 1 pour la durée spécifiée
L	A, E, M, ou D	Impulsion : quand l'étape est active, l'opérande se met à 1 pendant n secondes

CALL	FB, FC, SFB ou SFC	Appel de blocs : tant que l'étape est active, le bloc spécifié est appelé
------	--------------------	---

Tableau II.3 : Les types d'actions du S7-GRAPH.

Pour les transitions, elles peuvent être présentées en langage CONT ou LOG. Cliquer sur la colonne à gauche et insérer des éléments (contacts...) depuis le catalogue.

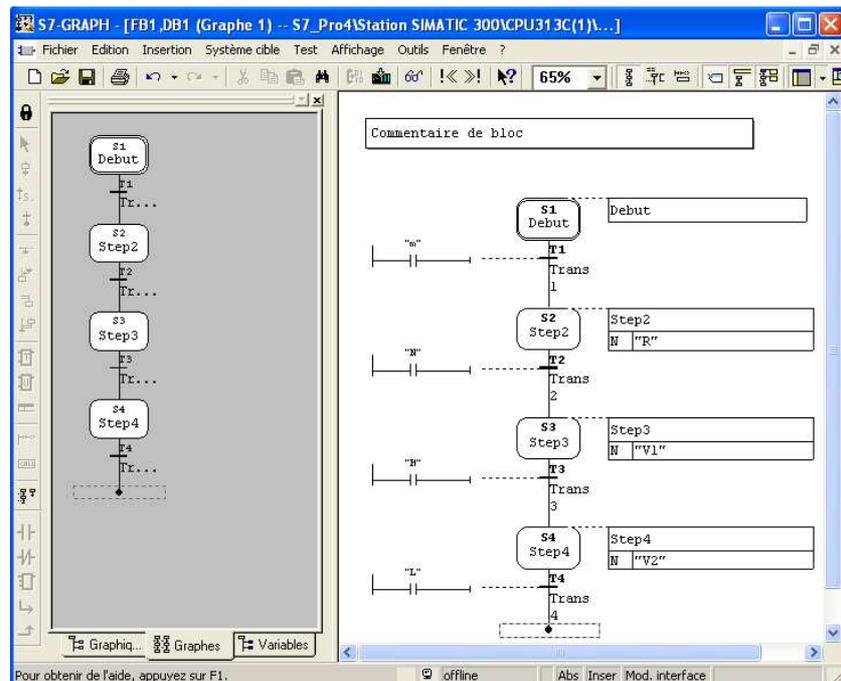


Figure II.24 : Le grafcet de l'exemple de remplissage.

En haut à gauche, on remarque deux autres éléments :

1. Verrouillage : l'Interlock, c'est une condition programmable servant à verrouiller une étape et influençant l'exécution de certaines actions.
2. Surveillance : Supervision, c'est une condition programmable servant à surveiller une étape et influençant l'évolution du graphe entre cette étape et la suivante.

II.2.2. Simulation des modules :

L'application de simulation de modules S7-PLCSIM permet l'exécution et le test du programme utilisateur destinés aux CPU S7-300 et aux CPU S7-400, ainsi qu'à WinLC. La simulation étant complètement réalisée au sein du logiciel STEP 7, il n'est pas nécessaire

qu'une liaison soit établie avec un matériel S7 quelconque. Lorsque S7-PLCSIM s'exécute, toute nouvelle liaison est automatiquement dirigée vers la CPU de simulation.

S7-PLCSIM dispose d'une interface simple permettant de visualiser et de forcer les différents paramètres utilisés par le programme (comme, par exemple, d'activer ou de désactiver des entrées).

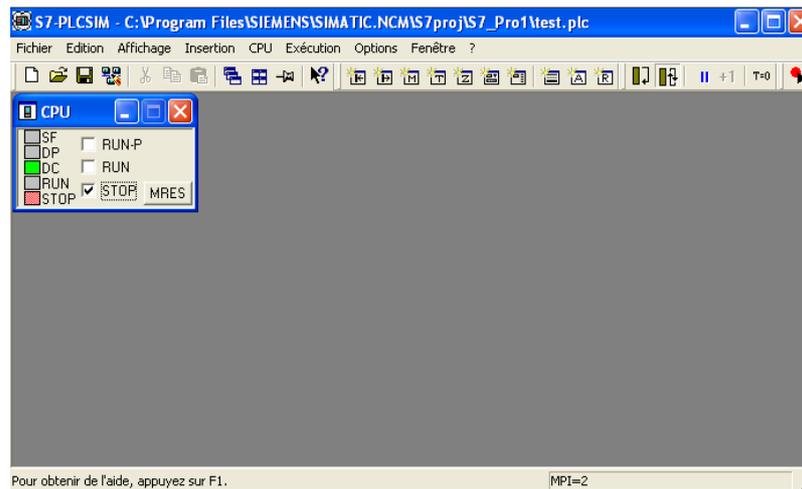


Figure II.25 : Interface de simulation de PLCSIM.

S7-PLCSIM possède les fonctions suivantes :

- On peut créer des "fenêtres" dans lesquelles on a la possibilité d'accéder aux zones de mémoire d'entrée et de sortie, aux accumulateurs ainsi qu'aux registres de la CPU de simulation. On peut également accéder à la mémoire par adressage symbolique (il faut juste charger la table des mnémoniques dans 'options', puis sur 'outils' 'insérer mnémoniques').
- On peut sélectionner l'exécution automatique des temporisations ou encore les définir et les réinitialiser manuellement.
- On a la possibilité de changer l'état de fonctionnement de la CPU (STOP, RUN et RUNP) comme pour une CPU réelle. De plus, on dispose d'une fonction de pause qui permet d'interrompre momentanément la CPU, sans affecter l'état du programme.

Bien que l'AP de simulation soit essentiellement logiciel, STEP 7 le considère comme une réelle composante matérielle, à quelques différences près :

- Contrairement à ce qui se passe avec une CPU réelle lors de la mise à l'arrêt de la CPU, l'état des sorties ne change pas.
- La CPU n'attend pas le début ou la fin du cycle pour actualiser une donnée qu'on a modifiée. Toute modification dans une fenêtre entraîne l'actualisation immédiate du contenu de l'adresse en mémoire.
- Les options d'exécution permettent de choisir le mode d'exécution du programme par la CPU :
 - ✓ La commande Cycle unique exécute un cycle du programme, puis attend qu'on démarre l'exécution du cycle suivant.
 - ✓ La commande Cycle continu exécute le programme de la même manière que dans un AP réel : elle démarre un nouveau cycle aussitôt que le cycle précédent est terminé.
- On peut déclencher manuellement les OB d'alarme (aller dans 'exécution' puis 'déclenchement OB Erreur').
- Les modules fonctionnels (FM) ne sont pas pris en charge.
- La communication d'égal à égal n'est pas possible.

Nous allons illustrer l'utilisation de S7-PLCSIM à travers un exemple : simulation d'un circuit RC par la méthode RK23 :

Soit le circuit suivant :

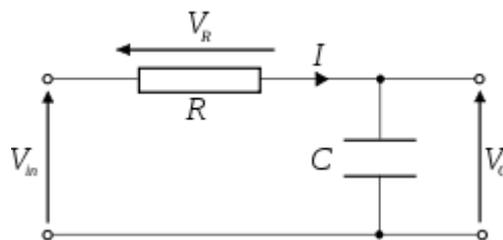


Figure II.26 : Circuit RC.

On a :

$$U = R \cdot I + V_c \quad \text{Et} : \quad I = C \frac{dV_c}{dt}$$

On obtient alors l'équation différentielle de premier ordre suivante :

$$RC \frac{dV_c}{dt} + V_c = U$$

Pour la simulation on prendra : $C=1 \mu\text{F}$, $R=10 \text{ K}\Omega$ et $U=20 \text{ volt}$ et $h=0.001 \text{ sec}$.
La méthode utilisée est celle RK23 :

$$V_{i+1} = V_i + \frac{1}{2}(k_1 + k_2) \quad \text{avec: } k_1 = h \cdot f(t_i, v_i) \text{ et } k_2 = h \cdot f(t_i + h, v_i + k_1).$$

On lance l'application S7-PLCSIM (aller dans 'Options' puis cliquer sur 'Simulation des modules') et on charge les blocs de code dans la CPU.

Dans le menu insertion, on ajoute le double mot de la sortie analogique, il faut sélectionner 'Réel' comme type sinon on se retrouve avec des résultats erronés.

On démarre la simulation en cochant la case RUN, et on relève la valeur de la tension de sortie en régime stationnaire comme représentée dans la figure qui suit :

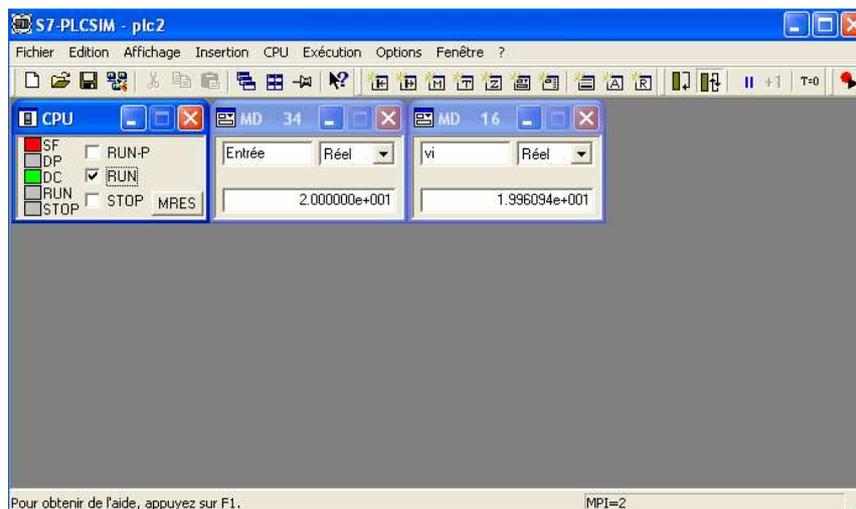


Figure II.27 : Résultat simulation du circuit RC.

Le programme correspondant à la simulation été placé en annexe. (Annexe E)

II.3 Régulation PID des processus à base d'automates SIMATIC S7 :

II.3.1 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.

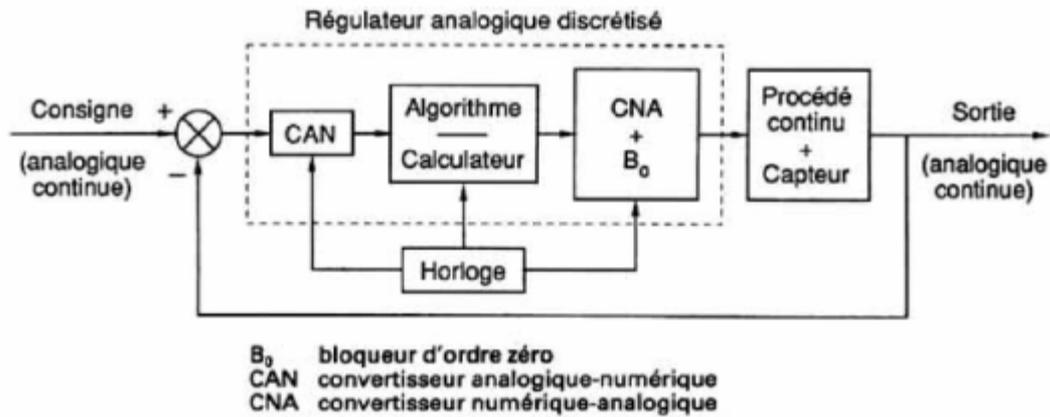


Figure II.28 : Schéma de principe de la régulation via un API.

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.

II.3.2 Concept de la régulation PID:

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

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

On peut combiner le bloc FB « PULSEGEN » avec le bloc FB « CONT_C » pour obtenir un régulateur à sortie sous forme d'impulsions pour organes de réglage proportionnels.

II.3.3 Régulation continue avec le bloc FB41 « CONT_C » :

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

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

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

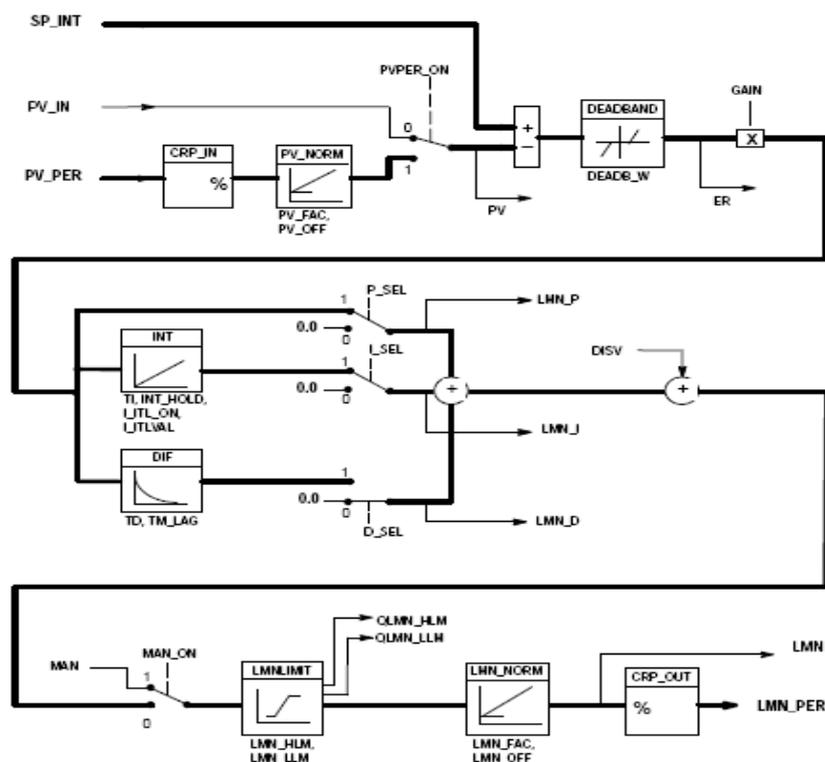


Figure II.29 : Schéma de principe du bloc SFB41 [6].

Les tableaux ci-dessous représentent les paramètres d'entrées et de sorties du bloc FB41 :

- 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	<p>MANUAL VALUE ON / Activation du mode manuel</p> <p>Quand cette entrée est à 1, la boucle de régulation est interrompue. La valeur de réglage manuelle est sortie comme grandeur de réglage.</p>
PVPER_ON	BOOL		FALSE	<p>PROCESS VARIABLE PERIPHERY ON / Activation de la mesure de périphérie</p> <p>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>
P_SEL	BOOL		TRUE	<p>PROPORTIONAL ACTION ON / Activation de l'action proportionnelle</p> <p>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.</p>
I_SEL	BOOL		TRUE	<p>INTEGRAL ACTION ON / Activation de l'action par intégration</p> <p>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.</p>
INT_HOLD	BOOL		FALSE	<p>INTEGRAL ACTION HOLD / Gel de l'action par intégration</p> <p>La sortie de l'intégrateur peut être gelée. Pour cela, il faut mettre à 1 cette entrée.</p>
I_ITL_ON	BOOL		FALSE	<p>INITIALIZATION OF THE INTEGRAL ACTION / Initialisation de l'action par intégration</p> <p>La sortie de l'intégrateur peut être positionnée sur la valeur initiale I_ITL_VAL. Pour cela, il faut mettre à 1 cette entrée.</p>

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	>= 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 II.4 : Entrées du bloc SFB41 [6].

- Paramètres de sortie :

Paramètre	Type de données	Valeurs admises	Par défaut	Description
LMN	REAL		0.0	MANIPULATED VALUE / Valeur de réglage Cette sortie donne en virgule flottante la valeur de réglage agissant réellement
LMN_PER	WORD		W#16#0000	MANIPULATED VALUE PERIPHERY / Valeur de réglage de

				périphérie Cette sortie fournit la valeur de réglage en format de périphérie
QLMN_HLM	BOOL		FALSE	HIGH LIMIT OF MANIPULATED VALUE REACHED / Grandeur de réglage à la limite supérieure La valeur de réglage est toujours bornée à une limite supérieure et à une limite inférieure. Cette sortie signale le dépassement de la limite supérieure
QLMN_LLM	BOOL		FALSE	LOW LIMIT OF MANIPULATED VALUE REACHED / Grandeur de réglage à la limite inférieure La valeur de réglage est toujours bornée à une limite supérieure et à une limite inférieure. Cette sortie signale le dépassement de la limite inférieure
LMN_P	REAL		0.0	PROPORTIONALITY COMPONENT / Composante P Cette sortie contient la composante proportionnelle de la grandeur de réglage.
LMN_I	REAL		0.0	INTEGRAL COMPONENT / Composante I Cette sortie contient la composante intégrale de la grandeur de réglage.
LMN_D	REAL		0.0	DERIVATIVE COMPONENT / Composante D Cette sortie contient la composante différentielle de la grandeur de réglage.
PV	REAL		0.0	PROCESS VARIABLE / Mesure Cette sortie donne la mesure effective
ER	REAL		0.0	ERROR SIGNAL / Signal d'erreur Donne le signal d'erreur effectif

Tableau II.5 : Sorties du bloc SFB41 [6].

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

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

II.3.4 Paramétrage du régulateur PID du type continue FB41 avec STEP 7 :

Pour l'utilisation et l'optimisation du fonctionnement du régulateur PID FB, on fait appel à l'outil « Paramétrage de la régulation PID », les paramètres choisis seront sauvegardés dans le DB d'instance local associée au bloc FB41.

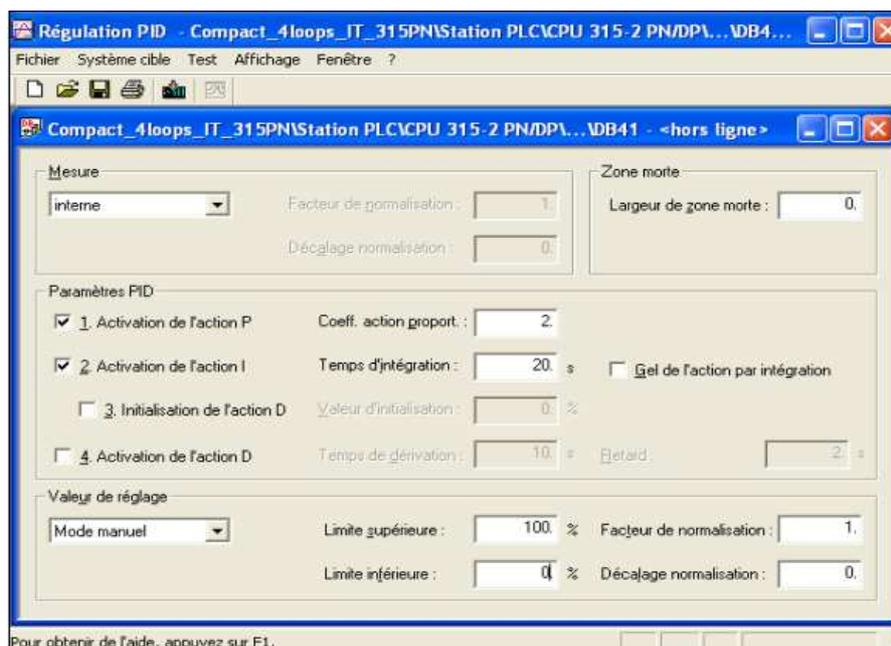


Figure II.30 : Interface de paramétrage PID [7].

Une fois le bloc de données paramètre, sauvegardé et chargé dans la CPU, c'est alors que peut intervenir la mise en route de la représentation graphique pour observer le comportement de la régulation.

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

Conclusion :

Munis d'un logiciel très performant, les automates programmables de la série S7-300 forment des unités de traitement de commande d'une grande flexibilité. En effet, simple à utiliser et doté d'une interface graphique très intuitive, le step7 permet d'exploiter de manière très optimale les différentes CPU de la gamme.

Ce chapitre nous a permis de voir les différentes possibilités offertes par les automates programmables : simulation des systèmes séquentiels et continus, calcul numérique.

CHAPITRE III

WinCC, logiciel de conception des interfaces HMI

Introduction :

Pour les applications au pied de la machine et du process dans la construction de machines et d'installations, SIMATIC HMI propose le logiciel d'ingénierie et de supervision SIMATIC WinCC. Ce dernier est un système de supervision de processus modulable qui offre des fonctions performantes de surveillance d'automatismes.

En effet, l'utilisateur de ce logiciel dispose de moyens de visualisation de process (synoptiques, des graphes, des baragrapes...) qui lui permettent de contrôler facilement et avec clarté toutes les opérations d'automatisation.

III.1 Présentation de WinCC :

III.1.1 Description générale :

WinCC est un système HMI performant qui est utilisé sous Microsoft Windows 2000 et Windows XP. HMI signifie "Human Machine Interface", il s'agit donc de l'interface entre l'homme (l'utilisateur) et la machine (le process). Le contrôle proprement dit du process est assuré par les automates programmables (API). Une communication s'établit donc entre WinCC et l'opérateur d'une part et entre WinCC et les automates programmables d'autre Part.

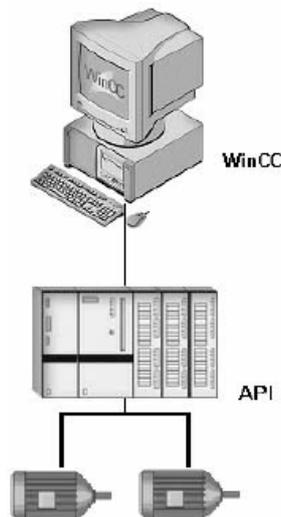


Figure III.1 : Schéma de communication entre le WinCC et l'API.

WinCC permet de visualiser le process et de concevoir l'interface utilisateur graphique destinée à l'opérateur.

- WinCC permet à l'opérateur de surveiller le process. Pour ce faire, le process est visualisé par un graphisme à l'écran. Dès qu'un état du process évolue, l'affichage est mis à jour.
- WinCC permet à l'opérateur de commander le process. A partir de l'interface utilisateur graphique ; il peut par exemple entrer une valeur de consigne ou ouvrir une vanne.
- Lorsqu'un état de process devient critique, une alarme est déclenchée automatiquement. L'écran affiche une alarme en cas de franchissement d'un seuil défini par exemple.
- Les alarmes et valeurs de process peuvent être imprimées et archivées sur support électronique par WinCC. Ceci vous permet de documenter la marche du process et d'avoir accès ultérieurement aux données de production du passé.

III.1.2 Fonctionnement du WinCC :

WinCC est un système modulaire. Ses éléments de base sont le logiciel de configuration (CS) et le logiciel runtime (RT) :

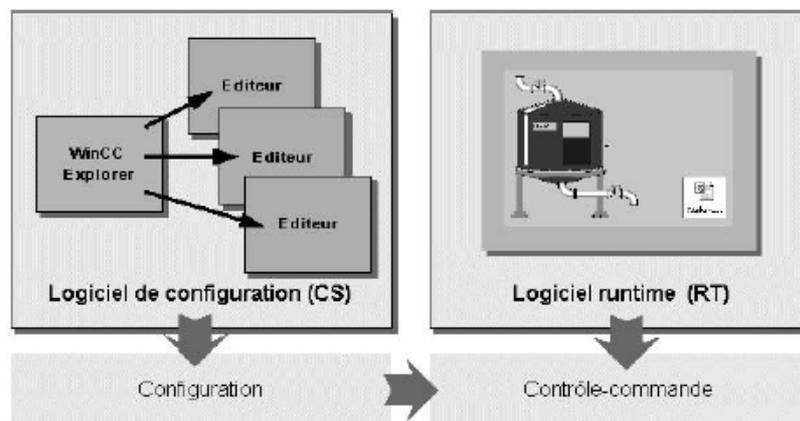


Figure III.2 : Structure interne du WinCC.

a. Logiciel de configuration (CS) :

Après démarrage de WinCC, l'écran affiche l'explorateur WinCC Explorer. WinCC Explorer est le noyau du logiciel de configuration. WinCC Explorer affiche la structure complète du projet et permet de gérer le projet.

Chapitre III : WinCC, logiciel de conception des interfaces HMI

La configuration s'effectue à l'aide d'éditeurs spécifiques que vous pouvez ouvrir à partir de WinCC Explorer. Chaque éditeur permet de configurer un sous-système particulier de WinCC.

Les principaux sous-systèmes de WinCC sont :

- Le système graphique – l'éditeur pour la réalisation des vues s'appelle Graphics Designer ;
- Le système de signalisation – l'éditeur permettant de configurer les alarmes s'appelle Alarm Logging ;
- Le système d'archivage – l'éditeur permettant de spécifier les données à archiver s'appelle Tag Logging ;
- Le système de « journalisation » – l'éditeur permettant de réaliser la mise en page des journaux, s'appelle Report Designer ;
- Le gestionnaire des utilisateurs – l'éditeur de gestion des utilisateurs et des droits d'utilisateur s'appelle User Administrator ;
- La communication – elle se configure directement sous WinCC Explorer.

Toutes les données de configuration sont enregistrées dans la base de données CS.

b. Logiciel runtime :

Le logiciel runtime permet à l'opérateur d'assurer la conduite du process. Les tâches incombant au logiciel runtime sont les suivantes :

- Lecture des données enregistrées dans la base de données CS ;
- Affichage des vues à l'écran ;
- Communication avec les automates programmables ;
- Archivage des données actuelles de runtime, par exemple des valeurs de process et événements de signalisation ;
- Conduite du process, par exemple spécification de consignes, mise en marche/arrêt.

III .2 Etapes de conception d'un projet sous WinCC pas à pas :

Nous allons aborder dans cette partie les différentes étapes pour aboutir à un projet et ceci en expliquant chaque étape [8].

a. Création d'un projet :

Après avoir lancé le SIMATIC WinCC Explorer, on clique sur : 'Fichier 'nouveau' ; une fenêtre de dialogue affiche alors les différents types de projets



Figure III.3 : Boite de dialogue de création de projets.

On choisira pour notre exemple 'Projet monoposte' qu'on nommera 'Qckstart'. L'Explorateur WinCC est représenté par la figure ci-dessous :

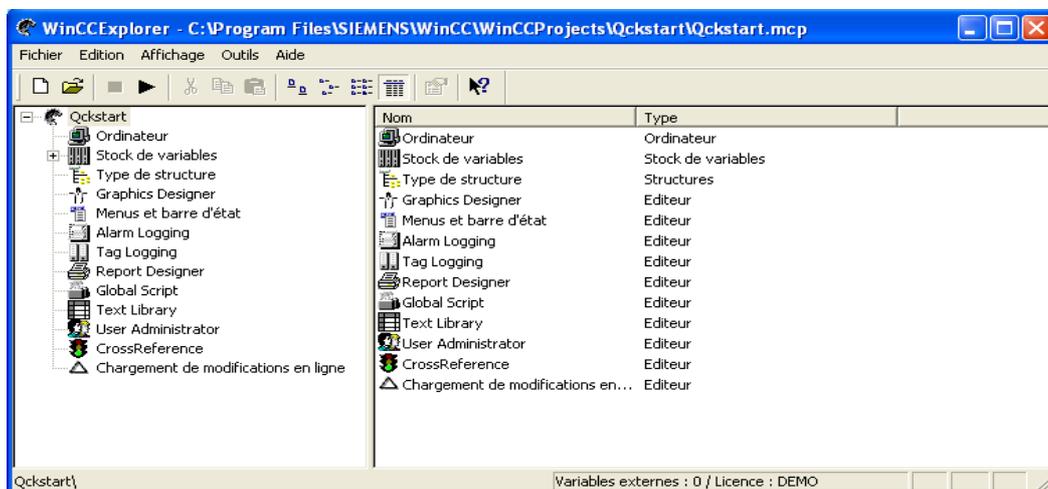


Figure III.4 : Explorateur WinCC.

b. Ajout d'un pilote d'API :

Nous allons maintenant configurer notre système pour que notre API puisse communiquer avec WinCC par l'intermédiaire du pilote de communication sélectionné. Le pilote à sélectionner dépend de l'API exploité. La famille d'API SIMATIC gère de quelques centaines à plusieurs milliers de points E/S.

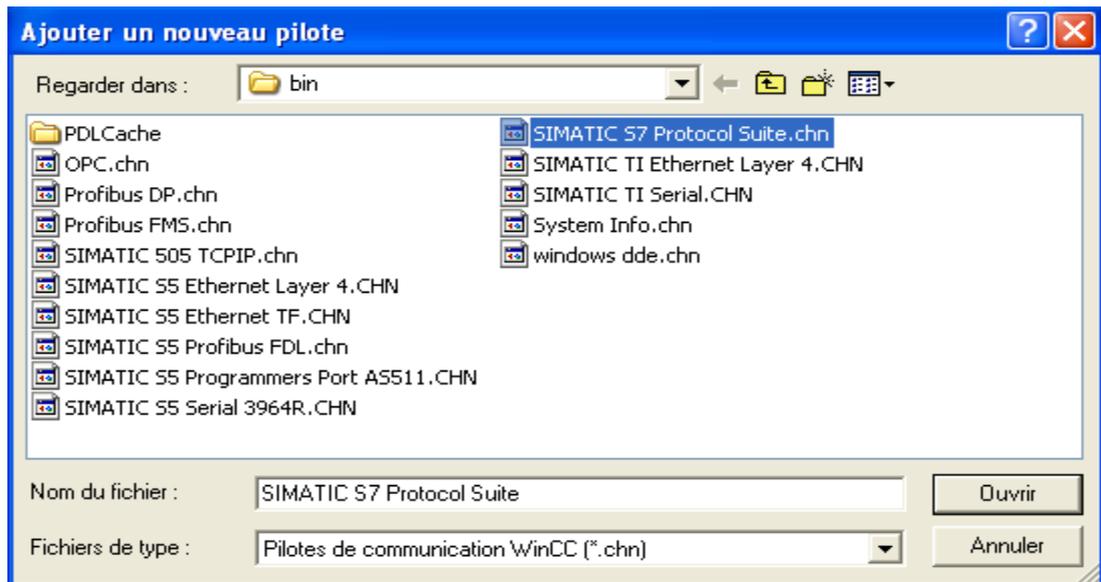


Figure III.5 : Liste des différents types de pilotes de communication.

Parmi les pilotes qui seront affichés, on choisit pour notre exemple ‘SIMATIC S7 Protocole Suite’.

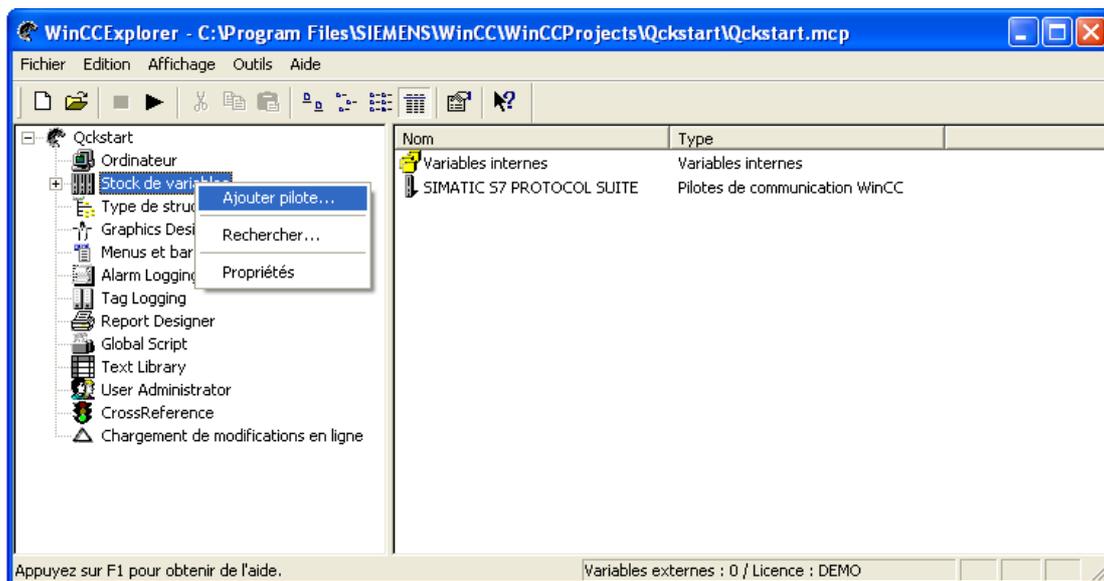


Figure III.6 : Ajout du pilote d'un API.

On crée ensuite une liaison de type ‘MPI’ qu’on nommera ‘API1’ comme suit :

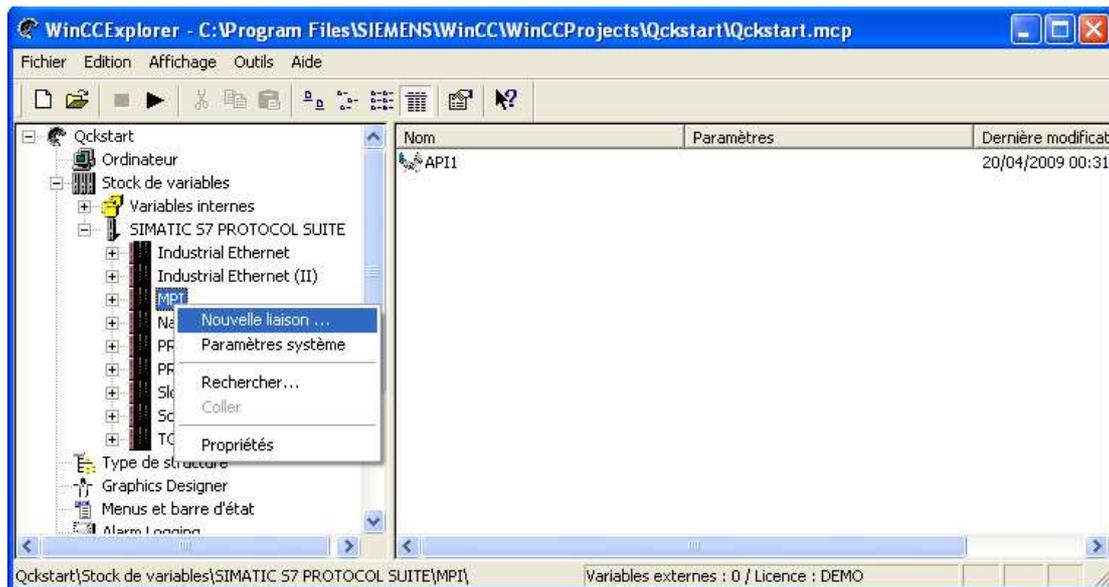


Figure III.7 : Création de la liaison MPI.

c. Définition des variables :

Les variables utilisées dans WinCC représentent soit des valeurs réelles, comme par exemple le taux de remplissage d'un réservoir d'eau, soit des valeurs internes, qui sont calculées où bien simulées en interne par WinCC :

- Variables externes :

Les variables externes sont des emplacements en mémoire d'un API ou d'un matériel semblable. Ainsi le niveau de remplissage du réservoir d'eau est relevé par un capteur de niveau et enregistré dans l'API. Le taux de remplissage est communiqué à WinCC par le canal de communication.

Lors de la configuration de variables externes, tous les types de données numériques doivent faire l'objet d'une adaptation de format.

- Sur l'API : p. ex. pour certaines fonctions (telles que les valeurs de temporisateur / affichages code BCD) ou par l'information à adresser (p. ex. adresse d'octet ou de mot dans le bloc de données ou dans la zone d'E/S).
- Dans WinCC : p.ex. pour le traitement de valeurs analogiques ou des calculs

Chapitre III : WinCC, logiciel de conception des interfaces HMI

Le type de données détermine le format des données côté WinCC. L'adaptation de format définit la conversion du format WinCC au format de l'API. Cette définition est valable pour les deux sens de la communication.

Le tableau qui suit liste des différents types de variables ainsi que la possibilité d'adaptation de ces derniers :

Type de données	Adaptation de format
Variable binaire	Non
Valeur 8 bits non signée	Oui
Valeur 8 bits signée	Oui
Valeur 16 bits non signée	Oui
Valeur 16 bits signée	Oui
Valeur 32 bits non signée	Oui
Valeur 32 bits signée	Oui
Nombre à virgule flottante 32- Bit IEEE 754	Oui
Nombre à virgule flottante 64 bits IEEE 754	Oui
Variable texte jeu de caractères 8 bits	Non
Variable texte jeu de caractère 16 bits	Non
Type données brutes	Non

Tableau III.1 : différents types de données.

La création d'une variable externe se fait comme suit :

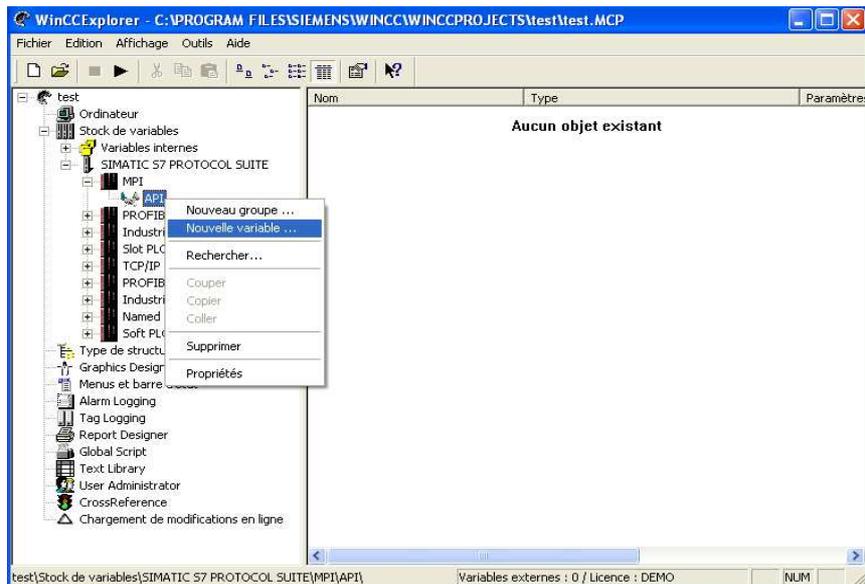


Figure II.8 : Création d'une variable externe.

Aussi, il faut attribuer une adresse à la variable externe et cette dernière doit être la même que celle définie dans l'automate :

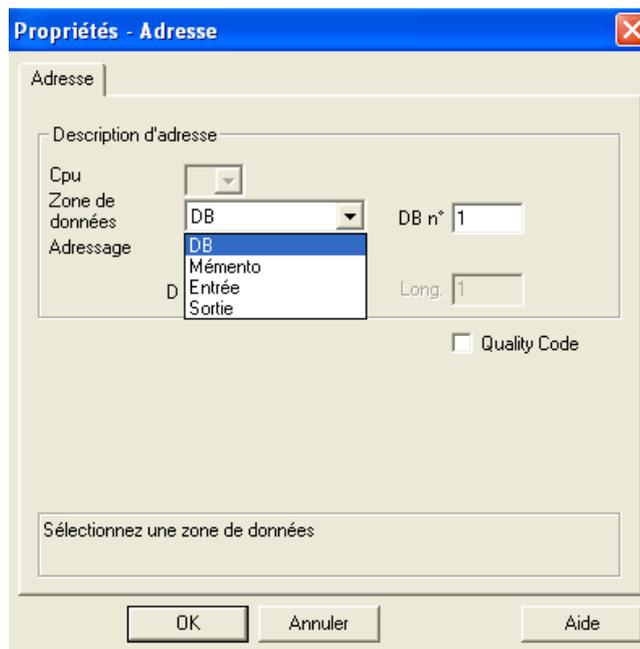


Figure II.9 : Adressage de la variable externe.

La variable externe peut être : une entrée, une sortie, un memento ou encore lue à partir d'un bloc de données.

- Variables internes :

Les variables internes sont des emplacements en mémoire de WinCC qui assurent les mêmes fonctionnalités qu'un API. Elles peuvent être calculées et modifiées en interne par WinCC.

Les groupes de variables servent à structurer les variables. Toutes les variables peuvent, pour plus de clarté, être rangées dans des groupes de variables.

Pour notre exemple nous nous contenterons d'utiliser une variable interne comme suit :

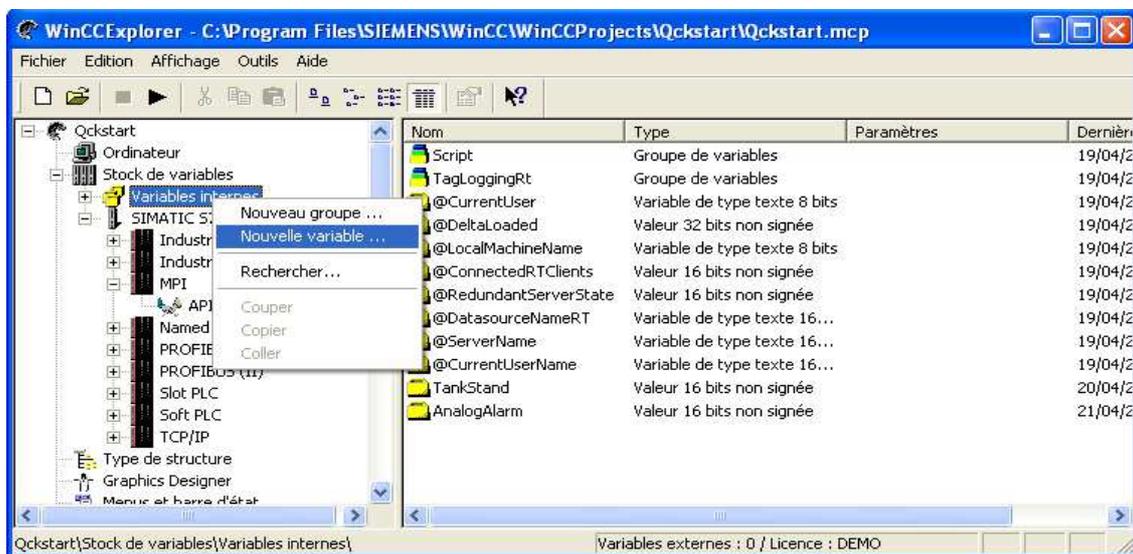


Figure III.10 : Création d'une variable interne.

On nommera la variable 'TankStand' et on choisit le type '16 bits non signée' comme le montre la figure ci-contre :

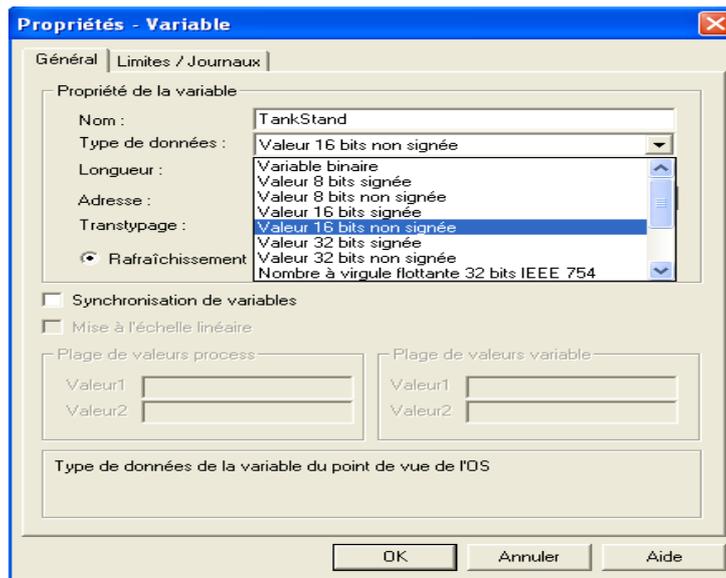


Figure III.11 : Choix du type et du nom de la variable.

d. Création d'une vue de process :

Dans cette partie nous allons configurer une vue de process pour notre projet et que nous allons illustrer à travers les figures qui suivent :

- On commence d'abord par la création d'une nouvelle vue à l'aide du 'Graphics Designer' qu'on nommera : 'START'.

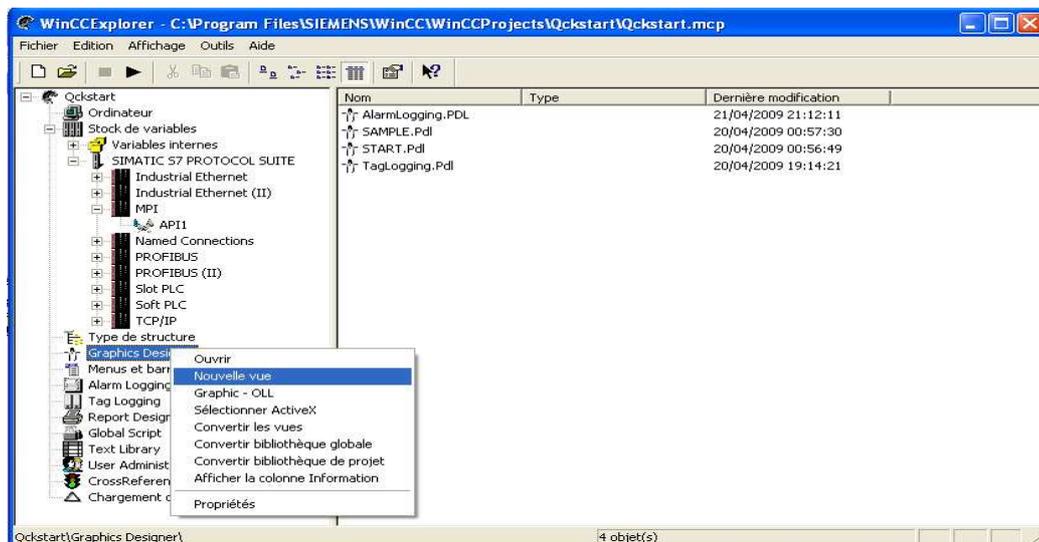


Figure III.12 : création d'une vue grâce au Graphics Designer (GD).

Chapitre III : WinCC, logiciel de conception des interfaces HMI

- On ouvre la vue ainsi créée pour y ajouter les différents composants constituant notre interface. Ci-contre détaille les différentes parties de Graphics Designer.

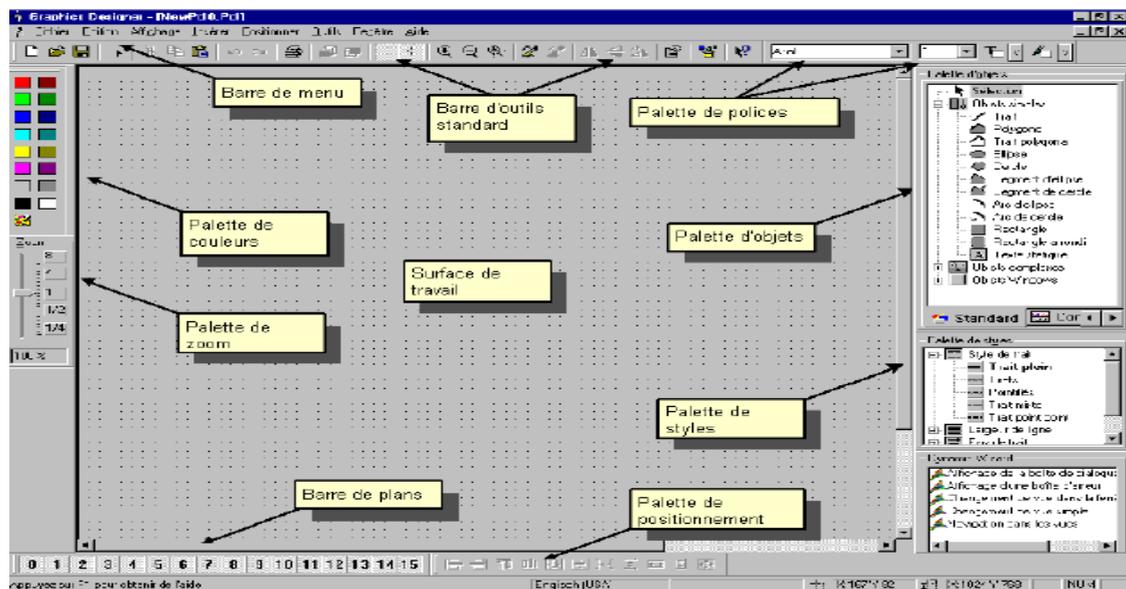


Figure III.13 : Vue détaillée de l'interface de GD.

- On ajoute les différents composants à partir de la bibliothèque : tuyaux, vannes, une cuve et champs E/S pour afficher le niveau de la cuve. Ce dernier sera dynamiser par notre variable interne 'TankStand'. On obtient lors la vue suivante :

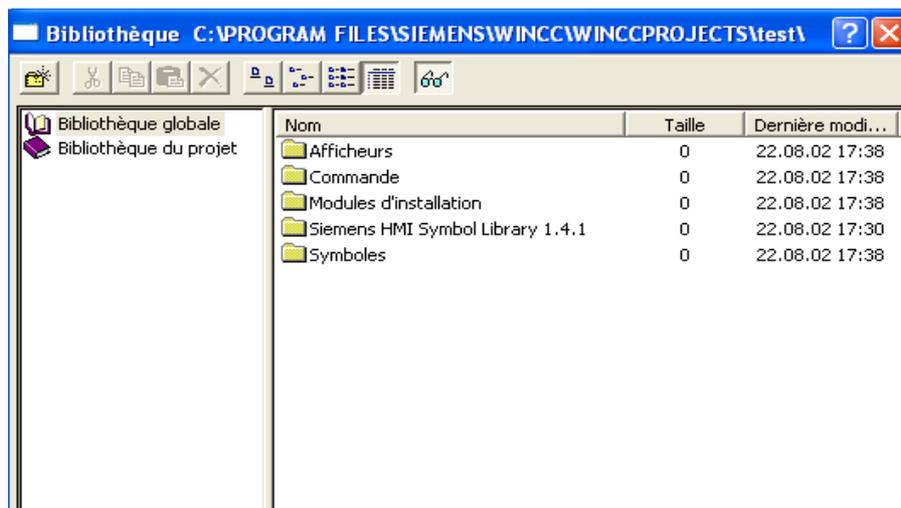


Figure III.14 : Liste des composants de la bibliothèque.

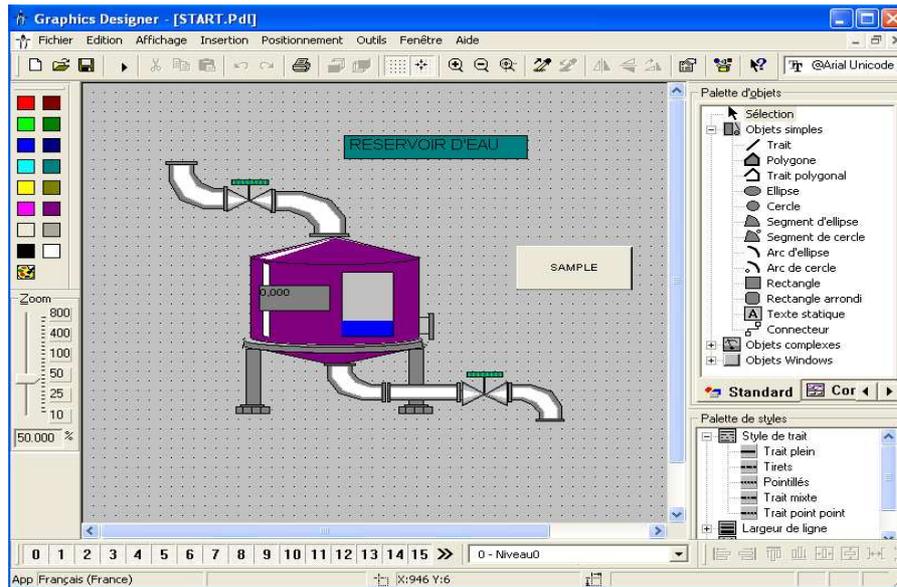


Figure III.15 : Vue de process obtenue par le GD.

e. Dynamisation de la vue de process:

La dynamisation de la vue se fait en affectant une des propriétés d'un objet à une variable mais aussi en affectant une variable à un champ d'E/S. On distingue trois parties :

- La partie 1 pour la détermination des valeurs de seuil (pour notre exemple : Min 0, Max 100).
- La partie 2 pour le choix de la variable ('TankStand') affectée à l'objet (réservoir).
- La partie 3 pour le choix de la valeur de rafraîchissement (pour notre exemple 2 sec).

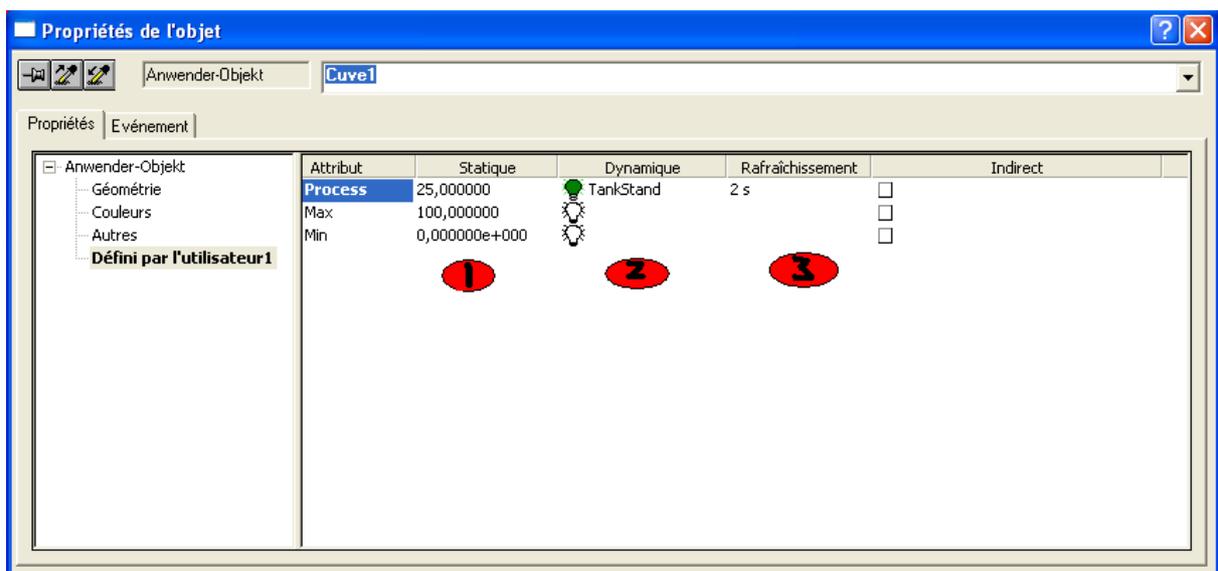


Figure III.16 : Dynamisation d'un objet.

f. Simulation de la vue de process :

On configure tout d'abord le Graphics-Runtime comme suit :

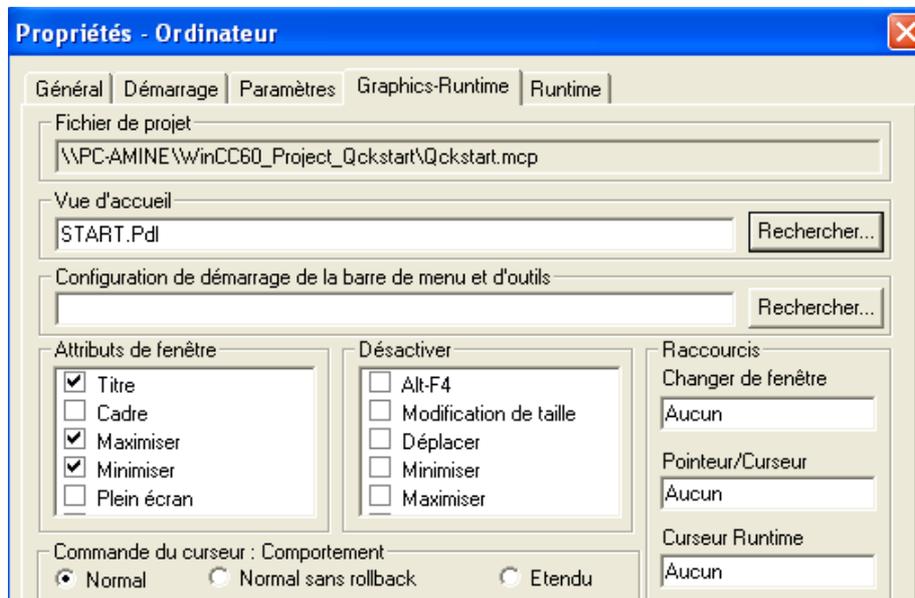


Figure III.17 : Configuration de Graphics-Runtime.

On obtient finalement la vue suivante ou le niveau d'eau de la cuve correspond à la valeur affiché sur le champ E/S :

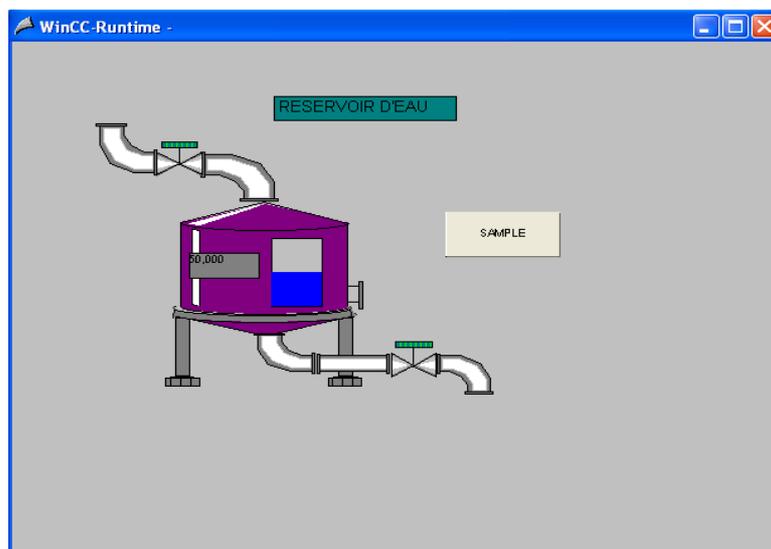


Figure III.18 : Vue de process.

III.3 Affichage des valeurs de process :

Dans cette partie nous allons présenter les composants de base pour l'affichage des valeurs de process ainsi que les variables d'archives à savoir les courbes et les tableaux :

- La fenêtre de courbe permet d'afficher des variables de process sous forme de courbe.
- La fenêtre de tableau permet d'afficher des variables de process sous forme de tableau.

Aussi, nous allons détailler la procédure à suivre pour ajouter ces deux éléments et les afficher au mode Runtime.

On commence tout d'abord par lancer le Tag Logging et qui est composé de trois parties, comme le montre la figure ci-dessous :

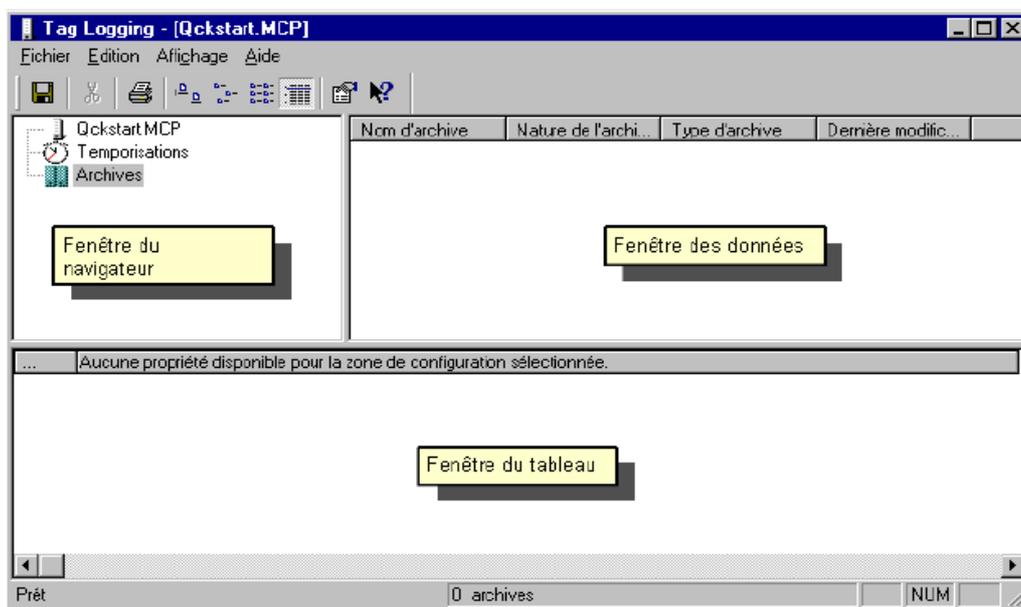


Figure III.19 : Différentes parties du Tag Logging.

Après avoir lancé la Tag Logging, on crée une archive qu'on nommera 'TankStand_Archive' à travers deux étapes comme suit:

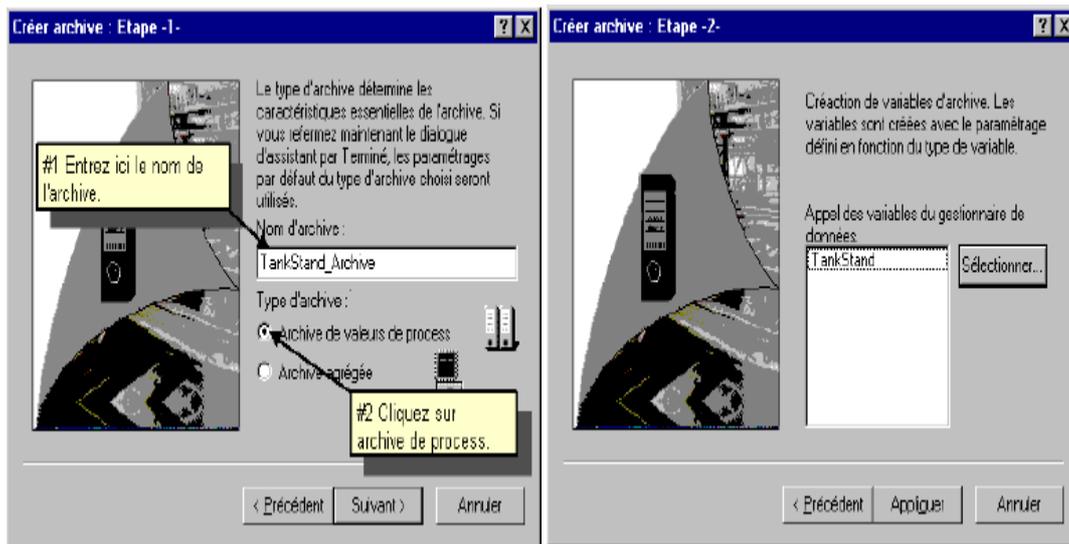


Figure III.20 : Etapes de la création de l'archive.

Les valeurs de la variable de process seront enregistrées dans la variable d'archive 'TankStand_Arch' avec un intervalle d'une seconde.

Une fois la variable d'archive créée, nous passons à la création d'une fenêtre de courbe et une autre de tableau. Pour cela on doit créer au préalable une nouvelle vue qu'on nommera 'TagLogging' dans laquelle on ajoutera 'WinCC Online Trend Control' pour la courbe et 'WinCC Online Table Control' pour le tableau et ceci à partir de la palette d'objet du Graphics Designer. Le paramétrage de ces deux éléments est identique et se fait d'une part en nommant l'objet et d'autre part en sélectionnant la variable d'archive comme suit :

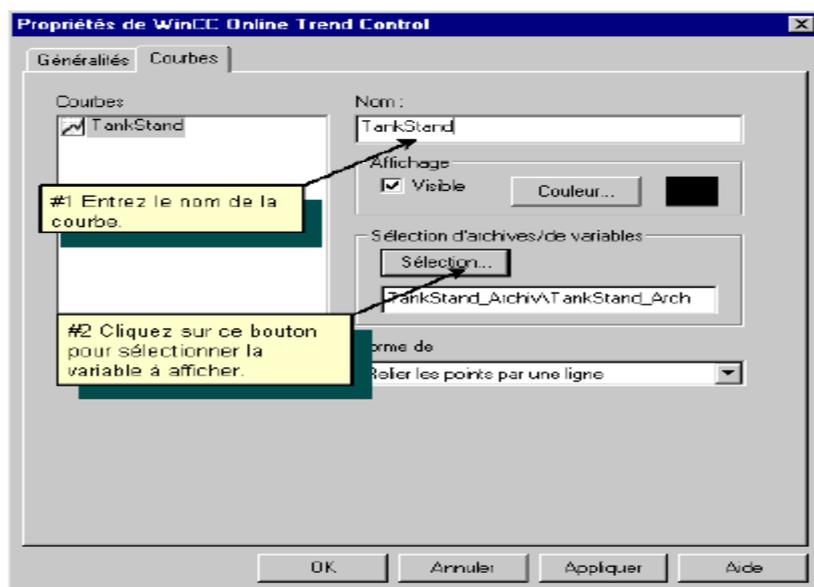


Figure III.21 : Paramétrage de la fenêtre de courbe et de tableau.

En activant le projet à l'aide du Runtime on obtient la simulation suivante :

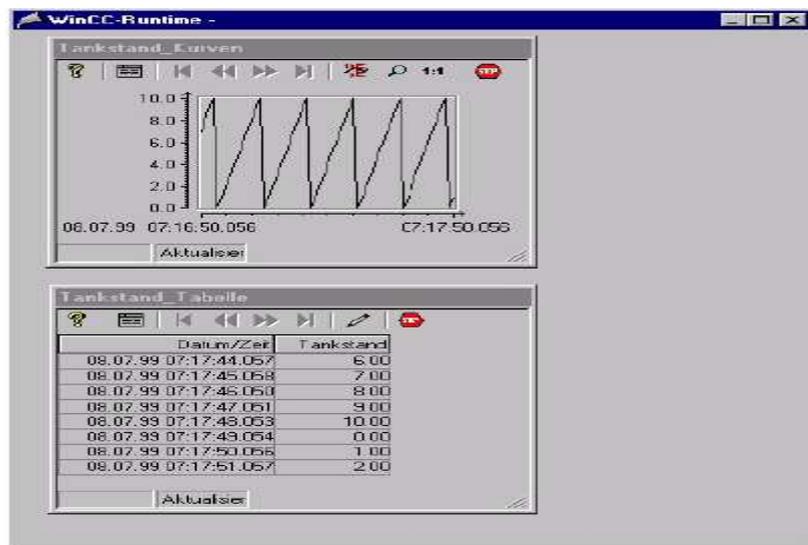


Figure III.22 Courbe et tableau au mode Runtime.

III.4 Configuration des alarmes :

Dans cette partie nous nous intéressons à la procédure de création d'un système d'alarmes pour le Runtime.

Le système d'alarmes informe l'opérateur des états de fonctionnement ou pannes du process et il est créé à l'aide de l'éditeur Alarm Logging.

Les alarmes configurées dans Alarm Logging sont sorties dans Runtime lorsque l'événement correspondant se produit, par exemple un défaut ou un dépassement de seuil.

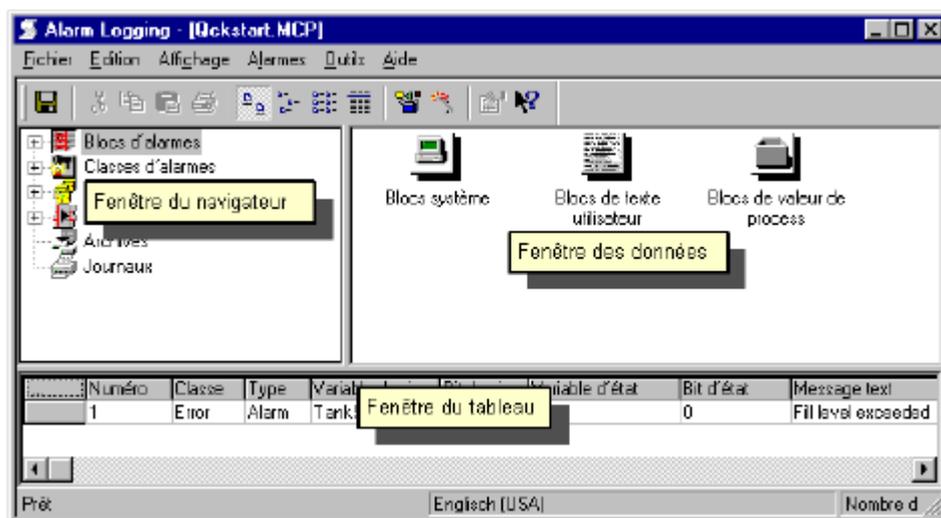


Figure III.23 : L'éditeur Alarm Logging.

Chapitre III : WinCC, logiciel de conception des interfaces HMI

Dans la partie ‘Bloc d’alarmes – Blocs de texte utilisateur’, on crée des alarmes pour lesquelles on doit définir les paramètres suivants :

- Variable de signalisation ;
- Bit de signalisation ;
- Texte d’alarme ;
- Localisation de l’incident ;
- Choix des couleurs pour l’état de l’alarme (arrivée, partie ou acquittée) dans la partie ‘Classes d’alarmes-Incident’.

Pour notre exemple, nous avons défini trois alarmes associées à notre variable de process ‘TankStand’ comme le montre la figure ci-dessous :

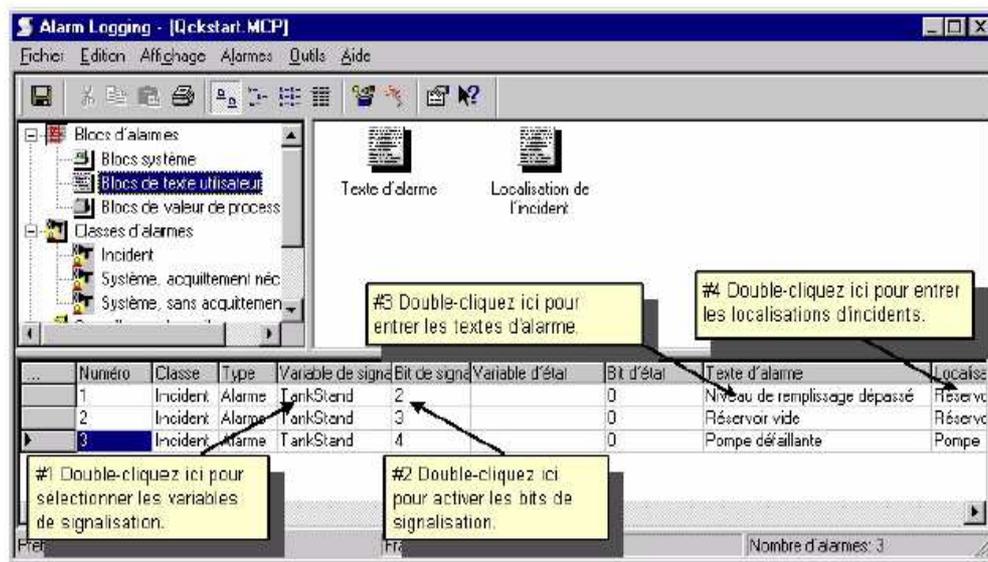


Figure III.24 : Configuration des propriétés des alarmes.

L’éditeur Alarm Logging nous permet aussi de définir de configurer la surveillance de seuils et ainsi surveiller le dépassement de seuils pas des variables.

On commence d’abord par créer une variable dans la partie ‘Surveillance de seuil’ pour laquelle on définit le seuil supérieur ainsi que le seuil inférieur (pour notre exemple cette variable est nommée ‘AnalogAlarm’).

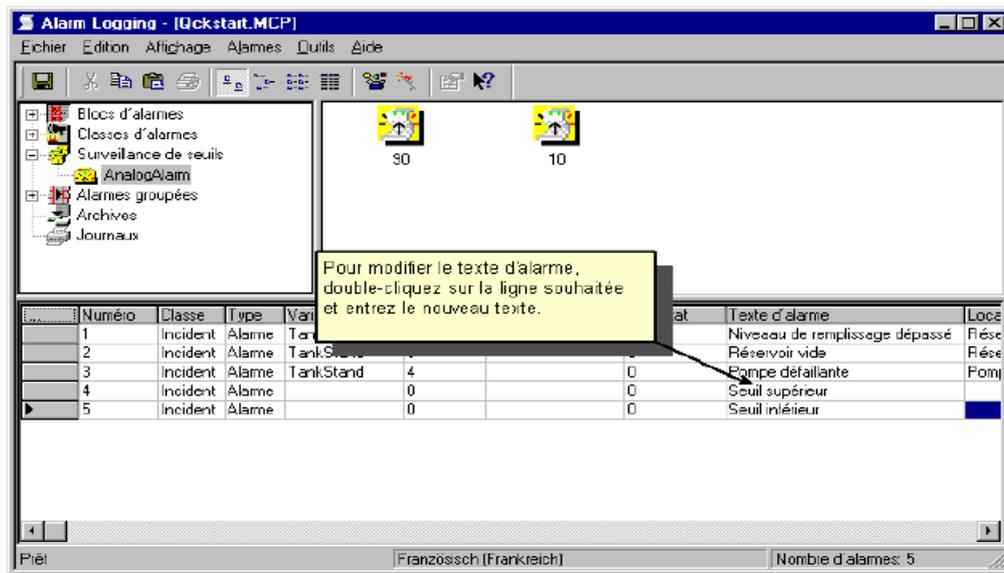


Figure III.25 : Configuration des alarmes de seuils.

L'affichage de ces alarmes se fait dans un tableau et ceci en ajoutant à une vue l'élément 'WinCC Alarm Control'. Le résultat obtenu est le suivant :

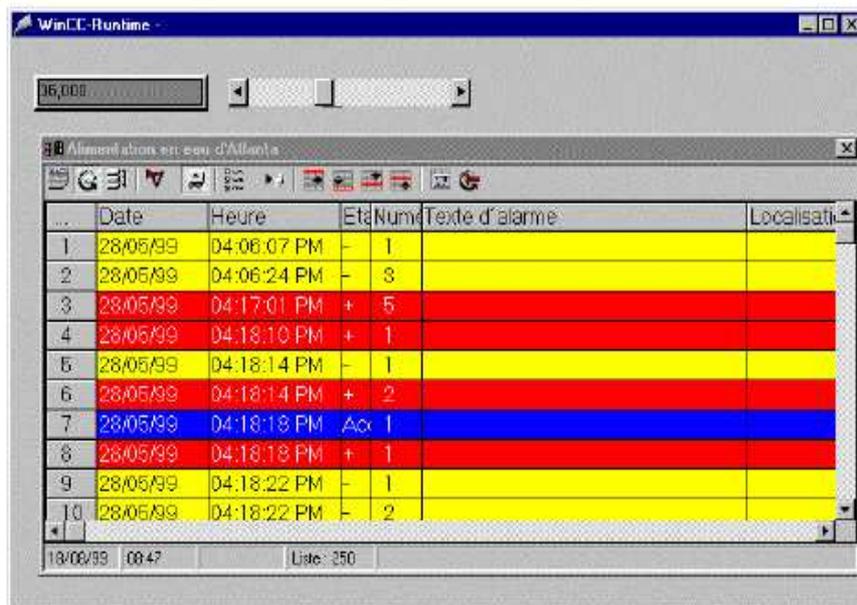


Figure III.26 : Fenêtre d'alarmes au runtime.

III.5 Création de fonctions et d'actions (Global Script) :

Le Global Script est l'éditeur dans lequel on peut créer des fonctions et des actions dans le but de dynamiser l'environnement WinCC Runtime [9].

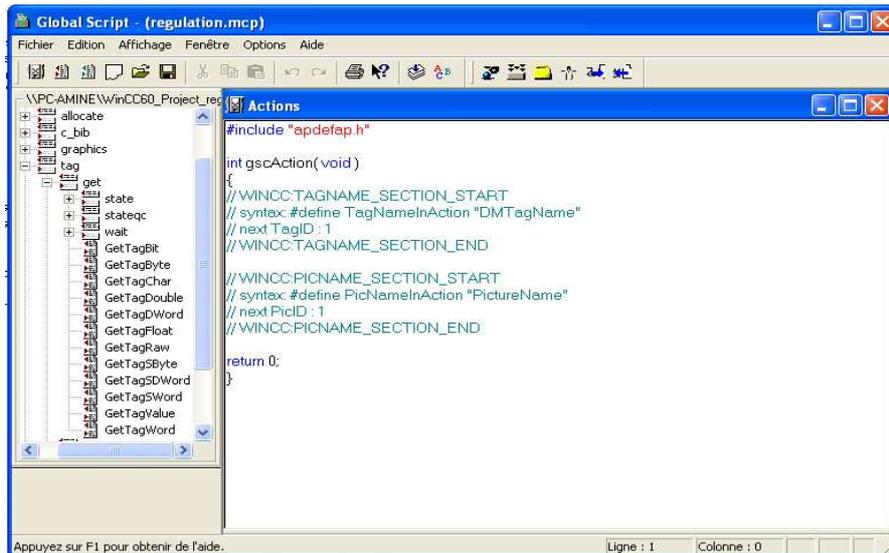


Figure III.27 : Global Script.

Deux interfaces de programmations sont disponibles :

III.5.1 VBScript :

VBScript (VBS) nous permet d'accéder en Runtime aux variables et aux objets du système graphique Runtime, mais aussi d'exécuter des actions ne dépendant pas des vues:

- Variables: On peut lire et écrire des valeurs variables pour par exemple proposer des valeurs variables pour l'automate, et ce par simple clic de souris sur un bouton.
- Objets: On peut dynamiser les propriétés de l'objet avec des actions et déclencher des actions via les événements pour objets.
- Actions ne dépendant pas de la vue: On peut déclencher des actions indépendamment des vues, de façon cyclique ou par le biais de valeurs variables, par exemple la transmission quotidienne de valeurs dans un tableau Excel.

VBS dans WinCC nous permet d'utiliser des procédures, des modules et des actions permettant de dynamiser l'environnement Runtime:

- Procédures: Une procédure correspond à une fonction en C. On inscrit dans les procédures le code qu'on veut utiliser en plusieurs endroits de la configuration. L'appel du code dans une action ou une autre procédure se fait en affichant le nom de la procédure. On peut créer dans WinCC des procédures avec ou sans valeur de retour. Les procédures n'ont pas leur propre Trigger mais sont toujours appelées par une action.

- **Modules:** Les modules nous permettent de regrouper les procédures en unités significatives. On crée par exemple des modules pour des procédures qui sont utilisées dans une vue déterminée ou qui font partie d'une rubrique particulière, par exemple les fonctions auxiliaires mathématiques ou les fonctions d'accès à la banque de données.
- **Actions:** Les actions sont toujours déclenchées par un Trigger. On configure les actions pour des propriétés d'objets graphiques, pour des événements qui interviennent pour un objet graphique ou bien globalement dans le projet. Dans les actions, on peut appeler sous forme de procédures un code utilisé plusieurs fois.

III.5.2 C-Script :

Dans le WinCC, la langue de programmation C-Script est également disponible comme interface de programmation pour dynamiser l'environnement WinCC. Il nous permet de créer des fonctions ainsi que des actions.

1. **Les fonctions :** On distingue les fonctions de projet, les fonctions standards et les fonctions internes. Un grand nombre de fonctions standards et de fonctions internes prêtes à l'emploi sont fournies avec WinCC. On peut en outre écrire nos propres fonctions de projet ou fonctions standards ou modifier celles déjà existantes.
2. **Les actions :** on distingue les actions globales et les actions locales. Dans un projet Client-Serveur, les actions globales sont exécutées sur tous les ordinateurs, les actions locales uniquement sur l'ordinateur désigné. Une action globale peut, par exemple, être utilisée pour effectuer un calcul sur tous les ordinateurs du projet. Une action locale peut, par exemple, être utilisée pour imprimer un journal sur un serveur. La création et l'édition des deux types d'action sont absolument identiques.

Pour illustrer cette partie de programmations des actions, nous avons choisi deux exemples :

a. Simulation d'un circuit RC :

Nous avons créé une vue contenant le circuit RC défini dans le chapitre précédent ainsi qu'une courbe de variation de la tension de sortie. La simulation se fait en cliquant sur le bouton qui a été dynamisé par une action en C comme suit :

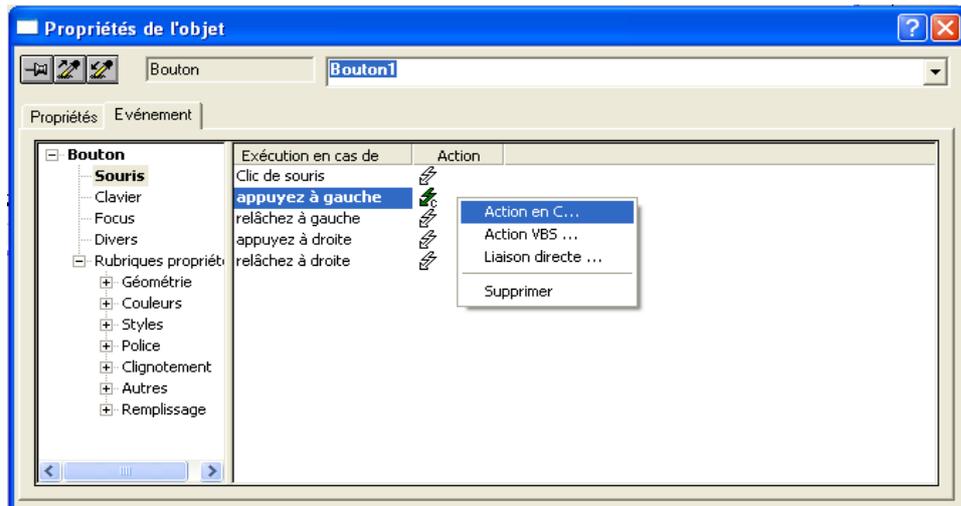


Figure III.28 : Ajout d'une action en C.

Le résultat obtenu est le suivant :

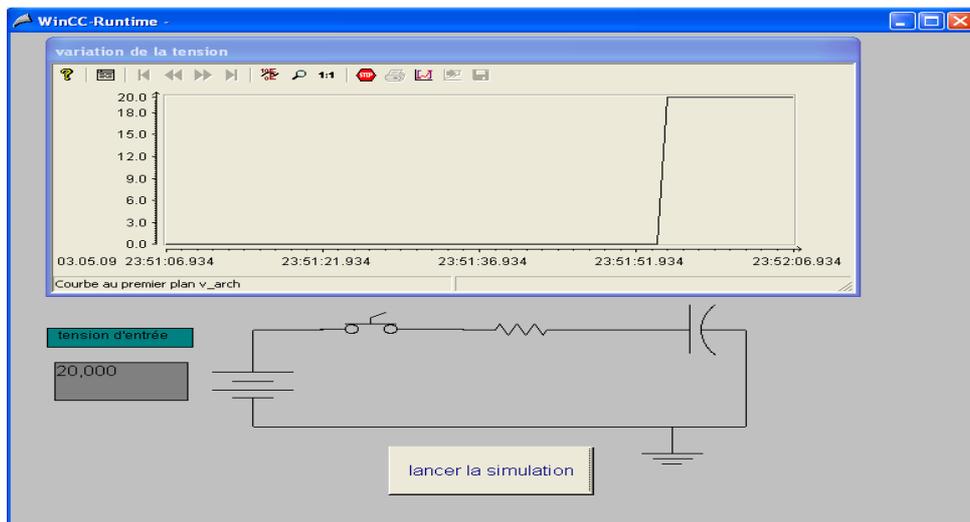


Figure III.29 : Simulation d'un circuit RC.

On voit bien que la tension de sortie converge vers la tension d'entrée (20V).

b. Régulation de la température dans une chaudière :

A travers cet exemple, nous allons voir qu'il est aussi possible de programmer un régulateur de type PI dans le but de faire la régulation de la température d'un four chauffé par un bruleur à gaz [10].

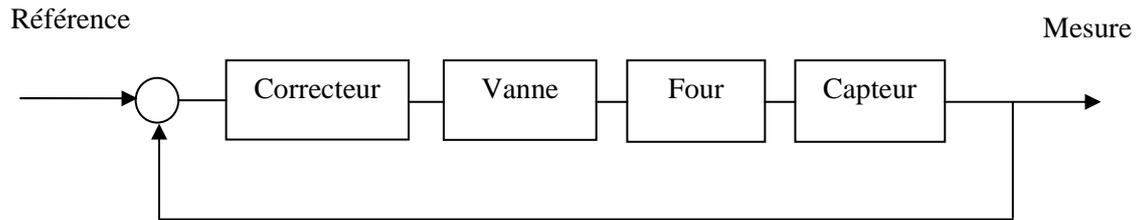


Figure III.30 : Schéma bloc du système de régulation.

La fonction de transfert du système est la suivante :

$$G(s) = \frac{1}{(1 + 300s)(1 + 45s)}$$

La synthèse du régulateur donne la fonction de transfert suivante :

$$R(s) = 1.48 \cdot 10^{-2} \frac{1 + 300s}{s}$$

Le programme de simulation est basé sur la méthode de Range-Kutta. Le programme en C utilisé est le suivant :

```

double t0,tf,h,n,ref,temp1,temp2,e,se,u,kp,ti,k1,k2,k3,k4,q1,q2,q3,q4;
int i;
//initialisation
t0=0;//instant initial
tf=10;//instant final
h=0.01;//le pas
n=(tf-t0)/h;//le nombre d'itérations
se=0;//somme des erreurs
temp1=0;//température initiale
temp2=0;
//lecture des entrées
ref=GetTagDouble("Tref");//la consigne
kp=GetTagDouble("Kp");
ti=GetTagDouble("Ti");
//simulation
for (i=0;i<n;i++)
{
e=ref-temp1;
se=se+e;
u=kp*e+1/ti*h*se;//calcul de la commande
if (u<0) u=0;
//coeff de RK4
k1=h*temp2;
q1=h*(u-temp1-345*temp2)/13500;
k2=h*(temp2+q1/2);
q2=h*(u-temp1-k1/2-345*temp2-172.5*q1)/13500;
k3=h*(temp2+q2/2);
q3=h*(u-temp1-k2/2-345*temp2-172.5*q2)/13500;
k4=h*(temp2+q3);
q4=h*(u-temp1-k3-345*temp2-345*q3)/13500;
//les températures
temp1=temp1+(k1+2*k2+2*k3+k4)/6;
temp2=temp2+(q1+2*q2+2*q3+q4)/6;
//affectation des sorties
SetTagDouble("Temperature",temp1); //Return-Type: BOOL
SetTagDouble("Commande",u);
}
    
```

Figure III.31 : Programme C de la régulation de température.

On obtient alors la courbe suivante :

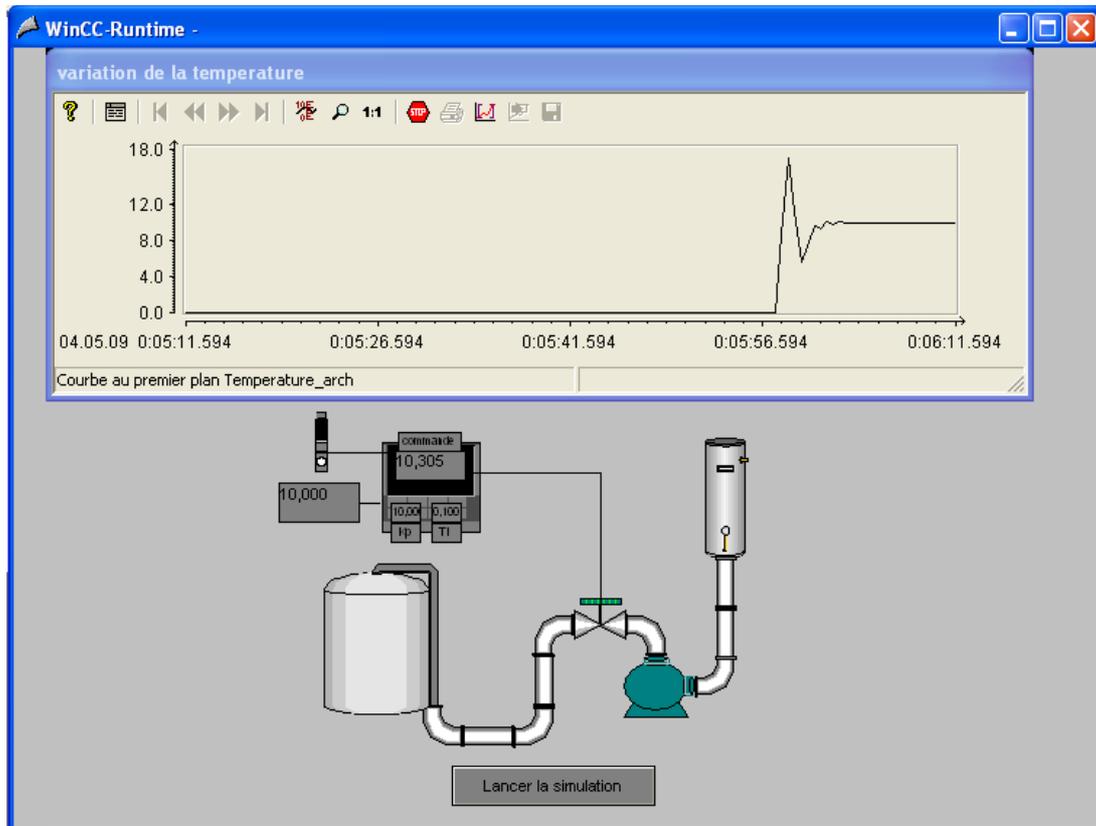


Figure III.32 : Régulation de température.

On voit bien que la température suit la consigne (10°C).

Conclusion :

Nous avons pu voir à travers ce chapitre que WinCC est un logiciel performant qui permet la supervision des processus ainsi que l'exploitation des données acquises à partir de ces derniers.

Aussi, il nous offre la possibilité d'intégrer nos propres programmes dans le but de piloter les processus.

CHAPITRE IV

Le système didactique de contrôle de processus

FESTO

Introduction :

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

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

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

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

Partie A : Description du système didactique du contrôle de processus FESTO.

IV.A.1. Présentation du système didactique du contrôle de processus FESTO [11]:



Figure IV.A.1 : Maquette didactique FESTO.

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

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

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

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

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

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

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

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

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

Chapitre IV : Le système didactique de contrôle de processus FESTO

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

Paramètres	Valeurs
Pression opérationnelle maximale	50 kPa (0,5 bar)
Alimentation	24 VDC
Support	7000 × 700 × 32mm
débit maximal de la pompe	10l/min
Volume des réservoirs	12l max
canalisations flexibles	DN15 (Ø_a 15mm)
Entrées binaires	7
Sorties binaires	5
Entrées analogiques	4
Sorties analogiques	2
Nombre de réservoirs	3
Éléments de contrôle	
Pompe (0.....24 VDC)	Tension 0...10V
Vanne proportionnelle 2/2 voies	Tension 0...10V
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

Tableau IV.A.1 : Caractéristiques techniques de l'installation.

Le diagramme synoptique de l'installation est le suivant :

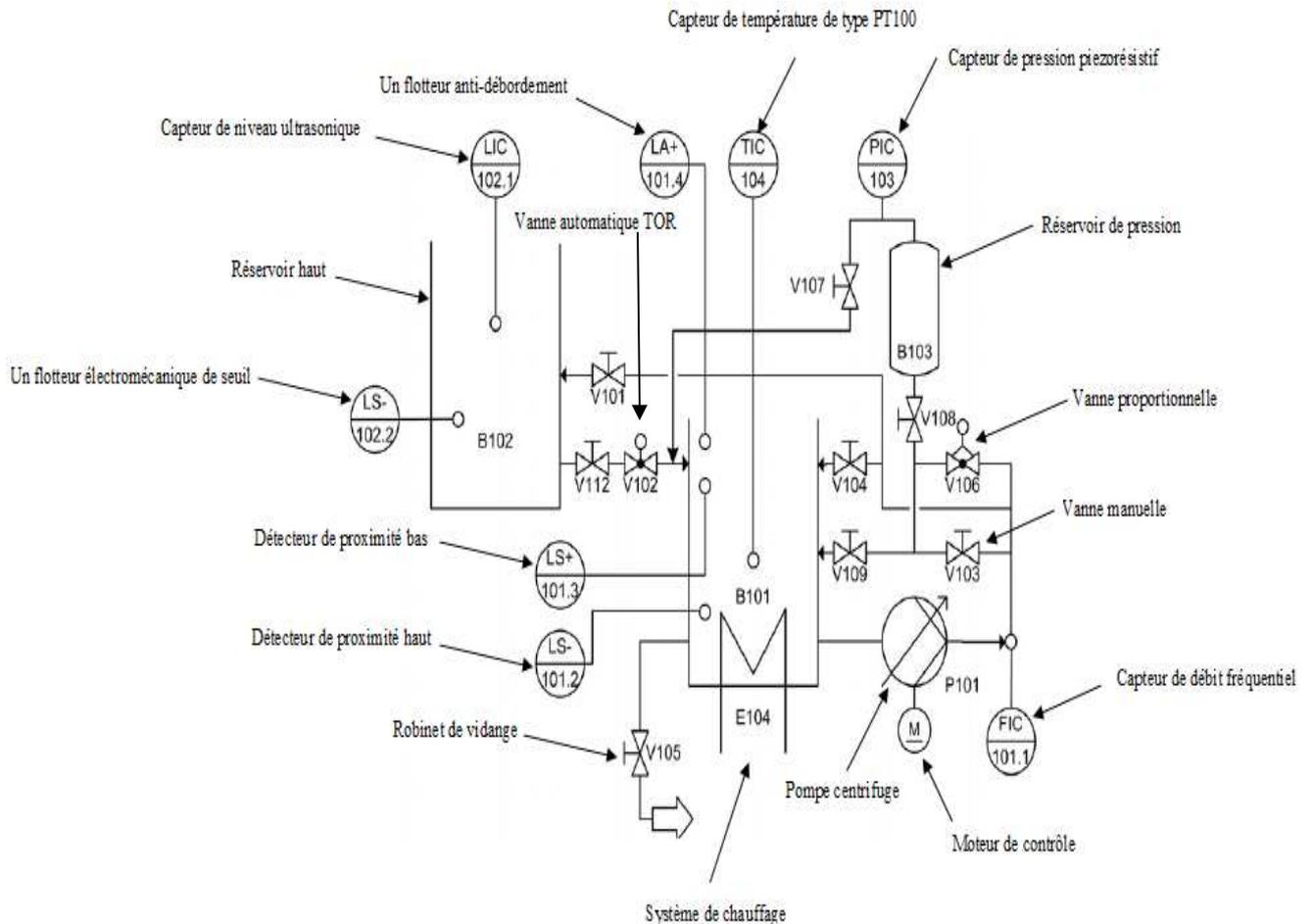


Figure IV.A.3 : Diagramme synoptique de l'installation didactique FESTO.

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

- Capteur de niveau ultrasonique (LIC102.1).
- Capteur de débit fréquentiel (FIC101.1).
- Capteur de pression piezorésistif (PIC 103).
- Jauge de pression 0...1bar.
- Capteur de température de type PT100 (TIC104).
- Deux détecteurs de proximité capacitifs dans le réservoir du bas B101 (LS-101.2, LS+101.3).
- Un flotteur électromécanique de seuil dans le réservoir du haut (LS-102.2).

- Un flotteur anti-débordement dans le réservoir du bas (LA+101.4).
- Pompe centrifuge (P101).
- Moteur de contrôle pour la pompe (M1).
- Vanne proportionnelle avec un module de contrôle électronique (V106).
- Système de chauffage avec un relai de contrôle intégré (E104).
- Processus de vanne à bille avec un entraînement rotatif pneumatique type (SYPAR) (V102).
- Un terminal E/S.
- Un terminal pour les signaux analogiques.
- Convertisseurs de signaux : courant/tension, fréquence/tension, PT100/tension.
- Automate programmable industriel.
- Panneau de contrôle.
- Canalisations.
- Réservoir de pression (B103).
- Vannes manuelles.
- Robinet de vidange (V105).
- Unité de service pneumatique.
- Support de montage.

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

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

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

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

- Connexion API/Panneau de contrôle : le Plug XMG (1/B) de l'API est connecté avec le Plug X1 (4) du panneau de contrôle à l'aide du câble SysLink (5).

- Connexion API/Unité d'alimentation : Le Plug d'alimentation de l'API est connecté à l'unité d'alimentation avec un câble de sécurité bleu et rouge (4mm).
- Connexion API/PC : le PC est connecté à L'API avec un câble de programmation MPI.

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

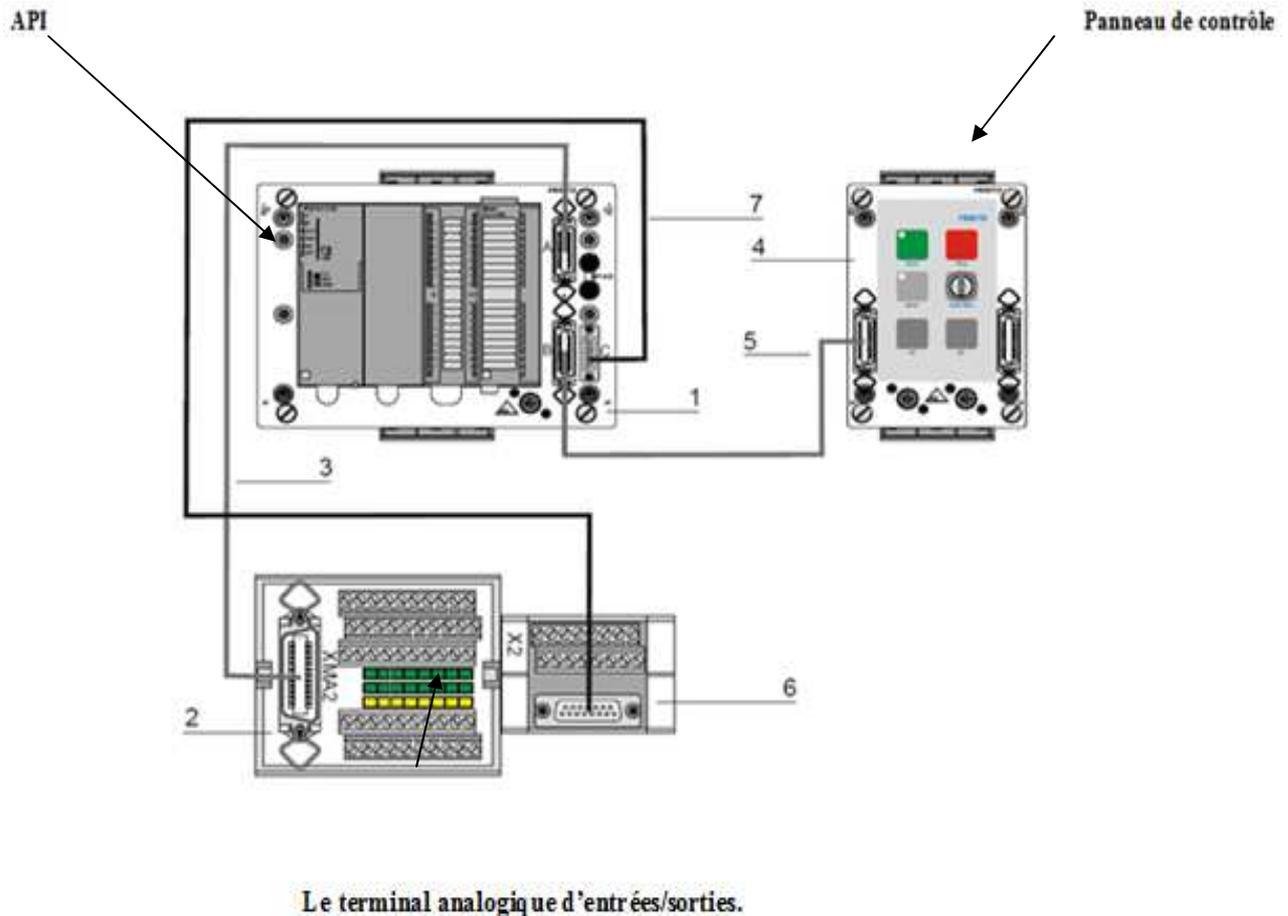


Figure IV.A.4 : Schéma de connexion API/Panneau de contrôle/Terminal analogique.

IV.A.2. Design et fonction des différents systèmes constituant l'installation :

IV.A.2.1. La pompe :

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



Figure IV.A.5 : Pompe centrifuge.

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

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

IV.A.2.2. Le Processus de la vanne à bille :

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

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



Figure IV.A.6 : Processus de la vanne à bille.

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

IV.A.2.3. 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)

1. 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 IV.A.7 : Détecteurs de proximité capacitifs.

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

2. 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 IV.A.8 : 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.

3. 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 IV.A.9 : 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.

IV.A.2.4. 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.

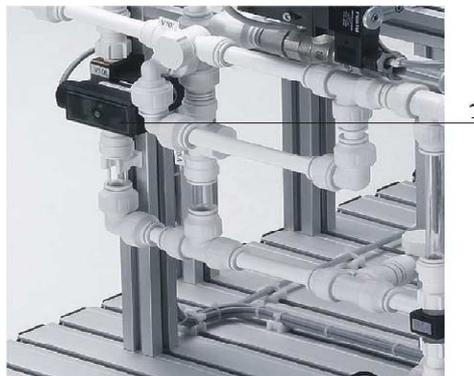


Figure IV.A.10 : Vanne proportionnelle.

La vanne proportionnelle est activée par une sortie binaire (O4 dans le XMA1) .Un signal analogique du canal (UA2 de X2) actionne le signal d'entrée de la vanne proportionnelle avec un signal analogique 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.

IV.A.2.5. Le capteur de niveau ultrasonique :

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



Figure IV.A.11 : Capteur de niveau ultrasonique.

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

IV.A.2.6. 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 IV.A.12 : capteur de débit.

IV.A.2.7. 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).

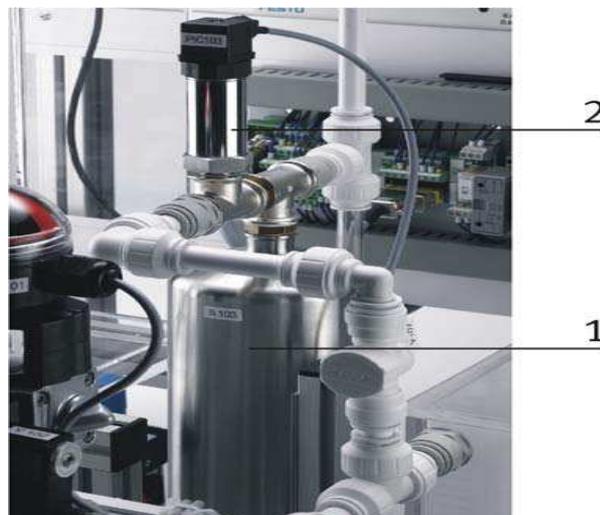


Figure IV.A.13 : Capteur de pression.

IV.A.2.8. 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 IV.A.14 : 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 IV.A.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 IV.A.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 IV.A.4 : Tableau d'adressage des entrées/sorties analogiques.

Partie B : Régulation des systèmes continus

IV.B.1. les systèmes de régulation en boucle fermée :

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

Composants	Système de contrôle de 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	Élément de contrôle	Élément de contrôle	Mode binaire on	Élément de contrôle	Élément de contrôle	Mode binaire on
Vanne proportionnelle V106	Off	Off	Élément de contrôle	Off	Élément de contrôle	Off
Élément chauffant E104	Off	Off	Off	Off	Off	Élément de contrôle
Vanne manuelle V101	Ouverte	Fermée	Fermée	Fermée	Fermée	Fermée
Vanne automatique tout ou rien V104	Ouverte/ Fermée	Fermée	Fermée	Fermée	Fermée	Fermée
Vanne manuelle V103	Fermée	Fermée	Fermée	Ouverte	Fermée	Ouverte
Vanne manuelle V104	Fermée	Ouverte	Fermée	Fermée	Fermée	Fermée
Vanne	Fermée	Fermée	Fermée	Fermée	Fermée	Fermée

manuelle de drainage V105						
Vanne manuelle V107	Fermée	Fermée	Fermée	Fermée	Fermée	Fermée
Vanne manuelle V108	Fermée	Fermée	Fermée	Ouverte	Ouverte	Fermée
Vanne manuelle V109	Fermée	Fermée	Ouverte	Ouverte/ Fermée	Ouverte/ Fermée	Ouverte
Vanne manuelle V112	Ouverte	Fermée	Fermée	Fermée	Fermée	Fermée

Figure IV.B.1 : Tableau des configurations relatives aux différentes boucles de régulation.

IV.B.1.1. Système de régulation de débit :

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

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

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

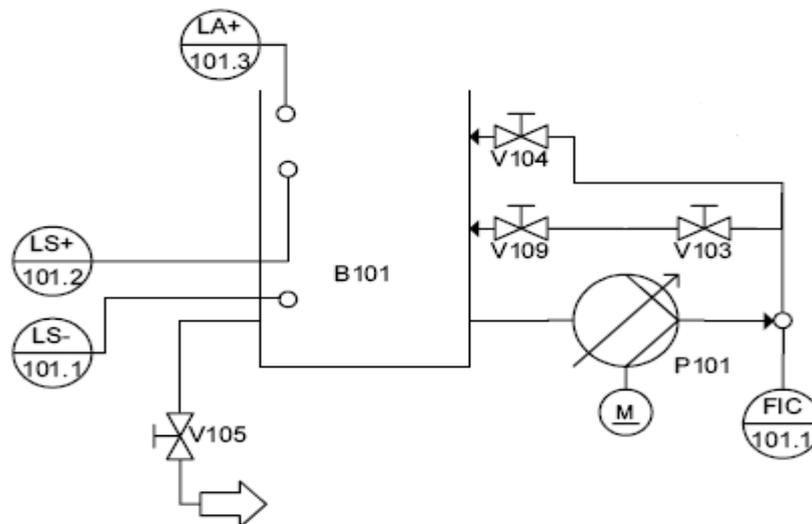


Figure IV.B.6 : Circuit relatif à la régulation de débit par le biais de la pompe.

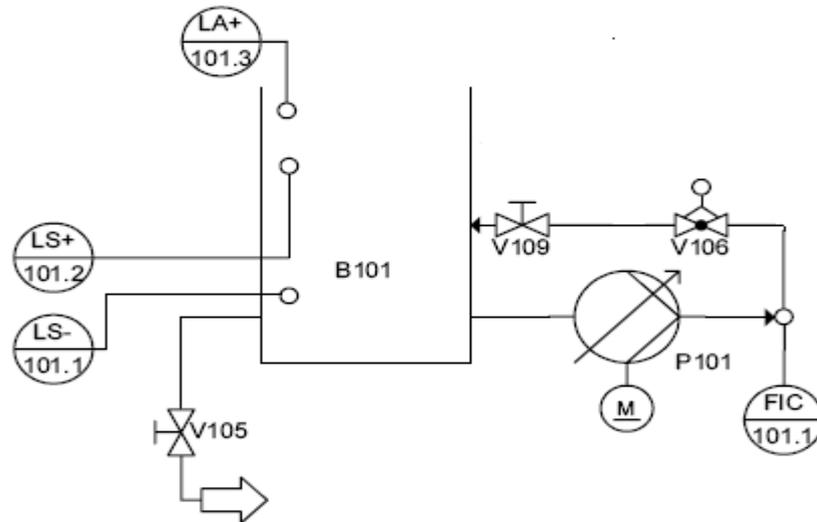


Figure IV.B.7 : Circuit relatif à la régulation de débit par le biais de la vanne proportionnelle.

1. Régulation de débit avec la pompe comme contrôleur :

a. Description du système pompe-débit :

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

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

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

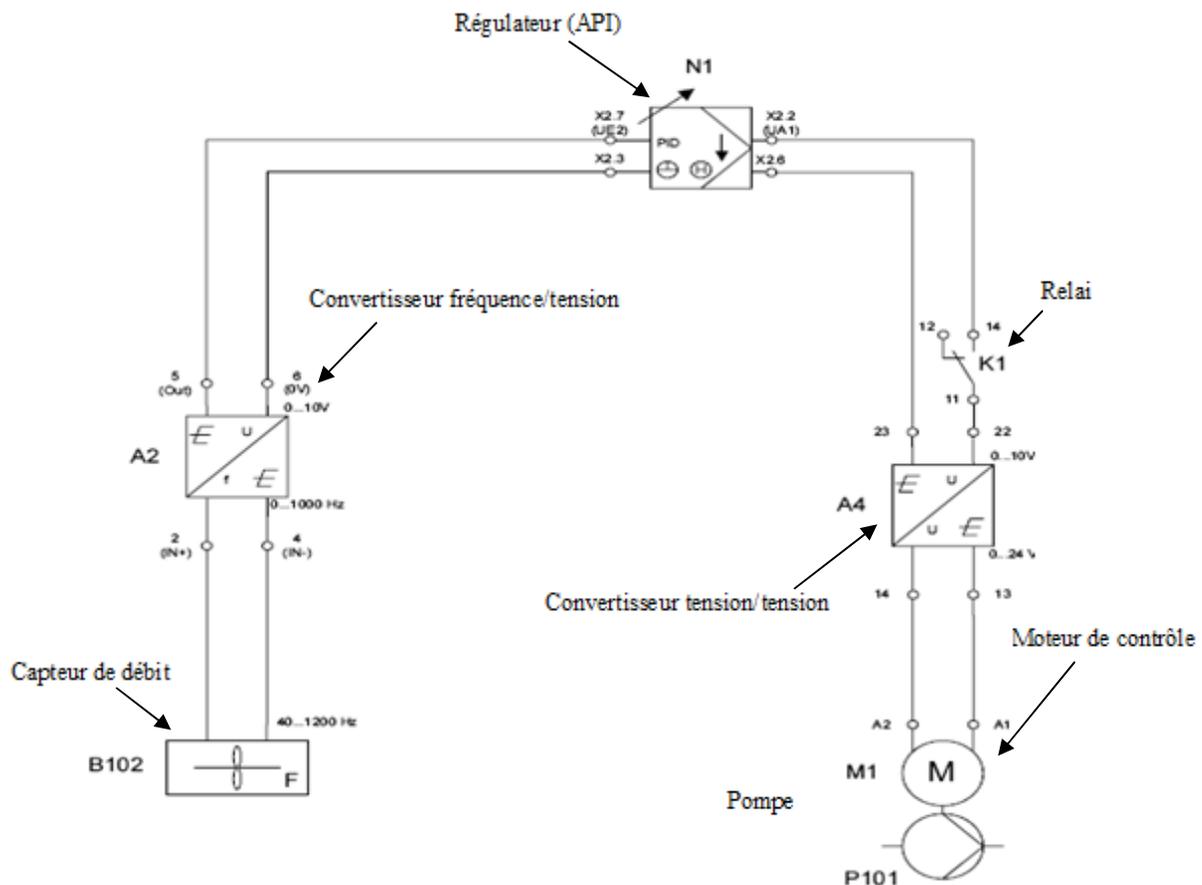


Figure IV.B.8 : Diagramme électrique de principe de la régulation de débit avec la pompe comme contrôleur.

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

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

b. Identification du système pompe-débit :

Pour l'identification de ce système, nous nous sommes basés sur l'étude de la réponse indicielle en boucle ouverte. On envoie un échelon à la pompe d'amplitude 90% et on relève la réponse qui est représentée dans la figure ci-dessous à l'aide de l'interface de paramétrage PID décrite dans le chapitre II:

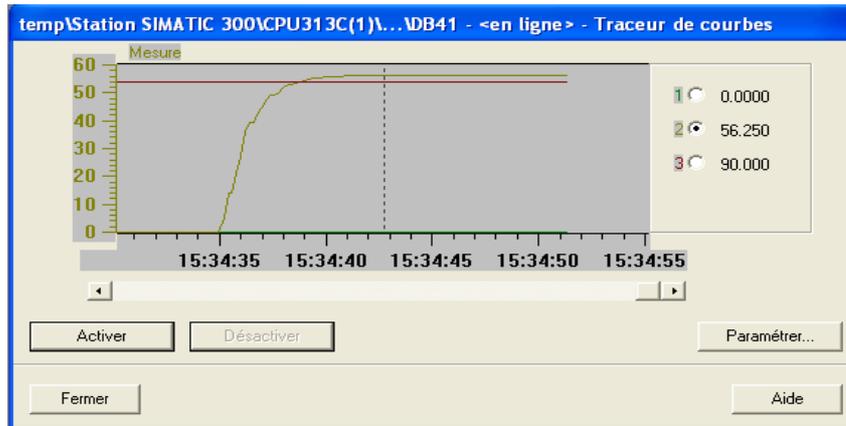


Figure IV.B.9 : Réponse indicielle du débit (pompe) en boucle ouverte.

On assimile la réponse de ce système à celui d'un ordre 1 (Voir Annexe-B) dont la forme de la fonction de transfert est :

$$\frac{K}{(1 + \tau s)} \text{ où: } \begin{cases} K = \frac{56.25}{90} = 0.625. \\ \tau = 1.75 \text{ sec.} \end{cases}$$

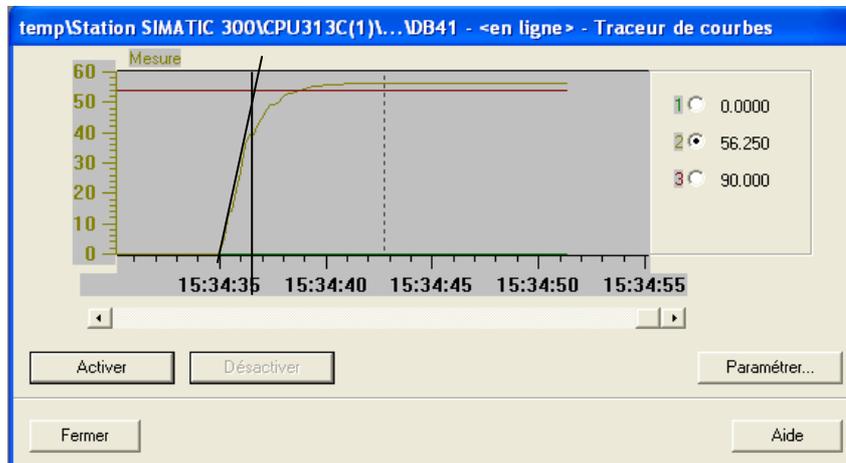


Figure IV.B.10 : Calcul de la constante de temps graphiquement.

c. Synthèse du régulateur PI :

On utilise la fonction de transfert calculée précédemment pour la synthèse du régulateur. La réponse indicielle ayant une erreur statique, le régulateur approprié est le régulateur PI dont la fonction de transfert est du type : $K_p \left(1 + \frac{1}{T_{is}} \right)$ (Voir Annexe-C).

Pour le calcul des paramètres de ce dernier on utilise la méthode du lieu d'Evans en imposant à la réponse en boucle fermée un seul dépassement ($\xi=0.7$).

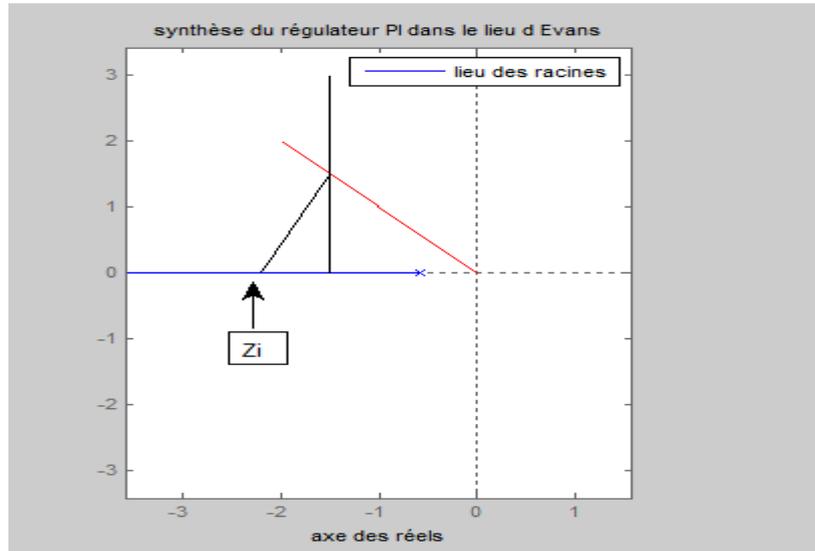


Figure IV.B.11 : Méthode du lieu d'Evans pour le calcul des paramètres du régulateur PI.

A partir de la condition des angles, on détermine la position du zéro du régulateur dans le lieu d'Evans ($\alpha_i = 68^\circ$) et on relève la valeur de ce dernier : $Z_i = 2,2$

A partir de la condition des modules on calcule le gain total du système avec régulateur :

$$K' = 2.1.$$

Finalement, on aura les paramètres suivants :

$$\begin{cases} K_p = \frac{K'}{K} = 3.36. \\ T_i = \frac{1}{Z_i} = 454 \text{ ms}. \end{cases} \quad \text{D'où : } R(s) = 3.36 \left(1 + \frac{1}{0.454s} \right).$$

On simule le système en boucle fermée et on obtient le résultat suivant :

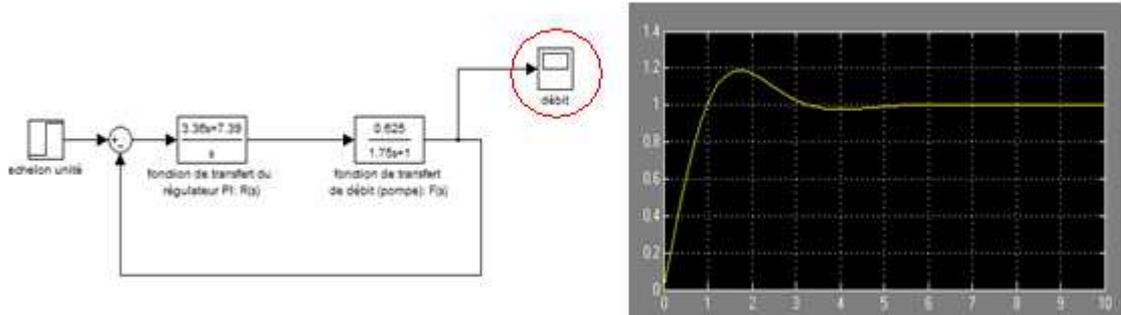


Figure IV.B.12 : Simulation de la boucle de régulation de débit (pompe).

On voit bien que le système est parfaitement régulé et il suit le cahier des charges imposé.

En appliquant le régulateur calculé théoriquement on obtient une réponse similaire à celle obtenue avec la simulation. La figure suivante représente la courbe réelle :

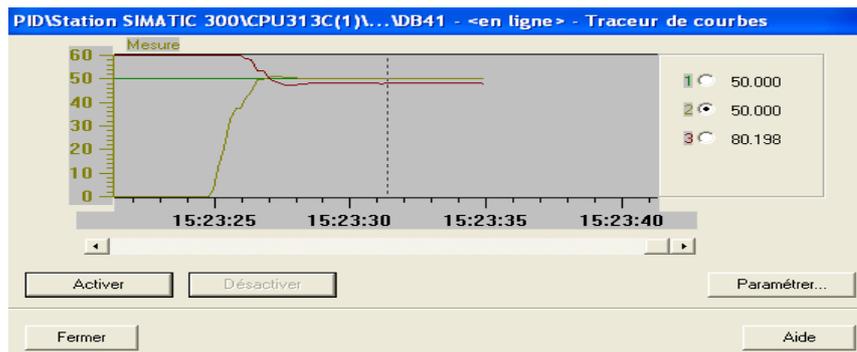


Figure IV.B.13 : Réponse indicielle du débit réelle (pompe) en boucle fermée.

2. Régulation de débit avec la vanne proportionnelle comme contrôleur :

a. Description :

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

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

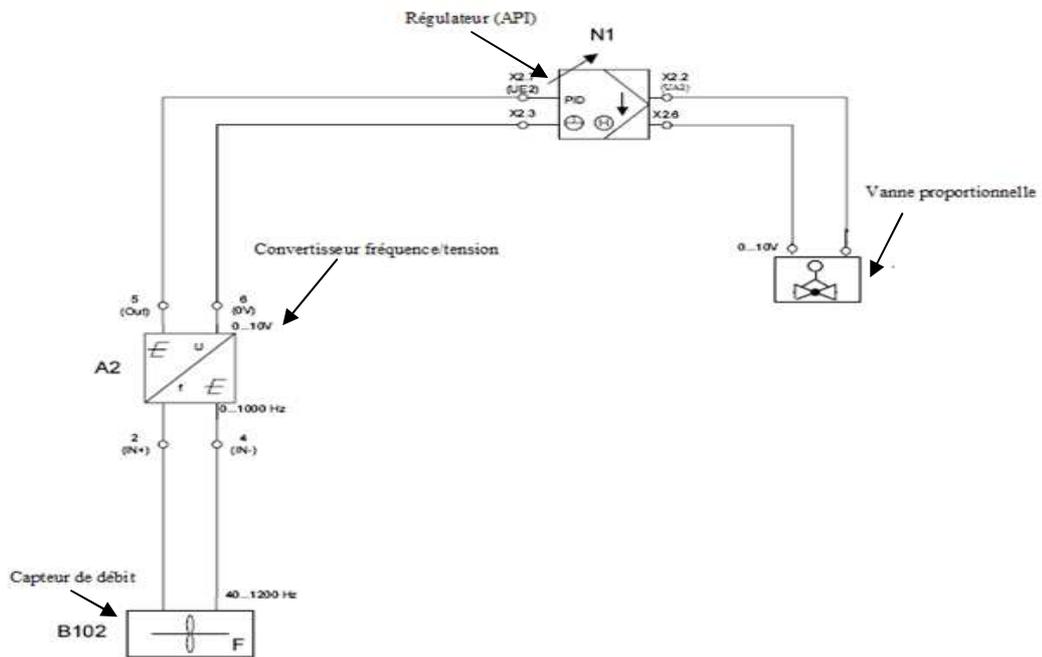


Figure IV.B.14 : Diagramme électrique de principe de la régulation de débit avec la vanne proportionnelle comme contrôleur.

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

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

b. Identification du système vanne-débit :

Tout comme pour la partie précédente, nous nous sommes basés sur l'étude de la réponse indicielle en boucle ouverte. On envoie un échelon à la vanne analogique d'amplitude 90% et on relève la réponse qui est représentée dans la figure ci-dessous à l'aide de l'interface de paramétrage PID :

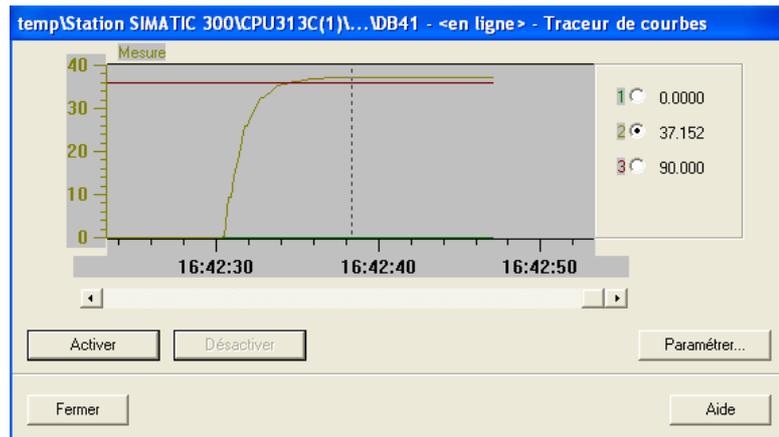


Figure IV.B.15 : Réponse indicielle du débit (vanne analogique).

On assimile la réponse de ce système à celui d'un ordre 1 dont la forme de la fonction de transfert est :

$$\frac{K}{(1 + \tau s)} \quad \text{où:} \quad \begin{cases} K = \frac{37.152}{90} = 0.41. \\ \tau = 1.75 \text{ sec.} \end{cases}$$

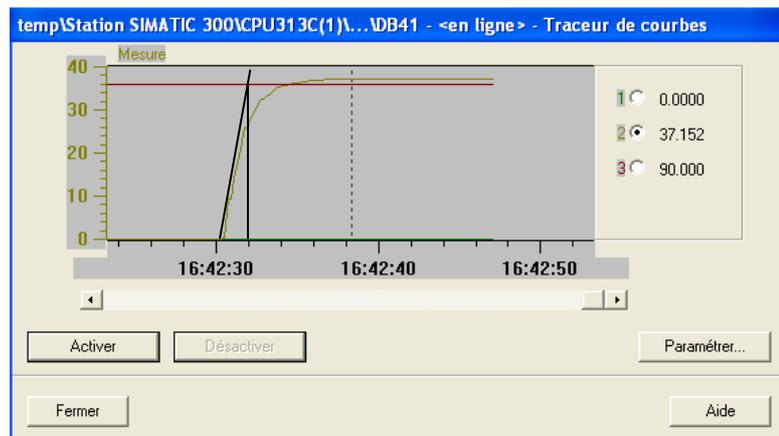


Figure IV.B.16 : Calcul de la constante de temps graphiquement.

c. Synthèse du régulateur PI :

On utilise la fonction de transfert calculée précédemment pour la synthèse du régulateur. La réponse indicielle ayant une erreur statique, le régulateur approprié est le régulateur PI dont la fonction de transfert est du type : $K_p \left(1 + \frac{1}{T_{is}} \right)$.

Pour le calcul des paramètres de ce dernier on utilise la méthode du lieu d'Evans en imposant à la réponse en boucle fermée un seul dépassement ($\xi=0.7$).

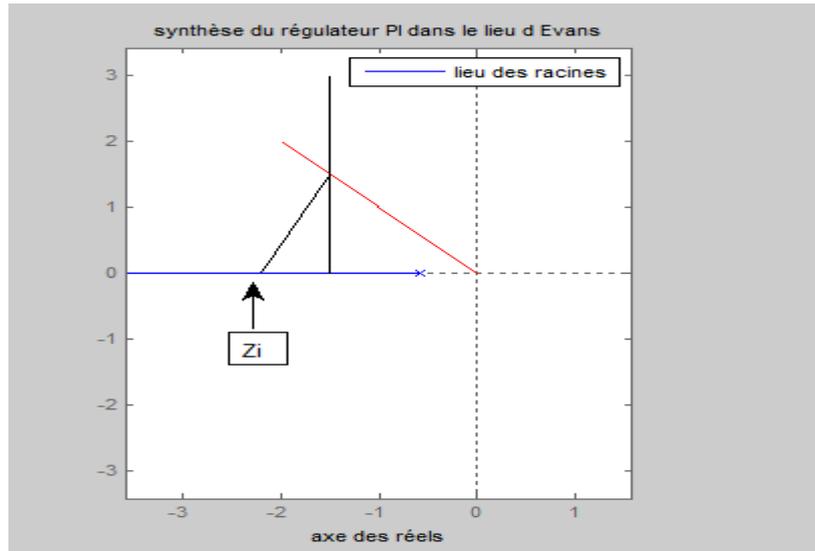


Figure IV.B.17 : Méthode du lieu d'Evans pour le calcul des paramètres du régulateur PI.

A partir de la condition des angles, on détermine la position du zéro du régulateur dans le lieu d'Evans ($\alpha_i = 68^\circ$) et on relève la valeur de ce dernier : $Z_i = 2,2$

A partir de la condition des modules on calcul le gain total du système avec régulateur : $K' = 2.1$.

Finalement, on aura les paramètres suivants :

$$\begin{cases} K_p = \frac{K'}{K} = 5.12. \\ T_i = \frac{1}{Z_i} = 454 \text{ ms}. \end{cases} \quad \text{D'où : } R(s) = 3.36 \left(1 + \frac{1}{0.454s} \right).$$

La simulation du système en boucle fermée donne les résultats suivants :

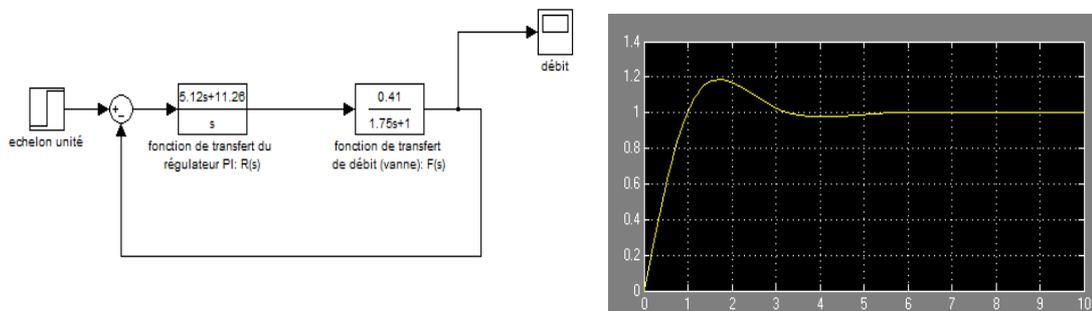


Figure IV.B.18 : Réponse indicielle théorique en boucle fermée (débit-vanne).

On voit bien que le système est parfaitement régulé et il suit le cahier des charges imposé.

En appliquant le régulateur calculé théoriquement on obtient une réponse similaire à celle obtenue avec la simulation. La figure suivante représente la courbe réelle :

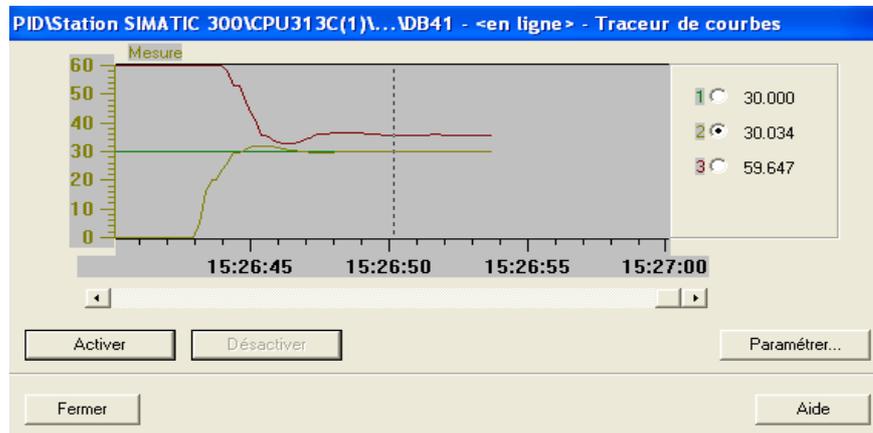


Figure IV.B.19 : Réponse indicielle réelle en boucle fermée (débit-vanne).

IV.B.1.2. Système de régulation de niveau :

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

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

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

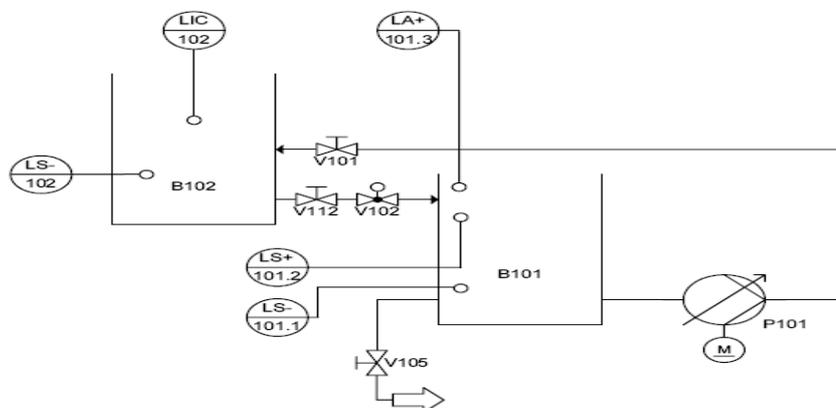


Figure IV.B.1 : Circuit relatif à la régulation de niveau.

1. Régulation de niveau analogique :

a. Description :

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

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

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

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

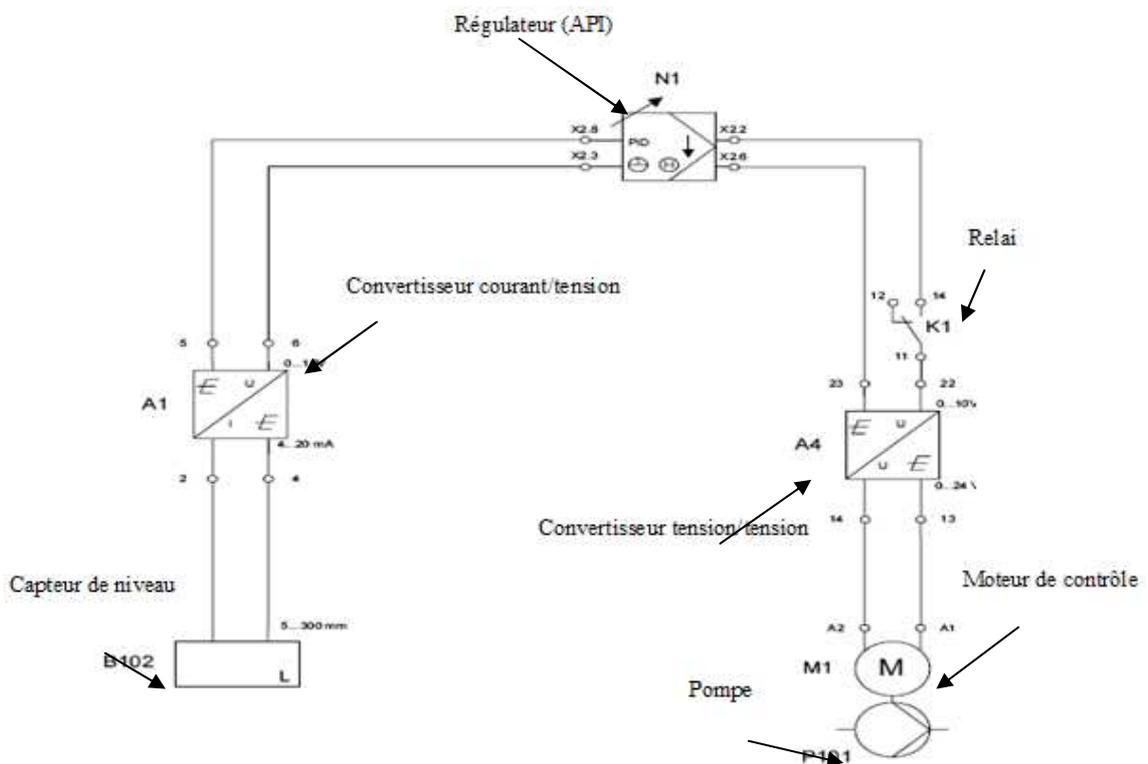


Figure IV.B.2 : Diagramme électrique de principe de la régulation de niveau analogique.

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

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

b. Identification du système niveau-pompe:

Pour la détermination de la fonction de transfert du niveau, nous nous sommes basés sur des lois physiques pour déterminer la relation entre le débit et le niveau.

Une fois cette relation établie, et connaissant la fonction de transfert du débit, on détermine la fonction de transfert du niveau.

Nous avons : $Q_p = \frac{dV}{dt}$ où : Q_p désigne le débit sortant de la pompe et V le volume sortant de la pompe.

En appliquant le principe de conservation du débit on aura : $Q_p = Q_{res}$ où : Q_{res} est le débit de remplissage du réservoir. Or :

$$Q_{res} = \frac{dV}{dt} \text{ avec : } V = N.S$$

où: N Est le niveau de l'eau dans le réservoir et S la section de ce dernier.

On aura alors :

$$Q_p = S. \frac{dN}{dt} \text{ , la section } S \text{ est constante et est égale à : } l.L$$

En passant dans le domaine de Laplace on trouve :

$$Q_p(s) = l.L.s.N(s)$$

Finalement :

$$N(s) = \frac{1}{l.L} \cdot \frac{Q_p(s)}{s}$$

Comme la fonction de transfert de débit est exprimée en gain relatif, on doit la formuler en débit réel (m^3/s). On aura alors :

$$N(s) = \frac{k}{l.L} \cdot \frac{Q_p(s)}{s} \quad \text{avec:} \quad Q_p = \frac{0.625}{(1 + 1.75s)} \quad \text{et} \quad \begin{cases} k = \frac{7,5 \cdot 10^{-3}}{60} m^3/s \\ l = 0.175m, L = 0.19 m \end{cases}$$

Finalement on obtient :

$$N(s) = \frac{2.35 \cdot 10^{-3}}{s(1 + 1.75s)} m$$

On normalise la fonction par le facteur de mise à l'échelle qui est égale à 0.3 et on obtient la fonction de transfert du niveau normalisé :

$$N(s) = \frac{7.83 \cdot 10^{-3}}{s(1 + 1.75s)}$$

c. Synthèse du régulateur P :

Le système contenant déjà une intégration, on utilise un régulateur P. Pour le calcul du gain K_p , on calcule la fonction de transfert en boucle fermée et on la met sous la forme :

$$\frac{w_n^2}{s^2 + 2\xi w_n + w_n^2} \quad \text{avec:} \quad \begin{cases} w_n^2 = K_p \cdot 4.46 \cdot 10^{-4} \\ 2\xi w_n = 0.57 \end{cases}$$

En imposant : $\xi = 0.7$ on trouve finalement :

$$K_p = \frac{1}{4 \cdot 0.0078 \cdot 1.75 \xi^2} = 37.3$$

La simulation du système en boucle fermée donne les résultats suivants :

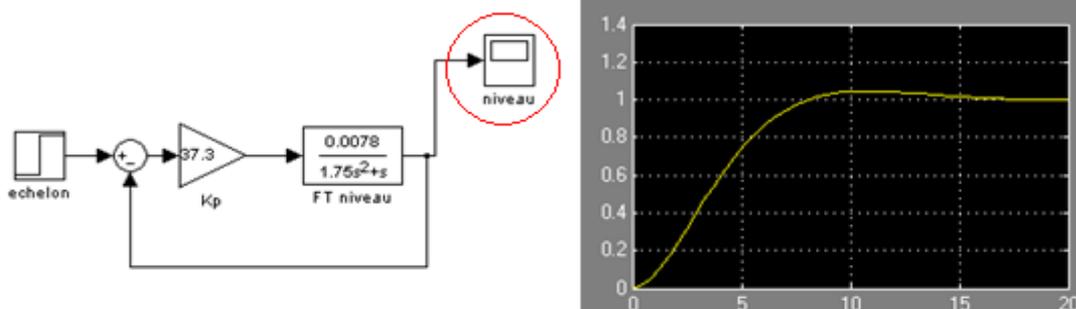


Figure IV.B.3 : Réponse indicielle théorique en boucle fermée de niveau.

On applique ces résultats sur le système réel et on obtient la courbe suivante :

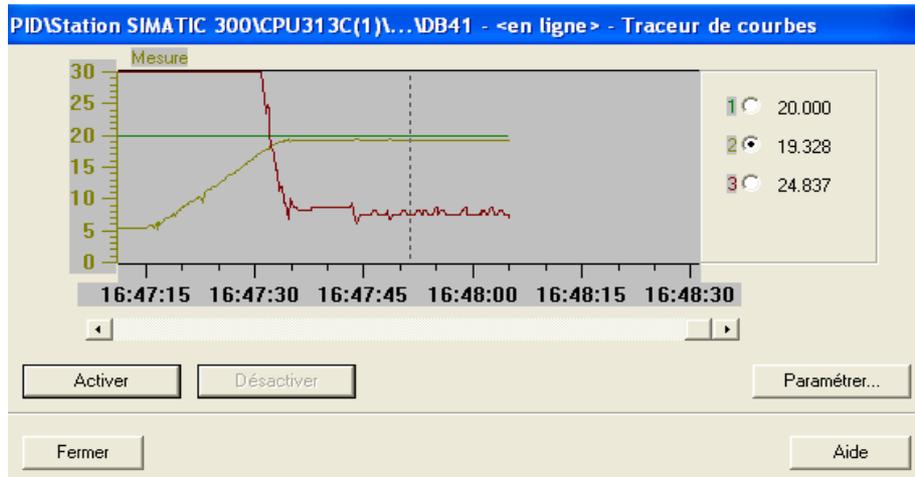


Figure IV.B.4 : Réponse indicielle réelle en boucle fermée de niveau.

On voit bien que le système est stable mais avec une petite erreur statique.

Ceci est dû au fait que la pompe ne soit pas équipée d'un clapet anti retour, et ainsi il y a un débit de vidange. On y remédie en ajoutant une action intégrale pour compenser cette erreur. On obtient alors la réponse suivante :

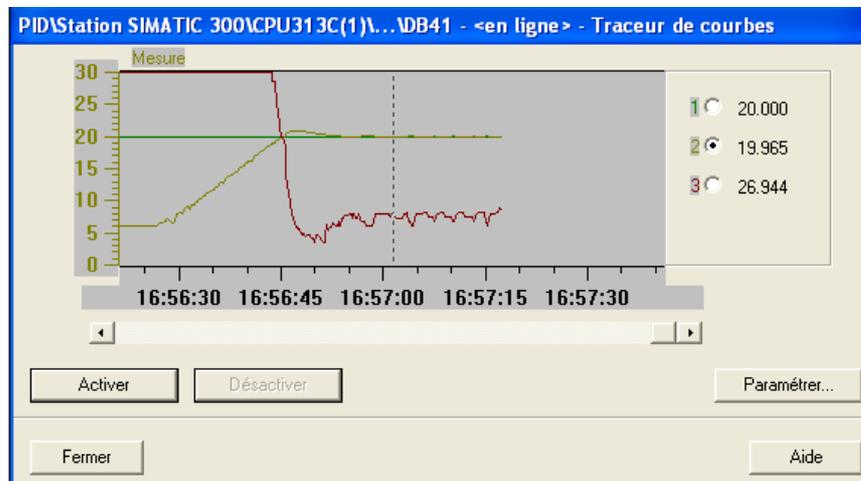


Figure IV.B.5 : Réponse indicielle réelle en boucle fermée de niveau avec action I.

On voit bien que le système est parfaitement régulé.

IV.B.1.3. Système de régulation de pression :

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

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

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

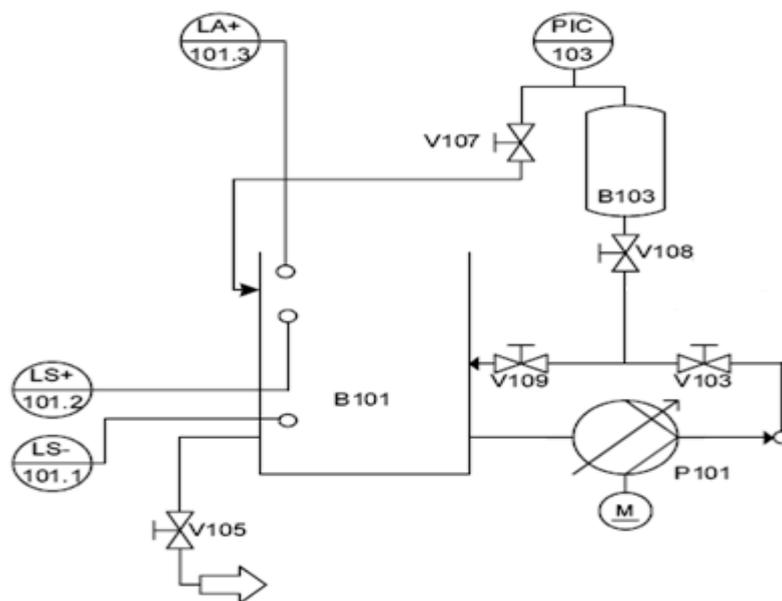


Figure IV.B.20 : Circuit relatif à la régulation de pression avec la pompe comme actionneur.

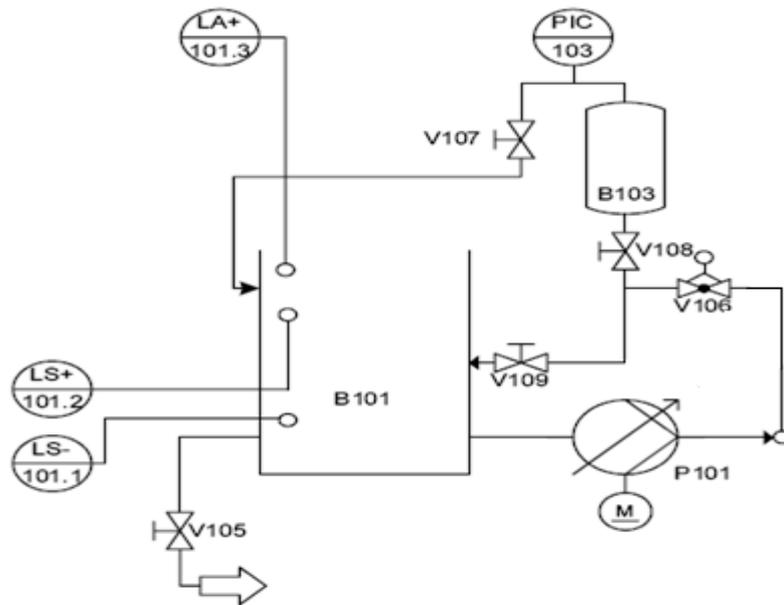


Figure IV.B.21 : Circuit relatif à la régulation de pression avec la pompe comme actionneur.

1. Régulation de pression avec la pompe comme actionneur :

a. Description :

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

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

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

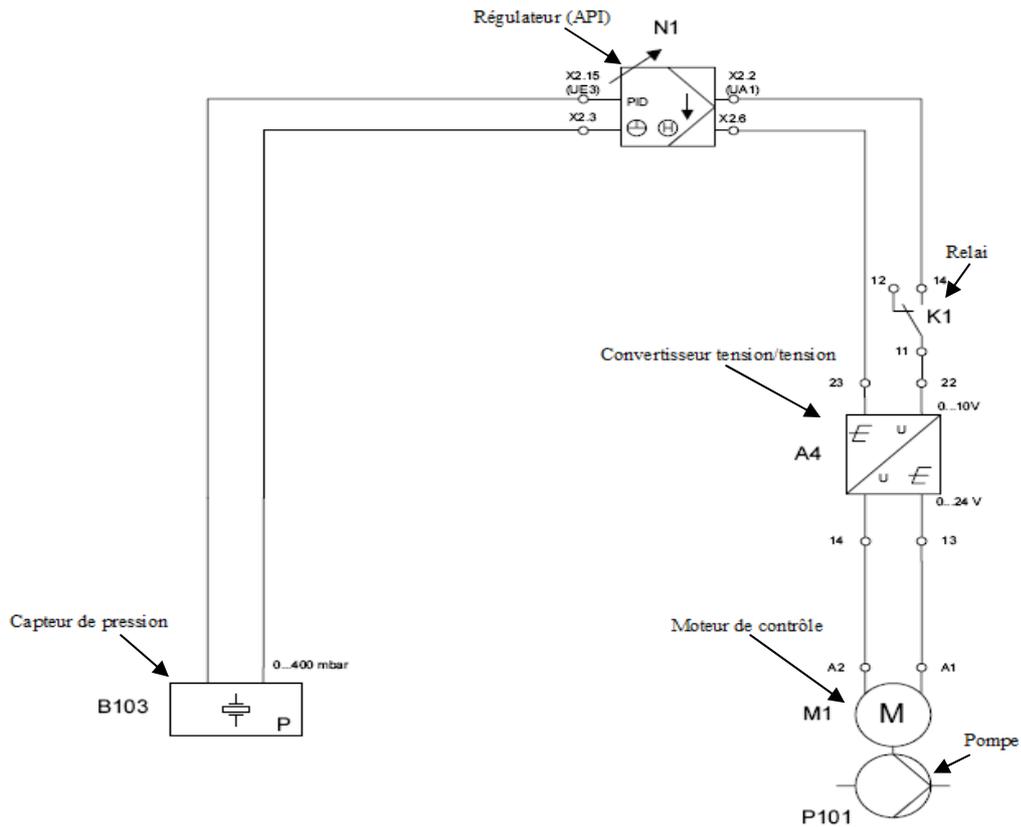


Figure IV.22 : Diagramme électrique de principe de la régulation de pression avec la pompe comme contrôleur.

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

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

b. Régulation PID par la méthode de Ziegler-Nichols :

La méthode que nous avons utilisée pour la synthèse du régulateur PID est celle de Ziegler-Nichols et qui consiste à régler les paramètres de ce dernier à partir d'un essai en limite de pompage (d'une période p) et ceci en faisant varier le gain du régulateur jusqu'à une certaine valeur (K_0) (Voir Annexe-D).

On obtient de cet essai les données suivantes : $K_0 = 4$ et $T = 1.5$ sec

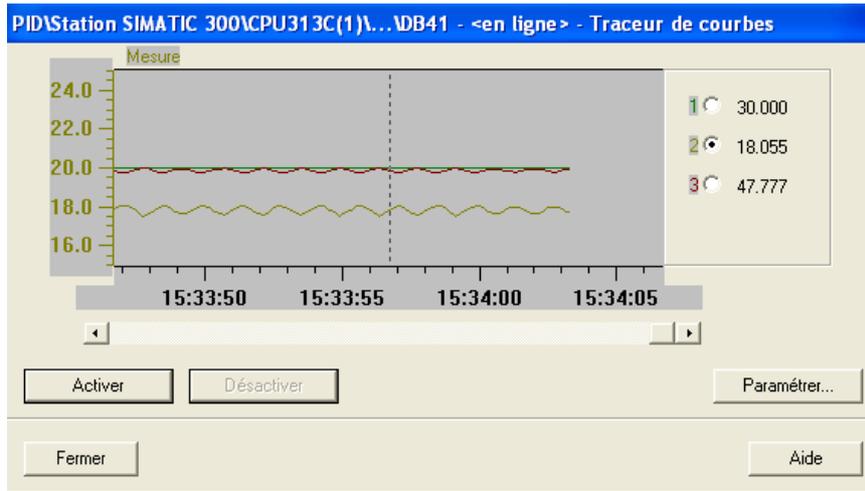


Figure IV.B.23 : Réponse en régime de pompage de la pression (pompe).

Le gain statique est calculé à partir de la réponse indicielle en boucle ouverte

$$K = \frac{57.87}{90} = 0.64$$

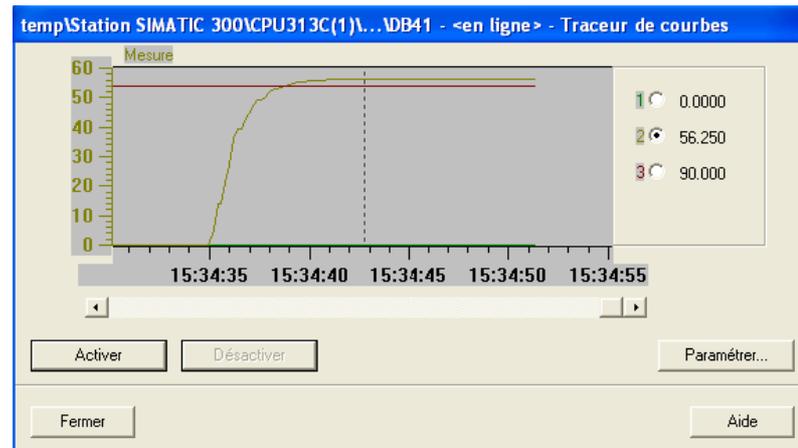


Figure IV.B.24 : Réponse indicielle en boucle ouverte de la pression (pompe).

On aura alors : $K_{cr} = K \cdot K_0 = 0.64 \cdot 4 = 2.56$

Le calcul des paramètres du régulateur PID est basé sur des relations empiriques :

$$\begin{cases} K_p = 0.6 & K_{cr} = 1.53 \\ T_i = \frac{T}{2} = 0.75 \text{ sec} \\ T_d = \frac{T}{8} = 0.18 \text{ sec} \end{cases}$$

Finalement, la fonction de transfert du régulateur est la suivante :

$$R(s) = 1.53 \left(1 + \frac{1}{T_i s} + T_d s \right)$$

En appliquant le régulateur ainsi calculé au système réel on obtient la réponse suivante :

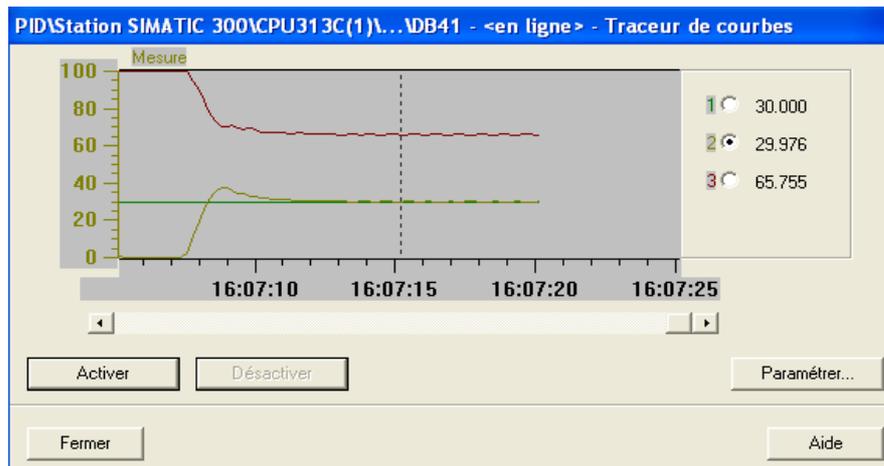


Figure IV.B.25 : Réponse indicielle en boucle fermée de la pression (pompe)

2. Régulation de pression avec la vanne proportionnelle comme actionneur :

a. Description :

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

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

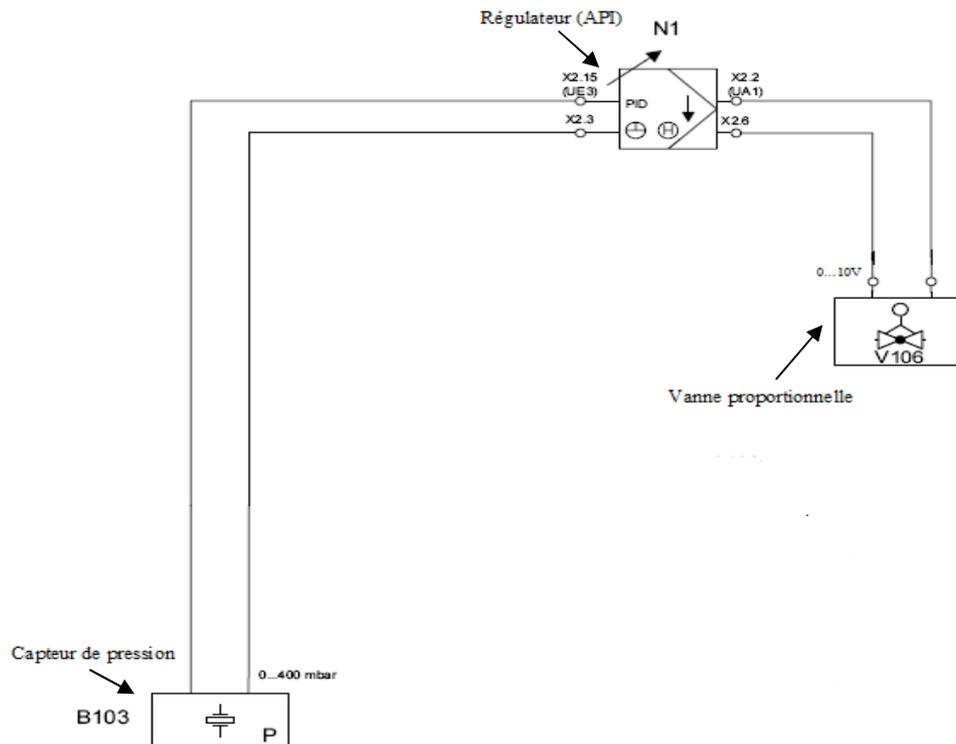


Figure IV.B.26 : Diagramme électrique de principe de la régulation de pression avec la vanne proportionnelle comme contrôleur.

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

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

b. Identification du système vanne-pression :

En effectuant l'essai de pompage dans ce cas, nous avons constaté que le système reste stable quel que soit la valeur du gain. En prenant en considération ce résultat ainsi que la réponse indicielle en boucle ouverte du système, on l'identifie alors par un ordre 1.

En effectuant les tests en boucle ouverte, on remarque que le gain statique varie suivant l'entrée suivant la relation suivante : $K = \frac{0.66}{E}$

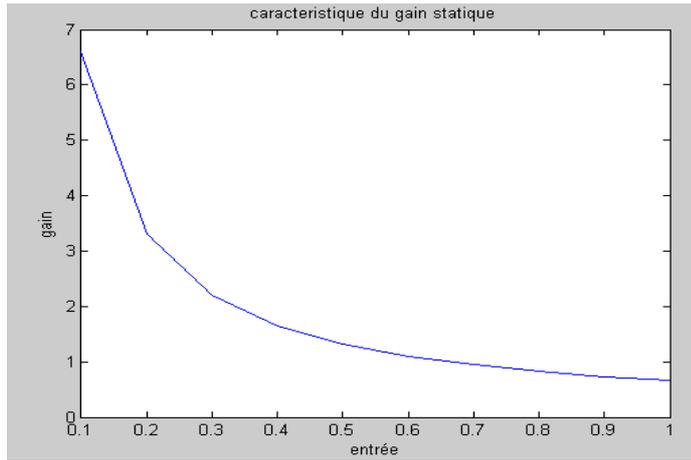


Figure IV.B.27 : Caractéristique du gain statique.

Nous avons donc choisit d'identifier notre système autour d'un point de fonctionnement (entrée = 40%).

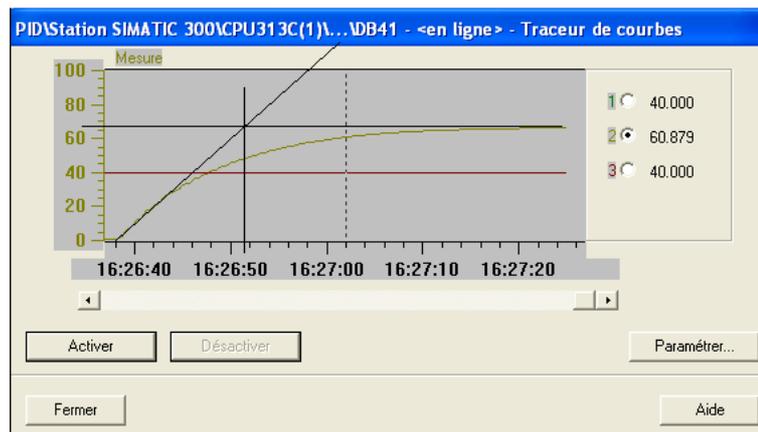


Figure IV.B.28 : Réponse indicielle en boucle ouverte de la pression (vanne analogique).

On utilisant la méthode d'identification graphique d'un système du premier ordre on obtient la fonction de transfert suivante :

$$F(s) = \frac{1.65}{(1 + 13.5s)}$$

c. Synthèse du régulateur PI :

Pour la synthèse de notre régulateur PI, nous avons utilisé la méthode du lieu des racines.

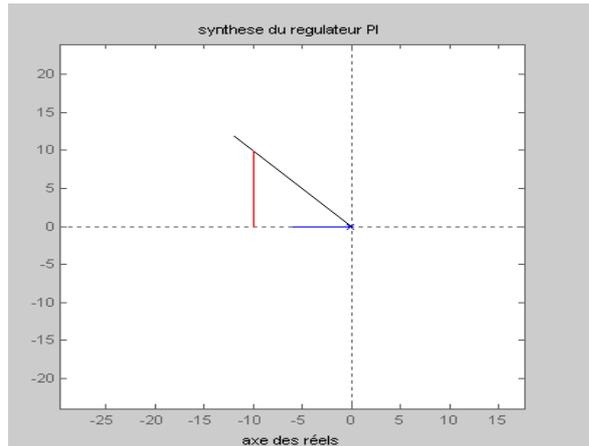


Figure IV.B.29 : Lieu d'Evans pour le calcul du régulateur PI.

On obtient alors le régulateur suivant :

$$R(s) = 12.12\left(1 + \frac{1}{0.1s}\right)$$

En utilisant ces paramètres on obtint la réponse suivante :

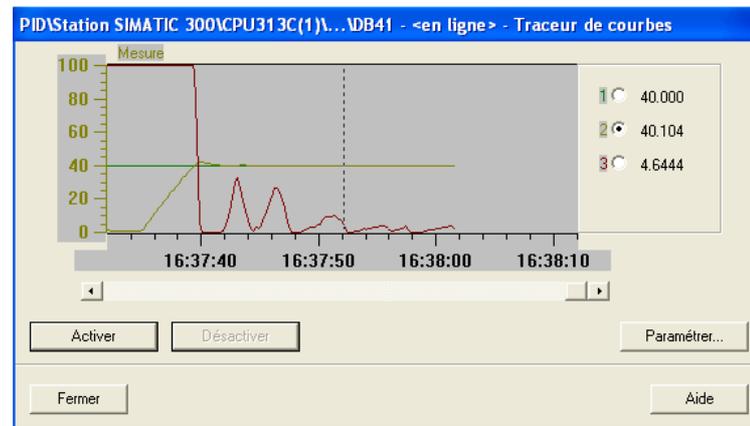


Figure IV.B.30 : Réponse indicielle en boucle fermée de la pression (vanne).

On voit bien que le système est parfaitement régulé.

IV.B.1.4. Système de régulation de température :

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

- La température opérationnelle du réservoir B101 ne doit pas dépasser les 65°C.

- La résistance chauffante ne doit être actionnée que si elle est complètement immergée dans le fluide.

Le circuit utilisé pour cette régulation est le suivant :

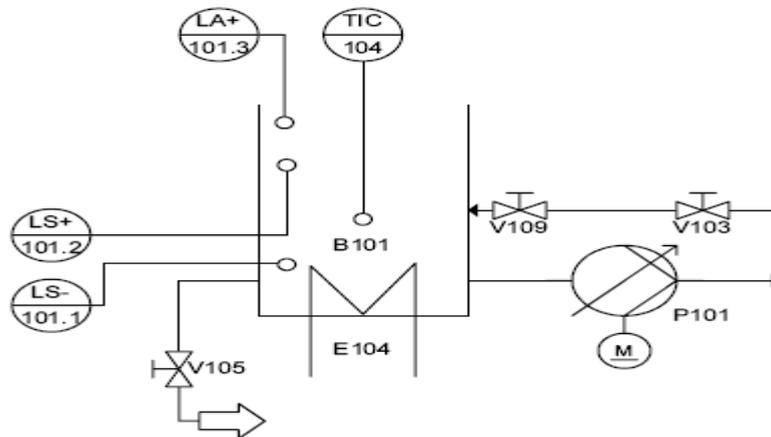


Figure IV.B.31 : Circuit relatif à la régulation de température.

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

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

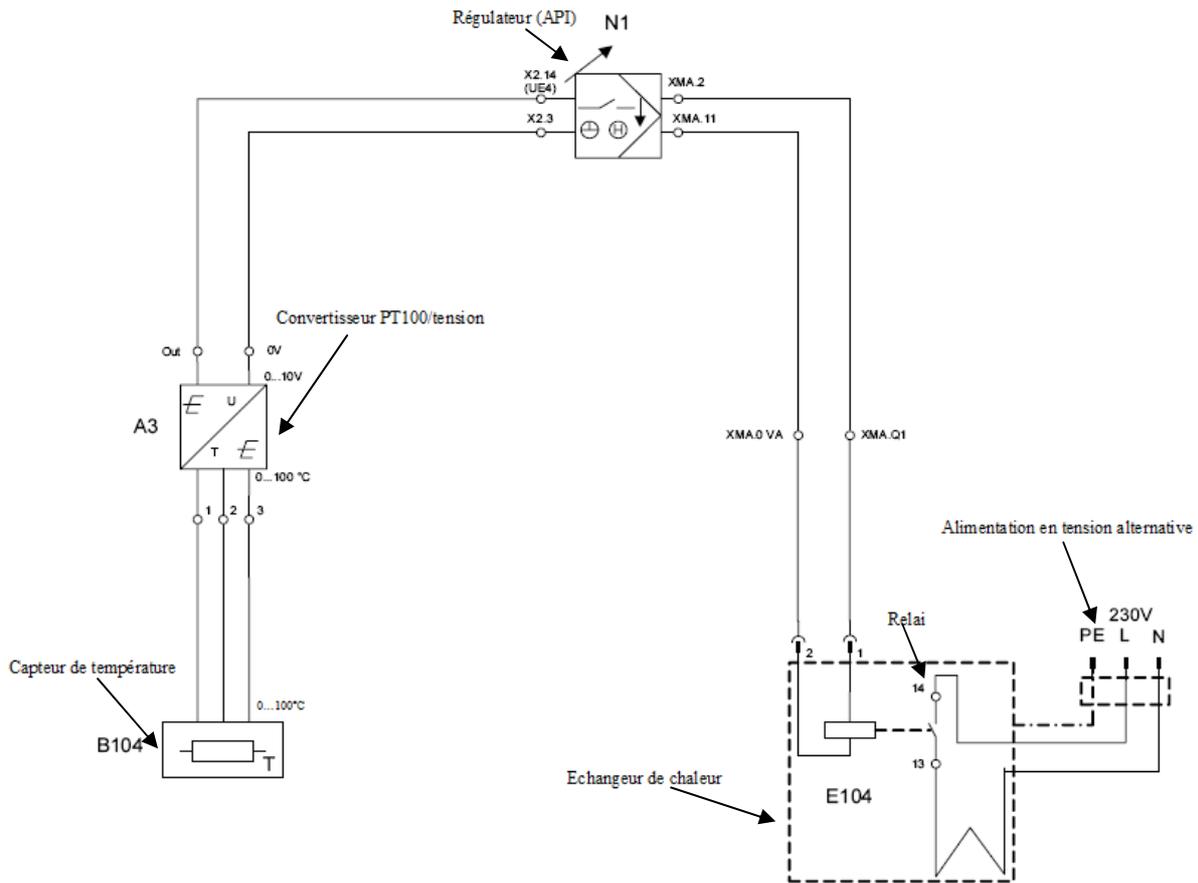


Figure IV.B.32 : Diagramme électrique de principe de la régulation de température.

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

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

IV.B.2. Description de l'application :

Nous avons étudié, plus haut dans ce chapitre, les différents types de régulation disponibles dans le système d'apprentissage didactique pour l'automatisation FESTO. Notre application consiste donc à élaborer un programme d'automatisation qui permet de gérer d'une manière optimale ces régulations individuellement et en cascade en ce qui concerne la régulation débit/niveau.

L'application est divisée en deux parties :

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

IV.B.2.1 Description des programmes STEP7 :

Vu la complexité du programme, nous avons adopté une organisation qui facilitera la compréhension et qui permettra une utilisation efficace. Pour une meilleure description du déroulement du programme, on se propose de diviser l'étude en plusieurs organigrammes complémentaires :

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

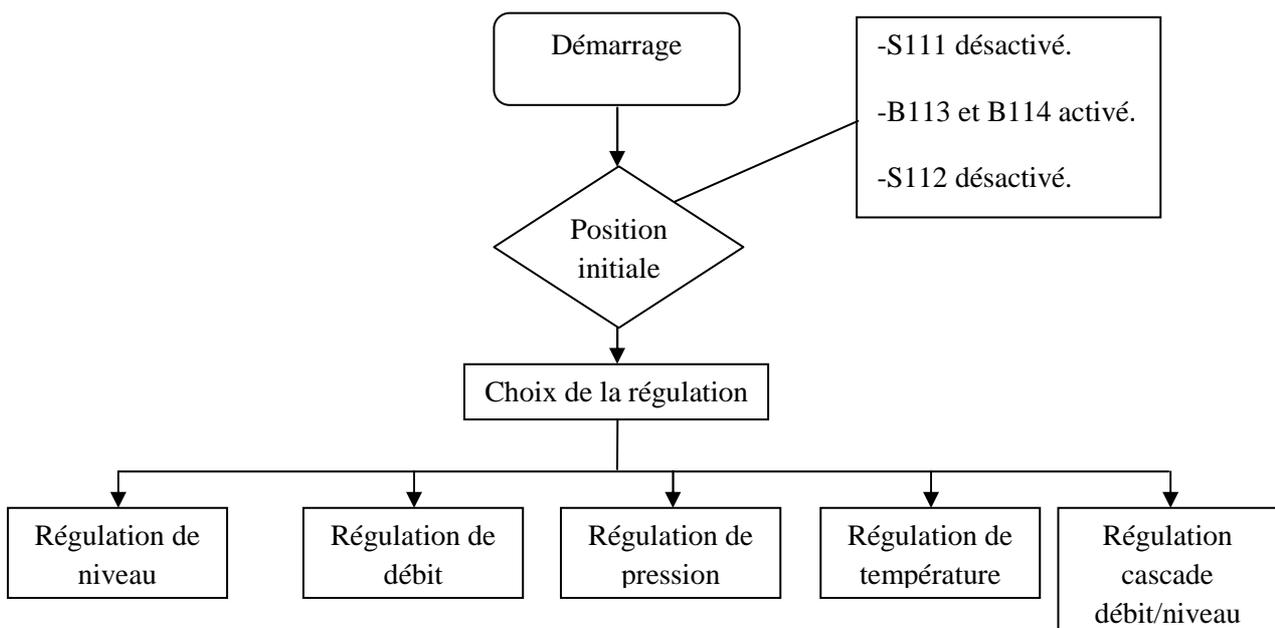


Figure IV.B.32 : Organigramme du choix des boucles de régulation.

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

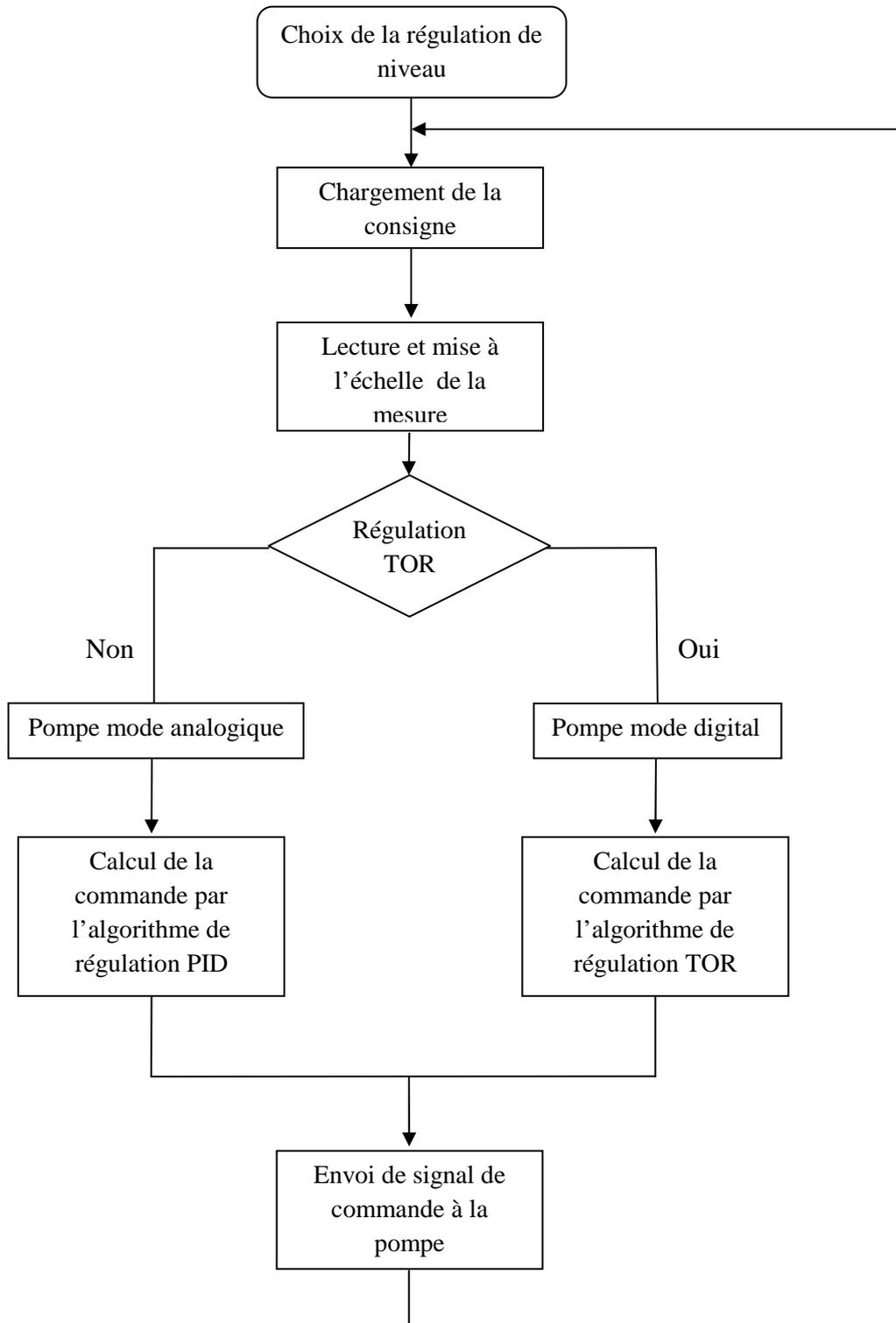


Figure IV.B.33 : Organigramme de la régulation de niveau.

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

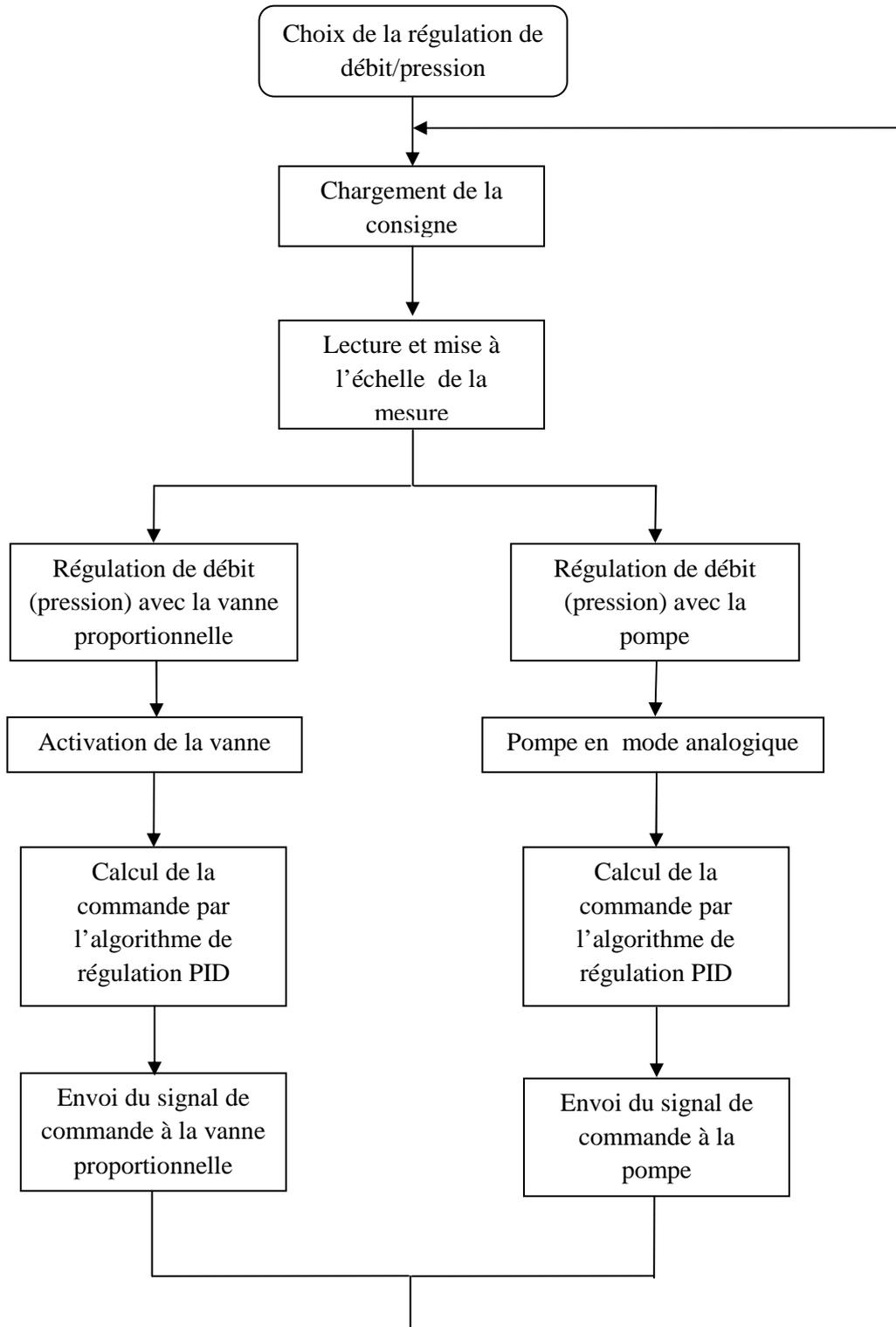


Figure IV.B.34 : Organigramme de la régulation de débit/pression.

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

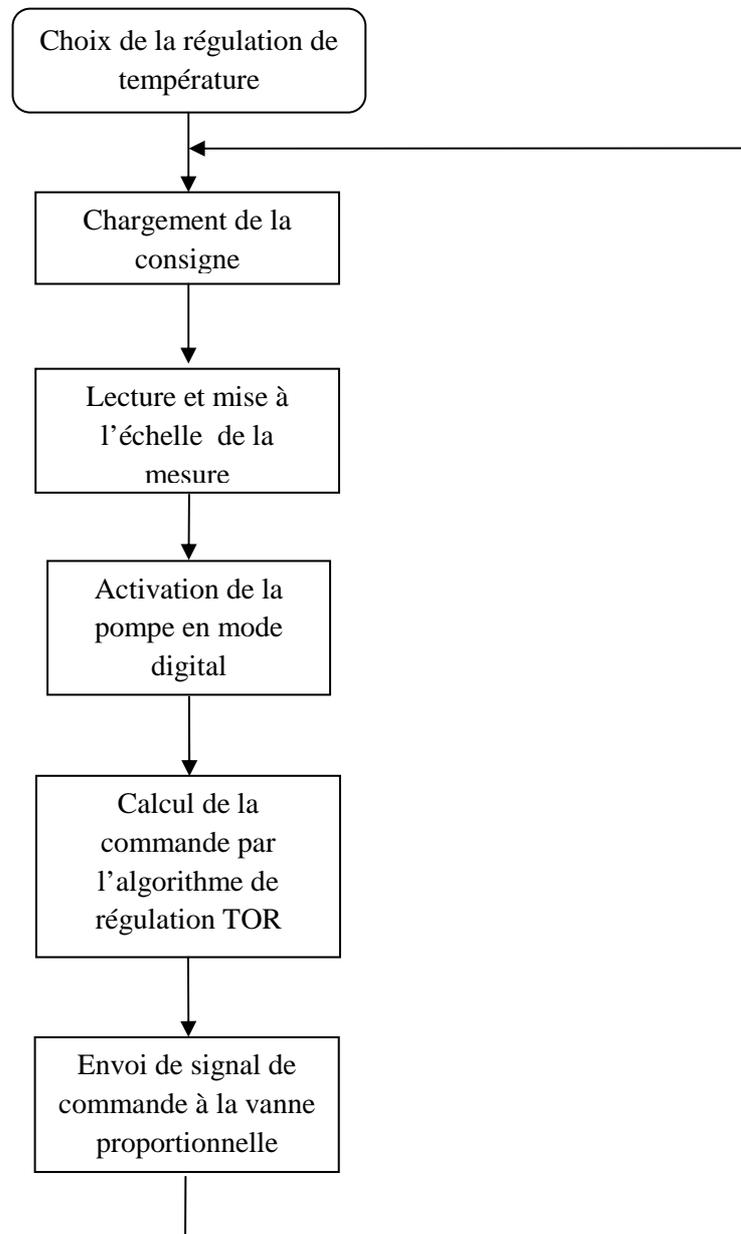


Figure IV.B.35 : Organigramme de la régulation de température.

Si cette fois le choix s'est arrêté sur la régulation cascade débit/niveau, l'organigramme de calcul de la commande est représenté sur comme suit :

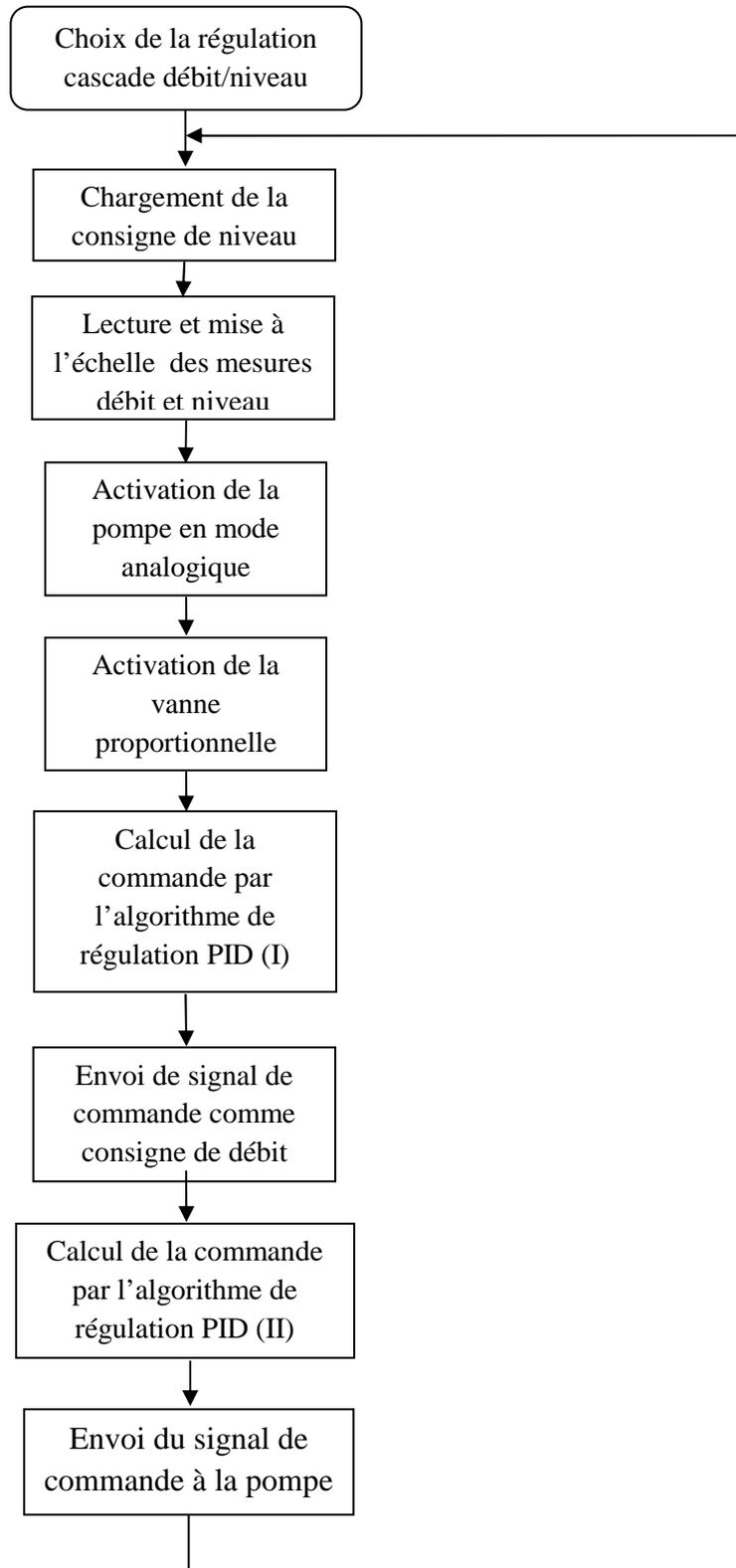


Figure IV.B.36 : Organigramme de la régulation cascade débit/niveau.

Le cheminement des organigrammes précédents est divisé en trois parties exécutées dans trois blocs d'organisation différents :

- ✓ Bloc d'organisation cyclique OB1 ;
- ✓ Bloc d'organisation d'alarme OB35.

1. Bloc d'organisation cyclique OB1 :

Le bloc OB1 à été utilisé pour faire appel aux différents blocs fonctionnels (FC) dont nous avons eu besoin :

a. Bloc fonctionnel de mise à l'échelle FC10 :

Dans cette fonction de types de mise à l'échelle ont été programmées :

➤ Mise à l'échelle des mesures :

Le signal de mesure, transmis par le capteur sous forme de tension standard 0...10V, doit être converti en une information numérique pour pouvoir être traité par l'automate, cette conversion est connue sous le nom de conversion analogique/numérique (CAN). Les automates programmables siemens possèdent leurs CAN intégrés. Ces derniers codent l'information sur 16 bits sur une plage de (0...27648). De ce fait pour obtenir la valeur physique de la mesure une mise à l'échelle est indispensable.

Exemple : la mise à l'échelle de la valeur transmise par le capteur de niveau ultrasonique est la suivante, en sachant que la plage de mesure de ce dernier est de 0...300mm :

$$Niveau_{réel} = \frac{Niveau_{numérique} \times 300}{27648}.$$

➤ Mise à l'échelle des erreurs et consignes:

Le bloc de régulation PID SFB41 travaille en valeurs normalisées entre 0 et 100%, de ce fait une consigne réelle entrée par l'utilisateur doit être normalisée avant d'être transmise au bloc SFB41. De même pour pouvoir visualiser l'erreur de régulation cette dernière doit aussi être normalisée.

Exemple : pour une régulation de niveau les mises à l'échelle respectives de la consigne et de l'erreur sont les suivantes :

$$Consigne_{Normalisée} = \frac{Consigne_{Niveau\ réel} \times 100}{300}.$$

$$Erreur_{Réelle} = \frac{Erreur_{Normalisée} \times 300}{300}.$$

b. Bloc fonctionnel pour le choix du type de régulation FC11 :

Dans ce bloc s'effectue le choix du type de régulation (d'un point de vue actionneur), ce choix déterminera ensuite la configuration nécessaire pour le bon déroulement de la boucle de régulation :

Par exemple si le choix s'est arrêté à la régulation de niveau analogique, la pompe doit être mise en mode analogique. Bien sûr d'autres actions doivent être effectuées dus aux conflits surgissant lors des permutations entre les différentes boucles de régulations.

c. Bloc fonctionnel pour l'arrêt d'urgence FC12 :

Dans ce bloc s'exécute le programme d'arrêt d'urgence, en appuyant sur le bouton « STOP » du panneau de contrôle, toutes les actions entreprises doivent être suspendues. Bien entendu, le bouton « START », du même panneau de contrôle, permettra de reprendre la régulation là ou elle à été interrompue.

2. Bloc d'organisation d'alarme OB35 :

Ce bloc d'organisation est utilisé pour faire appel au bloc de régulation PID SFB41. En effet un seul bloc PID a été utilisé, par souci d'optimisation, pour toutes les boucles de régulation. Le conflit est évité par un choix au préalable de la régulation (OB35) et du type de cette dernière (FC11).

IV.B.2.2 Description des programmes WINCC :

Pour la simulation et la visualisation des programmes STEP7 et afin de faciliter la navigation à travers les différentes boucles de régulation, nous avons confectionné une interface graphique interactive à l'aide du logiciel WINCC. Cette interface graphique est la succession de plusieurs vues (fenêtres) permettant la navigation entre les différentes régulations.

La première vue de l'interface propose :

- Le synoptique général de la station, ce dernier contiendra des objets animés (pompe, flotteurs, vanne tout ou rien....) ;
- Des sorties de visualisations des quatre mesures (niveau, débit, pression, température) ;
- Un menu contenant les cinq types de régulation : niveau, débit, pression, température, cascade (débit+niveau) ;
- Un bouton 'suivant', qui une fois le types de la régulation choisi, nous dirigera vers la vue correspondante.

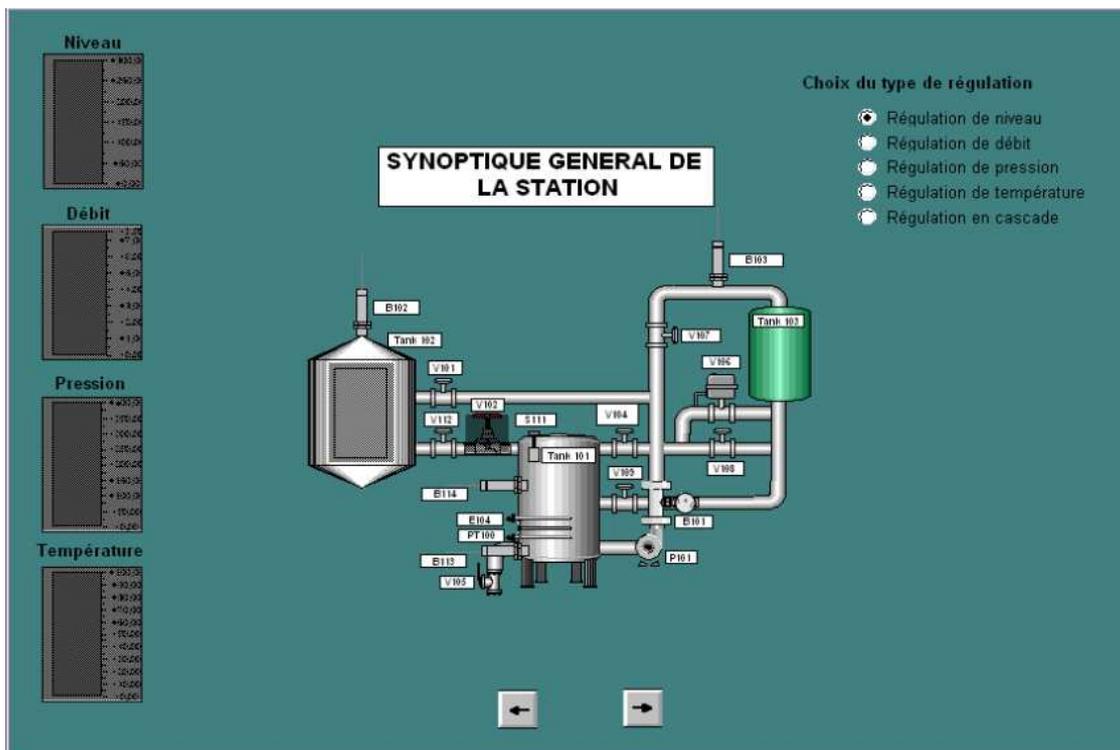


Figure IV.B.38 : Vue d'accueil de l'interface.

Si par exemple le choix s'est arrêté à la régulation de niveau, le bouton 'suivant' nous dirigera vers :

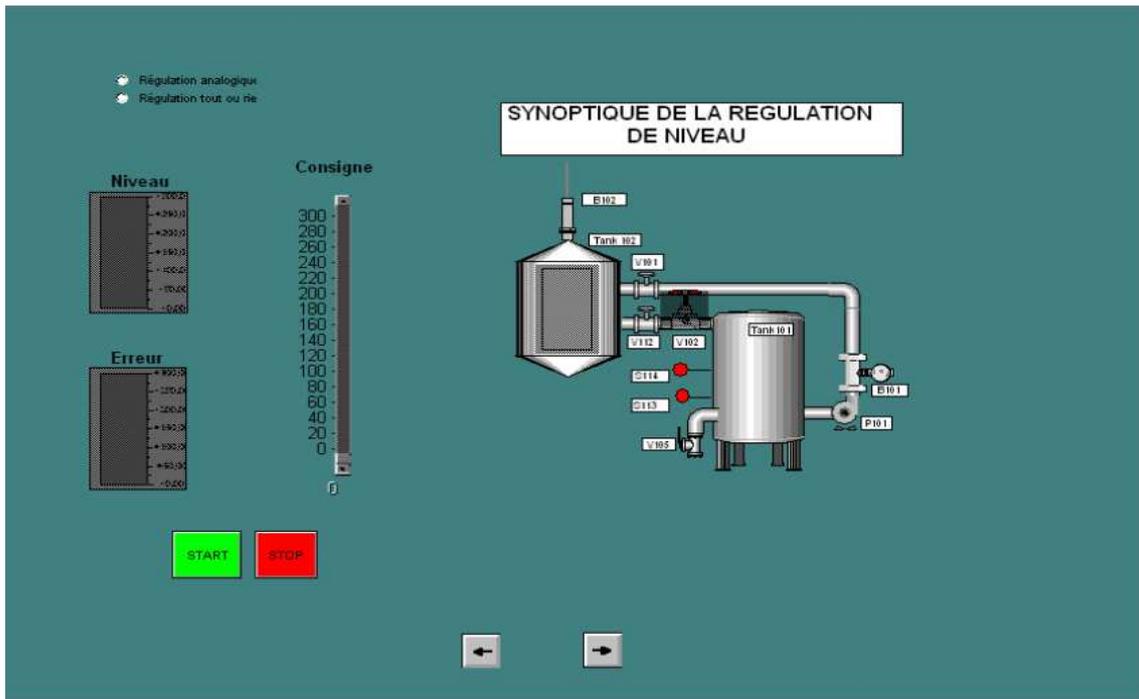


Figure IV.B.39 : Vue relatif à la régulation de niveau.

Cette vue propose :

- Le synoptique spécifique à la boucle de régulation de niveau, ce dernier contiendra des objets animés (pompe, flotteurs, vanne tout ou rien....) ;
- Des sorties pour la visualisation de la mesure (niveau) ainsi que de l'erreur de régulation.
- Un menu contenant les deux types de régulation possibles à savoir la régulation PID et la régulation TOR.
- Un défileur qui permet d'entrer la consigne de niveau.
- Deux boutons 'start' et 'stop' qui permettent respectivement de démarrer et de stopper la régulation.
- Deux boutons 'suivant' et 'précédant' qui permettent respectivement de revenir à la vue d'accueil, de passer à la vue de visualisation des courbes :

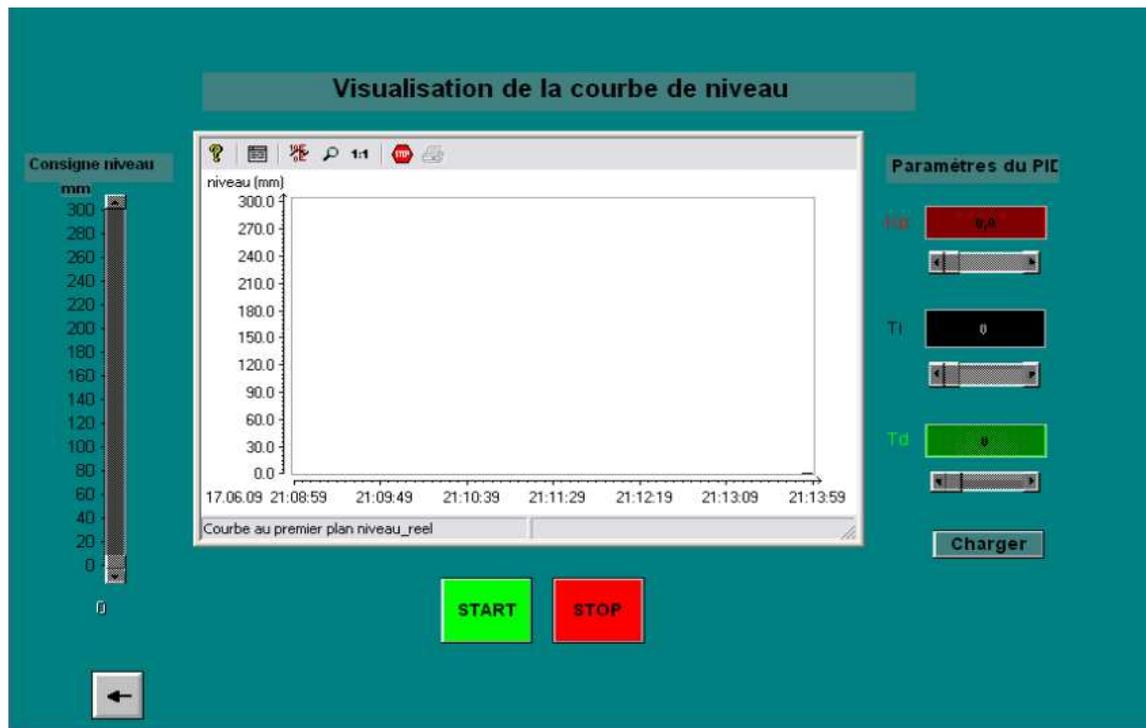


Figure IV.B.40 : Vue relatif à la visualisation des courbes de niveau.

Cette dernière propose aussi la possibilité :

- D'arrêter et de redémarrer la régulation (boutons 'start' et 'stop') ;
- De revenir à la vue précédente (bouton 'précédent' ;
- De donner la main à l'utilisateur afin de modifier les paramètres de la régulation PID.

La navigation à travers les autres vues s'effectue de la même façon que celle décrite ci-dessus.

Conclusion :

Nous avons pu voir à travers ce chapitre l'application de concepts théoriques d'identification des systèmes ainsi que de régulation à un système réel tout en obtenant des résultats satisfaisants et ceci à l'aide des outils de programmation offerts par STEP7 ainsi que la supervision des processus à travers l'interfaçage graphique réalisé à l'aide de WinCC.

Conclusion générale :

Munis de deux logiciels très performants, STEP7 et WinCC, les automates programmables industriels Siemens forment des unités de traitement et de commande de grande flexibilité.

En effet, le logiciel de programmation STEP7 permet l'accès de base aux automates programmables de la gamme SIMATIC, pour sa programmation en différents langages. Il assure également la fonction de moyen de communication en prenant en compte leurs réseaux. Le logiciel de conception des interfaces homme-machine WinCC est quant à lui, un logiciel d'ingénierie et de supervision, qui offre des fonctions de surveillance d'automatismes.

Notre projet nous a permis d'étudier la station didactique de contrôle des processus, FESTO, 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'identification, la régulation et la réalisation d'un programme de gestion, de tous les systèmes de contrôle disponibles sur la station FESTO, 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 ;
- L'ajout d'un convertisseur, qui permettra de commander la résistance chauffante de manière analogique.

Bibliographie :

- [1] G.MICHEL, « Les A.P.I Architecture et application des automates programmables industriels », Edition DUNOD, 1987
- [2] M.BERTRAND, « Automates programmables industriels », Technique de l'ingénieur, Vol. S 8 015
- [3] P.JARGOT, « Langages de programmation pour API, Norme IEC 1131-3 », Technique de l'ingénieur, Vol. S 8 030
- [4] SIEMENS, « PLC S7-300, CPU spécifications », SIMATIC, 2001
- [5] SIEMENS, « S7-300 and M7-300 Programmable controleurs module specifications », SIMATIC, 2001
- [6] SIEMENS, « Logiciel de base pour S7 300/400 Régulation PID », SIMATIC, 2001
- [7] SIEMENS, « Technique de régulation avec STEP7 », SIMATIC, 2004
- [8] SIEMENS, « WinCC Getting started », SIMATIC HMI, 2000
- [9] SIEMENS, « WinCC V6.0, Documentation de base », SIMATIC HMI, 2003
- [10] E.OSTERTAG, « Systèmes et asservissements continus », Ellipses, 2004
- [11] J.HELMICH, « Compact workstation manual », FESTO, 2004

ملخص :

العمل المنجز في المذكرة يتمحور على إستعمال مسير صناعي مبرمج " سيمنس ". عندما اكتسبت دايرة الأوتوماتيك للمدرسة الوطنية العليا المتعددة التقنيات نموذج محطة تسيير وضبط الانظمة الصناعية مستوى الماء التصريف، الضغط، درجة الحرارة(، قمنا بإنجاز برنامج تسيير، تطابق وضبط مختلف الانظمة المحتواة في المحطة.

للتطابق و التعرف على خصائص الطوابط المطبقة استعنا بطريقة زيجلرنكلس.

الكلمات المفتوحة:

مسير صناعي مبرمج " سيمنس " ، STEP7،Wincc ، الاعددة و المحركات فسطو.

RESUME :

Le travail présenté dans ce mémoire est basé essentiellement sur l'utilisation des automates programmables SIEMENS. Le Département d'Automatique de l'Ecole Nationale Supérieure Polytechnique ayant acquis une station de contrôle des processus (débit, niveau, pression et température), FESTO, nous avons entrepris la réalisation d'un programme de gestion, d'identification et de régulation des différents sous-systèmes constituant cette installation.

Pour l'identification, et la détermination des paramètres des régulateurs (P, PI, PID) appliqués, nous avons utilisé les méthodes de compensation des pôles et Ziegler-Nichols ainsi que la méthode du lieu d'Evans.

Mots clés :

Automate programmable Siemens S7-313C, STEP7, WinCC, capteurs et actionneurs FESTO.

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 management, identification and control of the various subsystems comprising in the installation.

For the identification and determination of the parameters of regulators (P, PI, PID) applied, we used the methods to compensate for the poles and Ziegler-Nichols method and the place of Evans.

Key words:

Siemens PLC S7-313C, STEP7, WinCC, sensors and actuators FESTO.

Liste des annexes :

Annexe A : Caractéristiques techniques des capteurs et des actionneurs de la station PCS.....	1
Annexe B : Identification d'un système d'ordre 1.....	18
Annexe C : Synthèse d'un régulateur PI dans le lieu d'Evans.....	19
Annexe D : Méthode de Ziegler-Nichols.....	20
Annexe E : Programme STEP7 pour la simulation d'un circuit RC.....	21
Annexe F : Programmes STEP7 pour la gestion et la régulation de la station PCS.....	23

Caractéristiques techniques des capteurs et des actionneurs de la station PCS

A.1 Les capteurs :

a. Les capteurs analogiques :

L'installation possède quatre capteurs et jauges de pression analogiques, deux détecteurs de proximité numérique ainsi que deux flotteurs.

1. Le capteur à ultrason :



Figure A.1 : Capteur ultrasonique de niveau.

Le principe de fonctionnement du capteur ultrasonique est basé sur la génération d'ondes acoustiques et sur la détection de leurs réflexions sur les différents objets présents sur leur chemin.

En effet, un générateur d'ondes acoustiques est actionné pendant un certain laps de temps et émet donc des impulsions ultra soniques inaudibles à l'oreille humaine. Ces ondes émises se réfléchissent sur des objets et l'écho de retour est ainsi détecté par un récepteur. Le temps mis entre l'émission et la réception d'une onde acoustique est évalué électroniquement. Le signal de sortie (courant électrique) n'est proportionnel au temps de parcours de l'impulsion ultrasonique, que dans une certaine plage de mesure.

Avec ce type de capteur deux types de setup sont possibles qui correspondent à deux types de mesure différentes :

- Mesure de la distance entre le capteur et un objet (setup MANUFACTURER).
- Mesure de niveau (setup ADIRO).

Le capteur disponible dans l'installation nous permet une mesure de niveau, il fonctionne suivant le setup ADIRO, ce dernier fait affecter au niveau maximum un signal de sortie (courant électrique) maximum, et respectivement un signal de sortie minimum correspond au niveau minimum.

Annexe A

Le capteur délivre donc un signal de sortie sous forme de courant (4...20mA), ce dernier passe par un convertisseur courant tension (0...10V) afin d'être transmis à l'automate.

Le capteur est positionné dans un réservoir à 330 mm du fond, la distance qui le sépare des murs de ce dernier doit être de 75mm.

Les caractéristiques techniques du capteur sont résumées dans le tableau suivant :

Paramètres	valeurs
Classe de protection	IP 67
Poids	67g
Température ambiante	-25 à 70°C
Précision	+ ou + 2.5%
Tension opérationnelle UE	24VDC
Tension opérationnelle UB	20...30V
Ondulation résiduelle admissible	10%
Courant idéal de consommation IO	<50mA
Permutation des sortie (NO/NF)/ sortie fréquentielle Courant opérationnelle Chute de tension	150mA 3V à 150mA
Sortie analogique Plage de courant Plage de mesure	4...20mA 0...300mm
Capteur actif	Voltage opérationnelle/Haute impédance Courant d'entrée 16mA
Capteur inactif	0...3V Courant d'entrée 11mA

Tableau A.1 : Caractéristiques technique du capteur niveau.

2. Le capteur de débit :



Figure A.2 : Capteur de débit type 2.

Le liquide transparent est admis et guidé, dans un mouvement circulaire, à l'intérieur d'une chambre de mesure, de ce fait le liquide actionne un mini rotor à trois ailettes. La vitesse du rotor est proportionnelle à celle de l'écoulement, et elle est détectée par le biais d'un système optoélectronique à infrarouges.

Annexe A

Le capteur délivre, sous la forme d'une onde carrée, un signal qui peut être mesuré avec un oscilloscope. Ce dernier est soit directement transmis comme entrée binaire à l'automate programmable, soit transmis par le biais d'un convertisseur fréquence/tension (0...10V).

Les caractéristiques du capteur sont résumées dans le tableau suivant :

Paramètres	valeurs
Voltage opérationnelle admissible	8...24VDC
Courant de consommation	18...30mA
Plage de fréquence (signal de sortie)	40...1200Hz
Charge maximum	2.2k
technologie	Infrarouge (optoélectronique)
K-facteur	8000
Plage de mesure	0...7.5 l/min
Fiabilité de la mesure	+/- 1% de la valeur mesurée
Erreur de linéarité	+/- 1% de la valeur mesurée
viscosité	< 15 cSt
Pression opérationnelle	10 bar au maximum
Plage de température standard	-40...+80 °C
Protection d'inversion de la polarité	oui
Matériel de construction	
Parties médianes	PVDF
Dimensions	
Longueur	47 mm
Fils	M20x2
Connexions électriques	câbles

Tableau A.2 : Caractéristiques technique du capteur de débit.

3. Le capteur de pression :

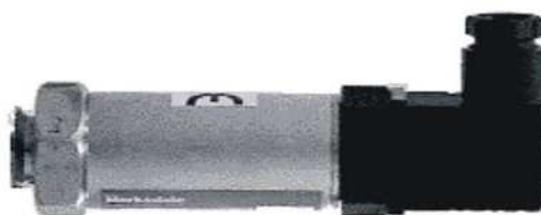


Figure A.3 : Capteur de pression.

Le capteur de pression analogique est un capteur de pression relative piézorésistif, avec un amplificateur intégré et muni d'un compensateur de température, le tout dans un corps en aluminium.

La pression à mesurer agit sur l'élément piézorésistif, créant ainsi un signal qui est émis à travers l'amplificateur sous la forme d'un signal standard 0...10V.

Annexe A

Les caractéristiques techniques sont ainsi résumées dans le tableau suivant :

Paramètre	valeurs
Plage de mesure	0...400 mbar
Surcharge	5.5 mbar
Tension d'alimentation UB	13...30VDC
Signal de sortie	0...10VDC
Courant de consommation	5 mA au maximum
Erreur de linéarité	+/- 1% f.s
Temps de réponse	1 ms
Membrane	Acier inoxydable
Plage de température standard	0...65°C
Connexions électriques	Plug 3-PIN
Processus de connexion	Filetage vise externe en acier inoxydable : G 1/2"
Poids	250g

Tableau A.3 : Caractéristiques technique du capteur de débit.

4. Le capteur de température :



Figure A.4 : Capteur de température.

Le principe du capteur de température est basé sur le changement de la valeur de la thermo-résistance le constituant en fonction de la température.

Le capteur est constitué d'un bouclier sous forme de tube, d'une tête de connexion et d'un élément de mesure (thermo-résistance de type PT100).

La valeur de la résistance passe par un convertisseur PT100/tension (0...10V) avant d'être transmise à l'API.

Les caractéristiques techniques du capteur sont les suivantes :

Paramètres	Valeurs
Design	DIN 43 763
Plage de mesure	-50...+150°C
Type de résistance de mesure	Pt 100
Tolérance 0°C 100°C	+/- 0.12 Ω +/- 0.30 Ω
Matériel de construction Boitier Tube de protection	Acier inoxydable Acier inoxydable

Dimensions	
Longueur	100 mm
Longueur de l'élément de mesure filetage	145 mm G 1/2"
Connexion électrique	Câble de longueur de 750 mm

Tableau A.4 : Caractéristiques technique du capteur de température.

b. Les capteurs digitaux :

1. Les Détecteurs de proximité capacitifs :



Figure A.5 : Détecteur de proximité capacitif.

Le principe de fonctionnement d'un capteur capacitif de proximité est basé sur l'évaluation de la variation de la capacité d'un condensateur dans un circuit RC résonant. En effet l'augmentation de la valeur de la capacité, due au rapprochement d'un objet, mène à un changement de l'oscillation du circuit RC.

La variation de la capacité dépend principalement de la distance, des dimensions et de la constante diélectrique du matériau détecté.

La sortie du capteur est un transistor PNP utilisé en commutation sous la forme d'un contact normalement ouvert.

Les caractéristiques techniques sont résumées dans le tableau suivant :

Paramètres	Valeurs
Tension opérationnelle admissible	10...55VDC
Interrupteur de sortie	PNP, contact normalement ouvert
Distance nominale de fonctionnement	2...8 mm
Hystérésis nominal	3...15%
Courant de commutation maximum	200 mA
Fréquence de commutation maximum	300 Hz
Courant de consommation en fonctionnement lent	7 mA
Température ambiante admissible	20...70°C
Degré de protection	IP 65
Matériel	Thermoplastique
poids	0.20 kg
Connexion électrique	Câble, 2000 mm de longueur

Tableau A.5 : Caractéristiques technique du détecteur de proximité.

2. Les flotteurs :

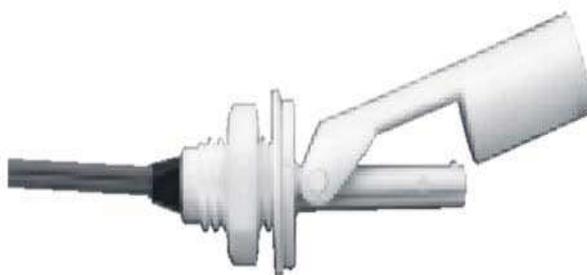


Figure A.6 : Flotteur.

Le principe de fonctionnement est simple, étant donné que le flotteur est destiné à être monté sur le mur latéral d'un réservoir. Le mode d'action est simple et basé sur la variation du niveau du liquide. L'aimant intégré dans le corps du flotteur active un interrupteur hermétiquement isolé.

Les caractéristiques techniques sont résumées dans le tableau suivant :

Paramètres	Valeurs
Matériel	
Corps flottant	Versaplast
Câble	PVC
Température	-40...121°C
Densité du liquide minimum	0.8
Pression opérationnelle	7bar
Interrupteur de lecture	20VA
Câble (longueur approximative 0.6 m)	22AWG
Corps flottant	55mm
Protection DIN40050	IP64
Poids (approximatif)	80g

Tableau A.6 : Caractéristiques technique du flotteur.

3. Flotteur anti-débordement :



Figure A.7 : Flotteur anti-débordement.

Le liquide en montant pousse vers le haut l'élément flotteur et actionne ainsi un interrupteur, Les caractéristiques techniques sont résumées dans le tableau suivant :

Annexe A

Paramètres	valeurs
Matériels	
Pipe de commutation	Polysulfon
Corps flottant	Polysulfon
Température	
Câble	-40...+80°C
Fil standard	-40...+107°C
Profondeur d'immersion du corps flottant (densité 1)	15 mm
Densité minimal du liquide	0.75
Type de l'interrupteur	SPST 50 VA pour le câble SPST 20 VA pour le fil
Connexion électrique (longueur approximative 0.6m)	0.34 mm ² PVC pour le câble AWG 22 PVC pour le fil
Protection DIN 40050	IP64
Poids	20g
Filetage.	1/8" NPT
Pression opérationnelle	3 bar

Tableau A.7 : Caractéristiques techniques du flotteur anti-débordement.

A.2. Les actionneurs :

a. Actionneurs analogiques :

1. La pompe :



Figure A.8 : La pompe centrifuge.

La pompe est une pompe centrifuge à eau, le moteur de la pompe est adapté pour un fonctionnement en continu. Les caractéristiques techniques sont résumées dans le tableau suivant :

Paramètres	Valeurs
Pompe	Plastique (PPA, GF 30%)
Arbre de transmission	Acier inoxydable
Plaque d'usure	Acier inoxydable
Joint annulaire	EPDM
Roue à lames	Corps : plastique (PPA, GF 40%) Matériaux magnétique : ferrite
Moteur à bride	Plastique (PA, GF 30%)
Boitier du moteur	Plastique (PA, GF 30%)
Vis	acier
Moteur	Moteur à aimant permanent 12/24V
Support du moteur	Aluminium
Classe de protection	IP67 (DIN 40050)
connexion	20 mm (3/4")
Suppression du bruit	EN 55014
Plage de température	
Fluide	-40...100°C
Ambiante	-40...70°C
Pression maximale	2.5 bar
Tension d'alimentation	24V
Puissance	26W

Tableau A.8 : Caractéristiques techniques de la pompe.

2. Vanne proportionnelle :



Figure A.9 : Vanne proportionnelle.

La vanne proportionnelle facilite le contrôle continu du débit des fluides et gaz neutres. Elle est actionnée directement, à travers un élément de contrôle électronique (1), par une électrovalve (vanne+solénoïde) 2/2 voies (2).

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.

Les caractéristiques techniques de la vanne proportionnelle sont résumées dans le tableau suivant :

Paramètres	Valeurs
Tension opérationnelle admissible (connectée au contrôleur électronique)	24VDC
Puissance de consommation (solénoïde)	8W
Type d'action	Opération en continu
Degré de protection	IP 64f
Taille nominale	6mm
Pression opérationnelle	0...0.5 bar
Température opérationnelle	55°C au maximum
Sensibilité de la réponse	0.5% de la valeur finale
Précision dans la répétition	0.5% de la valeur finale
fluides	Eau, air comprimé
Matériels	
Extérieur de la vanne	Laiton
Intérieur de la vanne	Acier inoxydable
Joint	FPM
Dimensions	
Hauteur avec le contrôleur électronique	108mm
Longueur	46mm
Connexions des pipes	G ¼
Connexions électriques	Pin pour le contrôleur électronique

Tableau A.9 : Caractéristiques techniques de la Vanne proportionnelle.

Annexe A

Les caractéristiques techniques du contrôleur électronique sont quand à eux résumées dans le tableau suivant :

Paramètres	Valeurs
Tension d'alimentation admissible	24....28VDC au maximum
Ondulation résiduelle	10% au maximum
Signal d'entrée	0...10V
Résistance d'entrée	16.8k Ω
Consommation en puissance	0.5W
Consommation en courant à 24V	18mA
Température opérationnelle	0...55°C
Matériel	Plastique
Connexions électriques	Câble de 7mm raccordé au terminal

Tableau A.10 : Caractéristiques techniques du contrôleur électronique.

b. Actionneurs digitaux :

1. Résistance chauffante :



Figure A.10 : Résistance chauffante.

L'unité de chauffage est alimentée avec une tension 230VAC, elle est activée et désactivée par le biais d'un relais fonctionnant quant à lui par une tension de 24VDC.

Les caractéristiques techniques sont les suivantes :

Paramètres	Valeurs
Consommation	1000W/203VAC
Tension de contrôle	
Dimensions	
Elément chauffant	150 mm x ϕ 20 mm
Filetage	G 1/2 "
Matériels (boitier, élément chauffant)	Acier inoxydable
Connexions	
Unité de chauffage	Câble avec un Plug de 2000mm de longueur
Connexion de contrôle	3 pins

Tableau A.11 : Caractéristiques techniques de la résistance chauffante.

2. Vanne automatique tout ou rien :

Cet équipement de contrôle consiste en une vanne à billes en cuivre ouverte/fermée par un actionneur quart de tour à double effet de type SYPAR, contrôlé par une électrovanne 5/2 voies de type NAMUR. Le tout est relié un indicateur de position.

a. Electrovanne de type NAMUR :

L'électrovanne est constituée par un distributeur pneumatique 5/2 voies (1), actionnée par un électroaimant (Solénoïde+piston) (2).

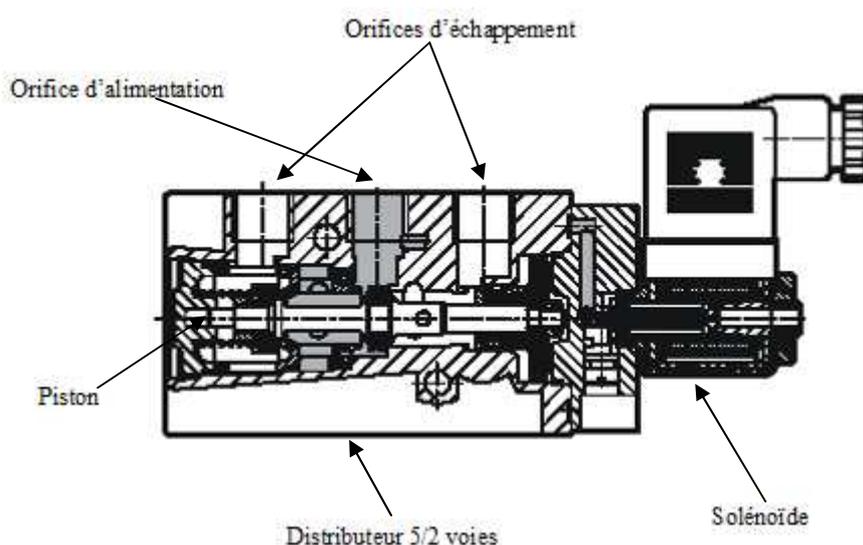


Figure A.11 : Electrovanne de type NAMUR.

Les caractéristiques techniques du solénoïde sont les suivantes :

Paramètres	Valeurs
Alimentation	24VDC
Ondulation admissible de la tension	+/-10%
Consommation en puissance	5.65 W
Classe de protection EN 60 529	IP 65
Connexion électrique	Câble moulé 3x0.75mm ² , 1m ou 5m de longueur
Température ambiante	-5...40°C
Temps de ramassage minimum	12ms
Matériels	Acier, cuivre...
Poids	0.175kg

Tableau A.12 : Caractéristiques techniques du solénoïde.

b. Semi-disque rotatif type SYPAR :



Figure A.12 : Semi-disques rotatifs type SYPAR.

Cet équipement est utilisé essentiellement pour des vannes avec angle de rotation limité à 90° ; c'est l'actionneur double effet quart de tour de la vanne à billes, muni d'un port pattern de type NAMUR (2).

Les caractéristiques techniques sont résumées dans le tableau suivant :

Paramètres	Valeurs
Dimensions	50 mm
Milieu	Air sec, avec ou sans huile, ou gaz compatible avec les actionneurs hydraulique
Consommation (volume) par cycle	0.06l/cycle
Fil de connexion	1/8"
Temps de commutation	
Ouverture	0.04 s
Fermeture	0.04 s
Durée de vie	1 Mio.cycles
Température opérationnelle	-20...80°C
Résistance à la corrosion	FN 940 070 Part 1, KBK3
Poids	0.75g

Tableau A.13 : Caractéristiques techniques du semi-disque rotatif type SYPAR.

Dans le tableau qui va suivre sont résumées les pressions nécessaires au bon fonctionnement du semi-disque rotatif :

Paramètres	Valeurs
Pression de fonctionnement en repos	0.4 bar
Pression de fonctionnement en travail	0.25 bar
Pression opérationnelle minimale	1 bar
Pression opérationnelle nominale	5.6 bar
Pression opérationnelle maximale	8.4 bar

Tableau A.14 : Pressions de fonctionnement.

c. Indicateur de positions :



Figure A.13 : Indicateur de positions.

L'indicateur de position consiste en deux micro-interrupteurs électriques avec levier à roulement. Les caractéristiques techniques sont les suivantes :

Paramètres	Valeurs
Elément de commutation	Contact normalement fermé
Pouvoir de coupure	16A, 250 VAC
Tension d'alimentation	0...30 VDC
Plage de température	-25...100°C
Classe de protection	IP65
Classification de la résistance à la corrosion CRC	2
Câble sécuritaire à vis	M20 x 1.5
Durée de vie minimale	2×10^5

Tableau A.15 : Caractéristiques techniques de l'indicateur de position.

A.3. Les éléments de connexion (API/ capteurs+actionneurs) :

1. Les interfaces (terminal analogique d'entrées/sorties) :

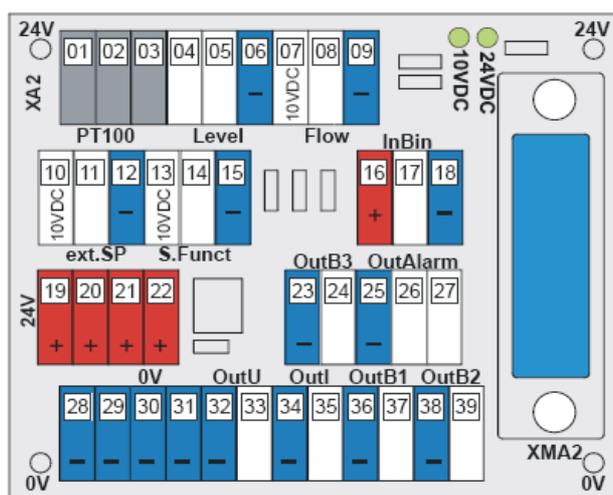


Figure A.16 : Terminal analogique d'entrées/sorties

Le terminal analogique d'entrées/sorties permet de relier les capteurs et actionneurs à l'API à travers un câblage SYSLINK. Une alimentation de tension intégrée 10VDC facilite la connexion des capteurs et les générateurs de consigne externes nécessitant cette tension. Les caractéristiques techniques sont les suivantes :

Paramètre	valeur
Tension opérationnelle permissible	24VDC
Entrées analogiques	5
Entrées binaires	1
Sorties analogiques	2
Sortie binaires	4
Tension opérationnelle affichée	LED verte 24VDC
Tension constante affichée	LED verte 10VDC
Connexion a l'API	24-pin centroniques

Tableau A.17 : Caractéristiques techniques du terminal analogique.

2. Le terminal de conversion des signaux analogiques :

Le terminal de conversion des signaux analogiques est constitué de plusieurs types de convertisseurs analogiques, qui permettent, d'un coté, de convertir les signaux envoyés par les différents capteurs analogiques constituant l'installation en signaux standard 0...10V et d'un autre coté de transformer les signaux analogiques envoyés par l'API vers les éléments de contrôles en signaux standard 0...24V ou 0...10V.

- a. Les convertisseurs :
 - i. Convertisseur PT100/tension :



Figure A.15 : Convertisseur PT100/tension.

Ce convertisseur transforme la valeur de la résistance PT100 en une tension standard 0...10V, la plage de transformation est de 0...100°C. Les caractéristiques techniques sont résumées dans le tableau suivant :

Paramètres	Valeurs
Plage de température	0...100°C
Alimentation en tension	DC 24V (+/-10%)
Courant nominal	30A
Sortie	0...10V
Charge	500kOhm
Erreur de sortie	Ub/ max.20mA
Erreur de transmission	0.3%
Coefficient de température	< 0.02% /K
Compatibilité électromagnétique (IEC 801-2/4/5)	EN 50082 T2 (E3.94)
Température ambiante permissible	0...55°C
Poids	29.8g
Couleur	Grise

Tableau A.18 : Caractéristiques techniques du convertisseur PT100/tension.

- ii. Convertisseur courant/tension :



Figure A.16 : Convertisseur courant/tension.

Annexe A

Ce convertisseur transforme la valeur (courant) transmise par le capteur de niveau ultrasonique en tension standard 0...10V. Les caractéristiques techniques sont résumées dans le tableau suivant :

Paramètres	Valeurs
Signal d'entrée	4...20 mA
Courant d'entrée	22 mA
Résistance d'entrée	< 400kOhm
Chute de tension en ligne maximum	< 8V
Signal de sortie	0...10V
Charge	>2kOhm
Détection de fil cassé	Led verte éteinte
Erreur de transmission (basée sur la valeur finale)	< 0.15%/0.1%
Coefficient de température (basée sur la valeur finale)	< 0.02%/K
Barrière de fréquence	1kHz
Isolation en tension entrée/sortie	4kV, 50Hz, 1min
Alimentation en tension	DC 20V, 30V
Température ambiante permissible	0...55°C
Couleur	Grise

Tableau A.19 : Caractéristiques techniques du convertisseur courant/tension.

iii. Convertisseur fréquence/tension :



Figure A.18 : Convertisseur fréquence/tension.

Ce convertisseur transforme la valeur (fréquence) transmise par le capteur de débit fréquentiel en tension standard 0...10V. Les caractéristiques techniques sont résumées dans le tableau suivant :

Paramètres	Valeurs
Température ambiante permissible	55°C
Alimentation en tension	20,30VDC
Consommation en courant	12mA
Erreur de linéarité	<0.1%
Erreur de transmission	<0.1%
Entrée	
Signal carré	0...1kHz
Amplitude	6V _{SS} - 30V _{SS}
Sortie	
Signal de sortie	0...10V

Annexe A

Charge	>2kOhm
Couleur	Grise

Tableau A.20 Caractéristiques techniques du convertisseur fréquence/tension.

a. Le contrôleur du moteur de la pompe :



Figure A.19 : Contrôleur du moteur de la pompe.

Ce contrôleur permet de faire varier la tension d'alimentation du moteur de la pompe, et donc sa vitesse. Dans le circuit de bord il y'a un potentiomètre qui permet d'ajuster le contrôleur. Le but de cet ajustement est de faire correspondre la tension 0V de l'entrée à celle de la sortie, et la tension 24V de sortie à une tension de 10V à l'entrée.

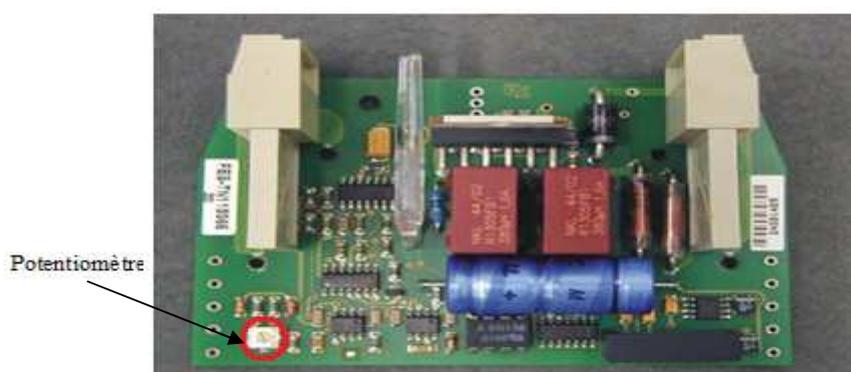


Figure A.20 : Circuit de bord du contrôleur du moteur de la pompe.

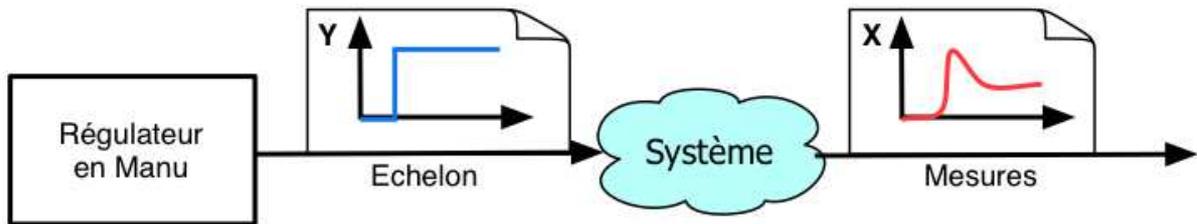
Les caractéristiques techniques du contrôleur sont résumées dans le tableau suivant :

paramètres	Valeurs
Tension opérationnelle permmissible	24VDC
Entrée	-10V...+10V
Sortie	-24V...+24V
Courant de sortie	1mA max
Connexion	Serrage a vis

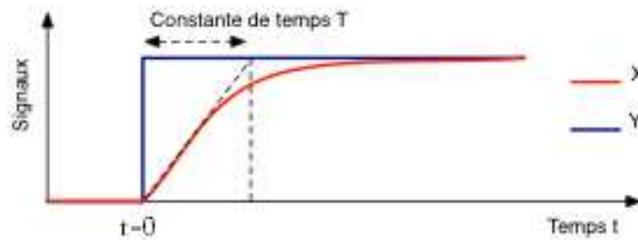
Tableau A.21 : Caractéristiques techniques du contrôleur du moteur de la pompe.

Identification d'un système du premier ordre

Cette méthode d'indentification est une méthode graphique basée sur la réponse en boucle ouverte du système à un échelon d'amplitude (Y_m) comme la montre la figure ci-dessous :



La réponse d'un système du premier ordre à la forme suivante :



Et la fonction de transfert de ce système est la suivante :

$$F(s) = \frac{K}{(1 + Ts)} \quad \text{avec: } K = \frac{X_m}{Y_m}$$

Synthèse d'un régulateur PI par la méthode d'Evans

Le but de cette méthode est la régulation d'un système dont la fonction de transfert à un gain ' K_0 ' et ceci en synthétisant un régulateur de type PI dont la fonction de transfert a la forme suivante : $R(s) = K_c \frac{(s-Z_i)}{s}$

Le calcul des paramètres de ce régulateur se fait dans le but de satisfaire les contraintes suivantes :

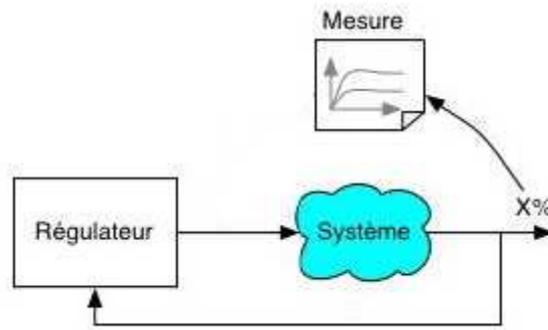
- ✓ Le système doit être stable
- ✓ Réponse indicielle apériodique ou oscillatoire optimale ($\xi=0.5$ ou $\xi=0.7$)
- ✓ Temps de montée (T_{rm})

La procédure à suivre est la suivant :

- Placement des pôles et des zéros dans le diagramme des pôles
- Tracé de la demi-droite $\psi = 30^\circ$ (apériodique) ou $\psi = 45^\circ$ (oscillatoire optimale)
- Tracé de la verticale : $\delta = -\frac{3}{T_{rm}}$
- Détermination du point d'intersection S, le pôle dominant
- Calcul de l'angle α_i du zéro Z_i à partir de la condition des angles
- Calcul du gain optimal K_{oc} à partir de la condition des modules
- Détermination des paramètres du régulateur :
$$\begin{cases} K_p = K_c = \frac{K_{oc}}{K_0} \\ T_i = -\frac{1}{Z_i K_c} \end{cases}$$

Synthèse de contrôleurs PID – Méthode de Ziegler et Nichols

Cette méthode est basée sur l'analyse de la stabilité du système à contrôler. Elle consiste à analyser le système en présence d'un contrôleur *proportionnel pur*, et à déterminer le gain de ce contrôleur qui met le système à la limite de stabilité, en assurant une oscillation entretenue. On note le gain de ce contrôleur (gain critique K_{cr}) et la période de l'oscillation (T_0). Remarquons le parallèle avec l'analyse de Bode en boucle ouverte : la fréquence critique correspond à un déphasage de -180° , et la stabilité à cette fréquence exige que le gain de la boucle ouverte G_{OL} soit unitaire. La présence d'un contrôleur purement proportionnel ne change pas le déphasage, mais son gain vient multiplier celui des autres sous systèmes de la boucle.



Essai en limite de pompage (boucle fermée)

On calcule les paramètres de réglage du contrôleur à partir des corrélations de Ziegler-Nichols rappelés dans le tableau suivant :

Type de contrôleur		T_I	T_d
Proportionnel	$\frac{K_{cr}}{2}$	-	-
P + intégral	$\frac{K_{cr}}{2.2}$	$\frac{T_0}{1.2}$	-
P I D	$\frac{K_{cr}}{1.7}$	$\frac{T_0}{2}$	$\frac{T_0}{8}$

Tableau récapitulatif du calcul des paramètres du régulateur par la méthode de Ziegler-Nichols

Annexe E

Le programme de la simulation du circuit RC avec STEP7:

OB35 : "Cyclic Interrupt"

Commentaire :

Réseau 1: Titre :

Commentaire :

```
L    "i"                MW0          -- le compteur d'incrementation
L    "n"                MW2          -- le nombre d'itérations
<I
SPBN fin
```

Réseau 2: Titre :

Commentaire :

```
//calcul de k1
L    2.000000e+001
L    "vi"                MD16        -- la valeur de la tension à l'in
-R
L    1.000000e+003
*R
L    "h"                MD20        -- le pas de discretisation
*R
T    "k1"                MD24
```

Réseau 3: Titre :

Commentaire :

```
//calcul de k2
L    2.000000e+001
L    "vi"                MD16        -- la valeur de la tension à l'instant i
-R
L    "k1"                MD24
-R
L    1.000000e+003
*R
L    "h"                MD20        -- le pas de discretisation
*R
T    "k2"                MD28
```

Annexe E

Réseau 4 : Titre :

Commentaire :

```
//calcul e v(i+1)
L    "k1"                MD24
L    "k2"                MD28
+R
L    5.000000e-001
+R
L    "vi"                MD16    -- la valeur de la tension à l'instant i
+R
T    "vi"                MD16    -- la valeur de la tension à l'instant i
```

Réseau 5 : Titre :

Commentaire :

```
//actualisation du compteur
L    "i"                MW0     -- le compteur d'incrementation
L    1
+I
T    "i"                MW0     -- le compteur d'incrementation
fin: NOP 0
```

OB100 : "Complete Restart"

Commentaire :

Réseau 1 : Titre :

Commentaire :

```
L    0
T    "i"                MW0     -- le compteur d'incrementation
//le pas h
L    1.000000e-003
T    "h"                MD20    -- le pas de discretisation
//initialisation
L    0.000000e+000
T    "vi"                MD16    -- la valeur de la tension à l'instant i
L    3.000000e-002
T    "tf"                MD8     -- temps final
L    0.000000e+000
T    "t0"                MD4     -- temps initial
//calcul du nombre d'itérations
-R
L    "h"                MD20    -- le pas de discretisation
/R
RND
T    "n"                MW2     -- le nombre d'itérations
```

Annexe F

Les programmes de l'application « Régulation de débit, niveau, pression et température à base d'automate programmables Siemens S7-313C »

1. Le Bloc d'organisation OB1 :

OB1 : "Main Program Sweep (Cycle)"

Commentaire :

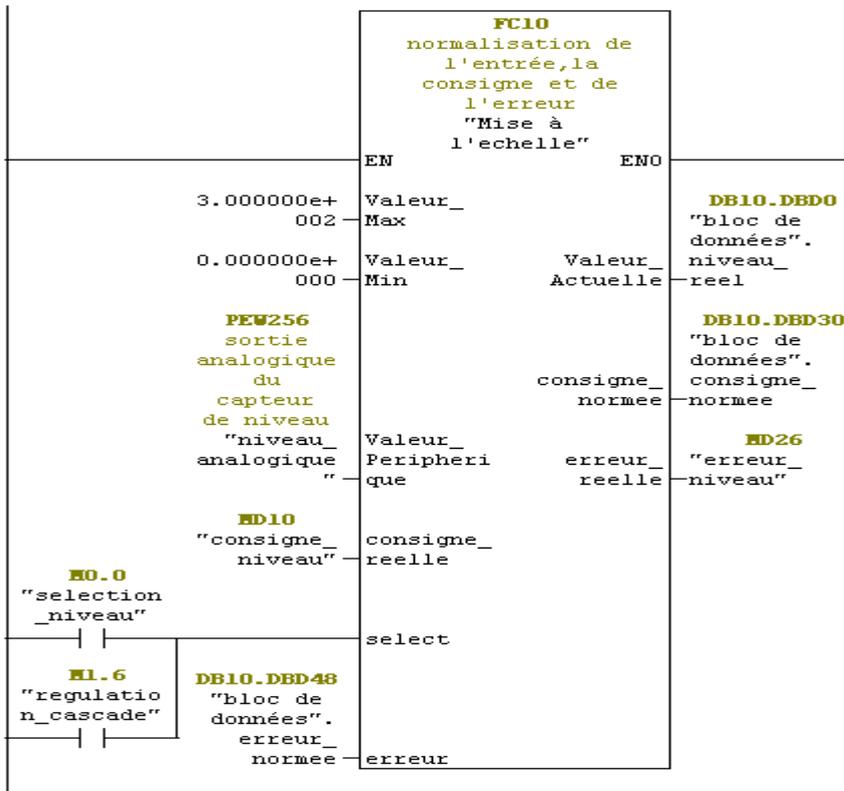
Réseau 1: appel du bloc données DB10

Commentaire :



Réseau 2: mise à l'échelle du niveau

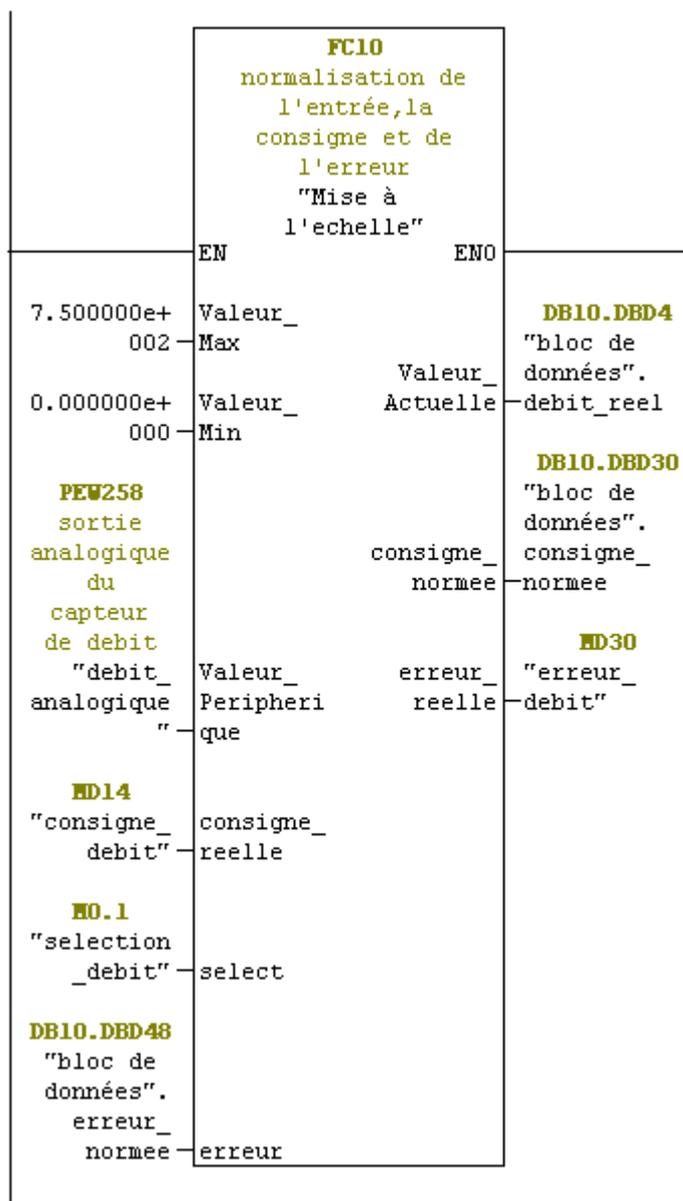
Commentaire :



Annexe F

Réseau 3 : mise à l'échelle du débit

Commentaire :



Réseau 4 : mise à l'échelle de la température

Commentaire :

```

L   "temperature_analogique"           PEW262           -- sortie analogique
ITD
DTR
L   2.764800e+004
/R
L   1.000000e+002
*R
T   "bloc de données".temperature_reelle  DB10.DBD12
    
```


Annexe F

La fonction de mise à l'échelle FC10 :

FC10 : Mise à l'échelle de l'entrée peripherique

Commentaire :

Réseau 1: mise à l'échelle de l'entrée

Commentaire :

```
L   #Valeur_Max
L   #Valeur_Min
-R
T   #Templ
L   #Valeur_Peripherique
ITD
DTR
L   #Templ
*R
L   2.764800e+004
/R
L   #Valeur_Min
+R
T   #Valeur_Actuelle
```

Réseau 2 : normalisation de la consigne

Commentaire :

```
U   #select
SPEN etql
L   #Valeur_Max
L   #Valeur_Min
-R
T   #Templ
L   1.000000e+002
L   #Templ
/R
T   #Temp2
L   #consigne_reelle
L   #Valeur_Min
-R
L   #Temp2
*R
T   #consigne_normee
etql: NOP 0
```

Annexe F

Réseau 3 : dimensionnement de l'erreur

Commentaire :

```
U      "selection_niveau"      MO.0
SPBN  etq2
L      #erreur
L      3.000000e+000
*R
T      #erreur_reelle
SPA   etq5
etq2: U      "selection_debit"    MO.1
SPBN  etq3
L      #erreur
L      7.500000e+000
*R
T      #erreur_reelle
SPA   etq5
etq3: U      "selection_temperature" MO.3
SPBN  etq4
L      #erreur
T      #erreur_reelle
SPA   etq5
etq4: U      "selection_pression"  MO.2
L      #erreur
L      4.000000e+000
*R
T      #erreur_reelle
etq5: NOP  0
```

Annexe F

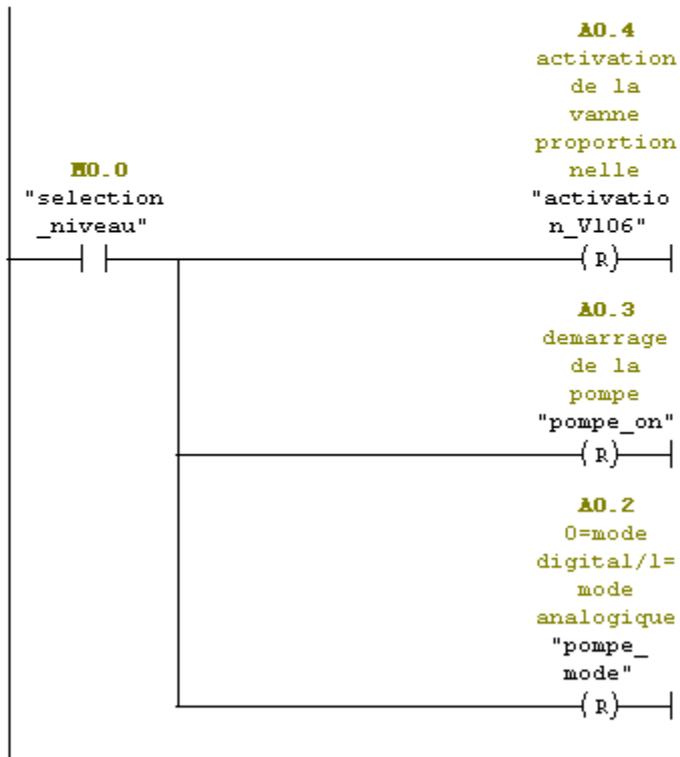
La fonction de choix de régulation FC11 :

FC11 : choix du type de regulation

Commentaire :

Réseau 1: activation de la regulation niveau

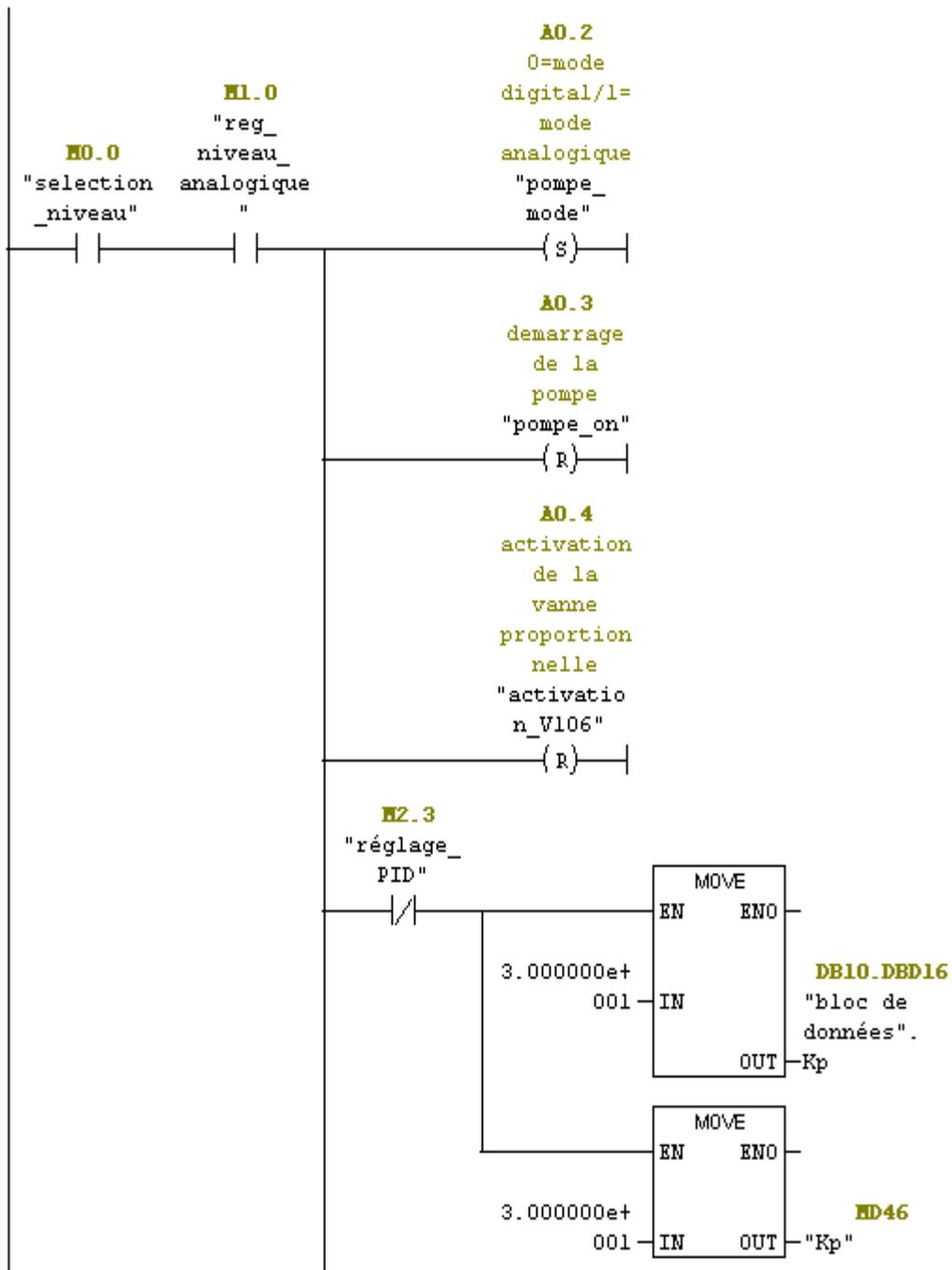
Commentaire :



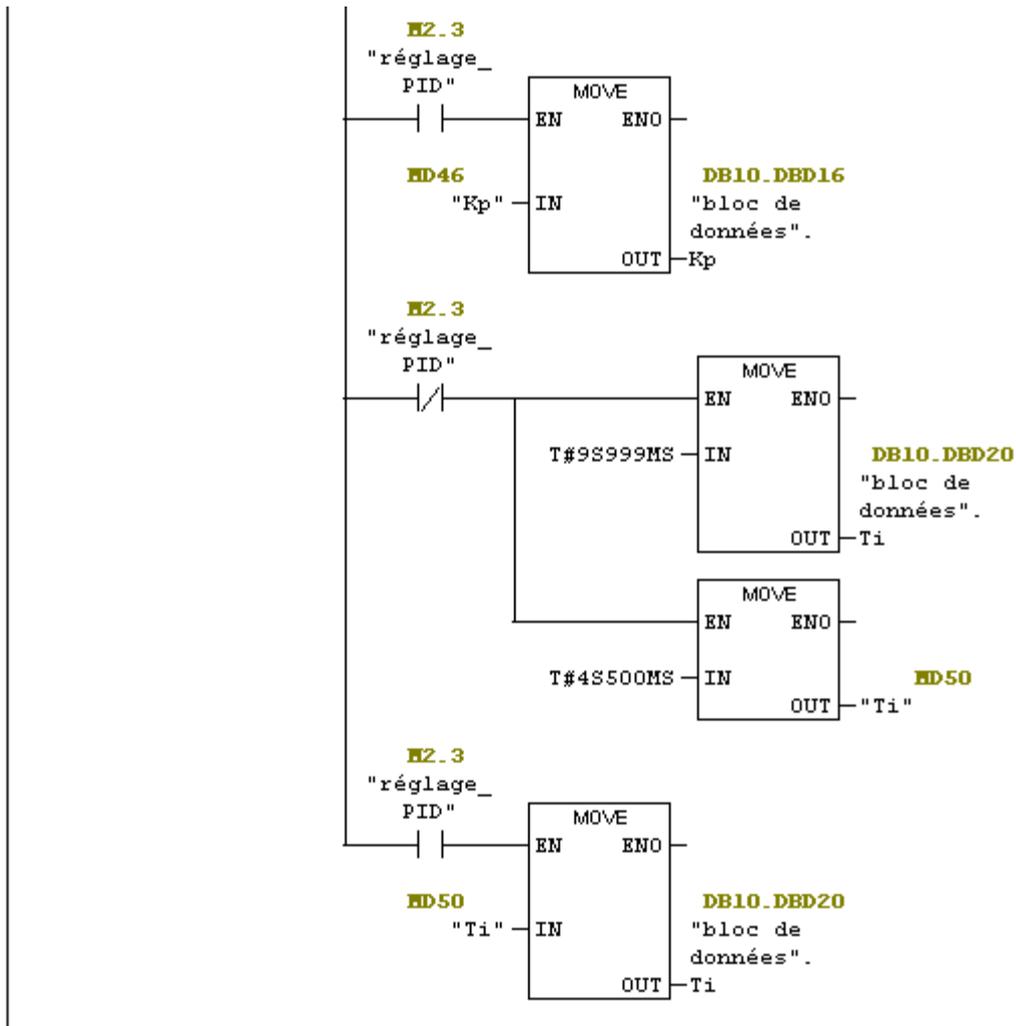
Réseau 2 : selection de la regulation du niveau mode analogique

Commentaire :

Annexe F

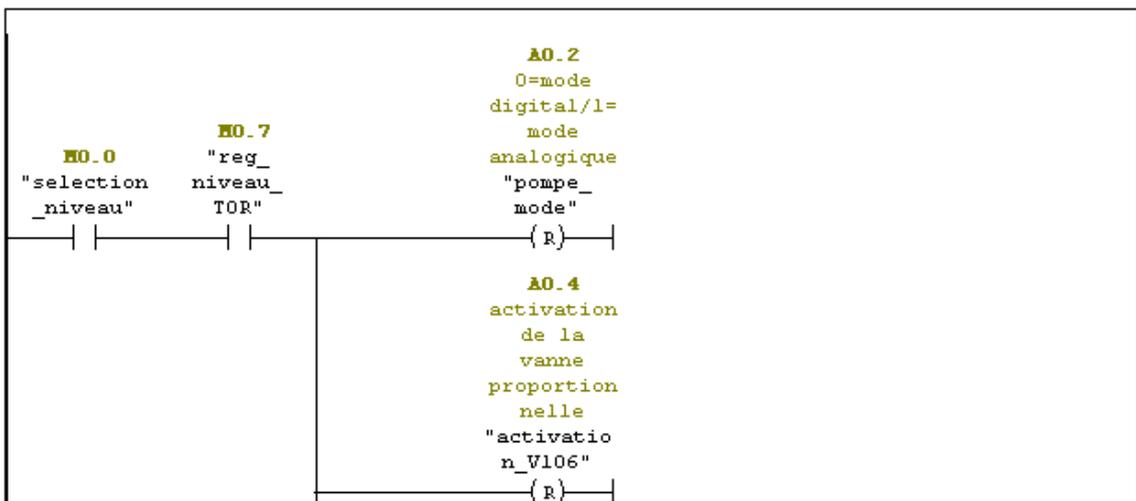


Annexe F

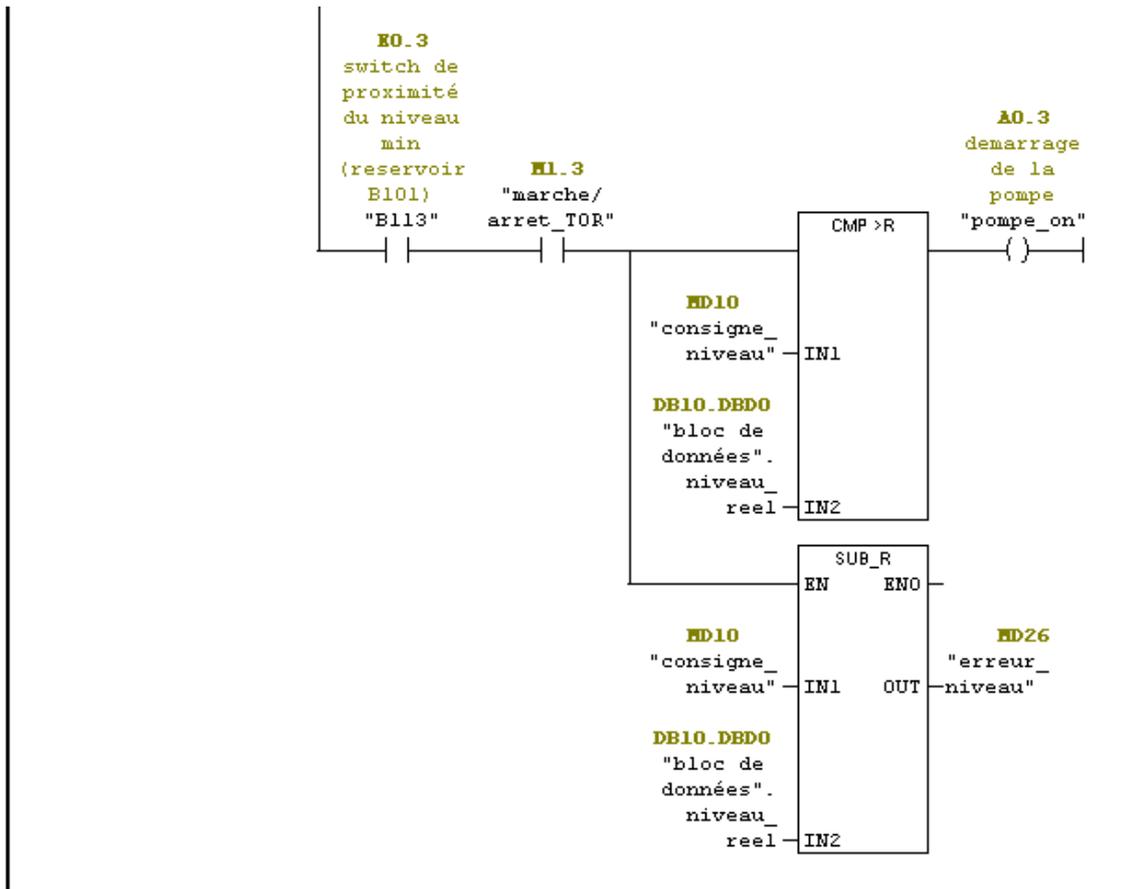


Réseau 3: selection de la regulation de niveau TOR

Commentaire :

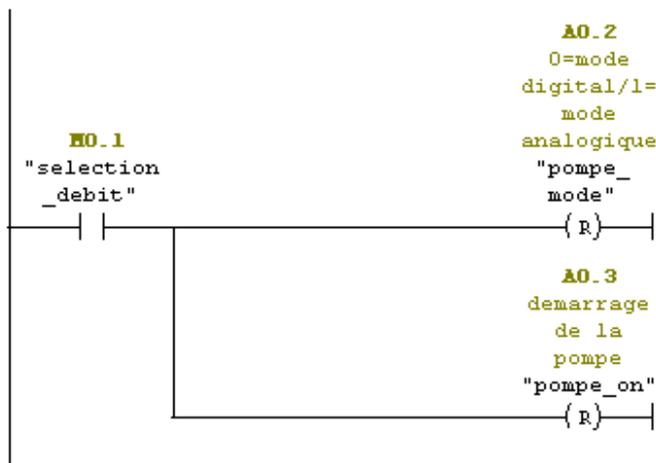


Annexe F



Réseau 4 : selection de la regulation de débit

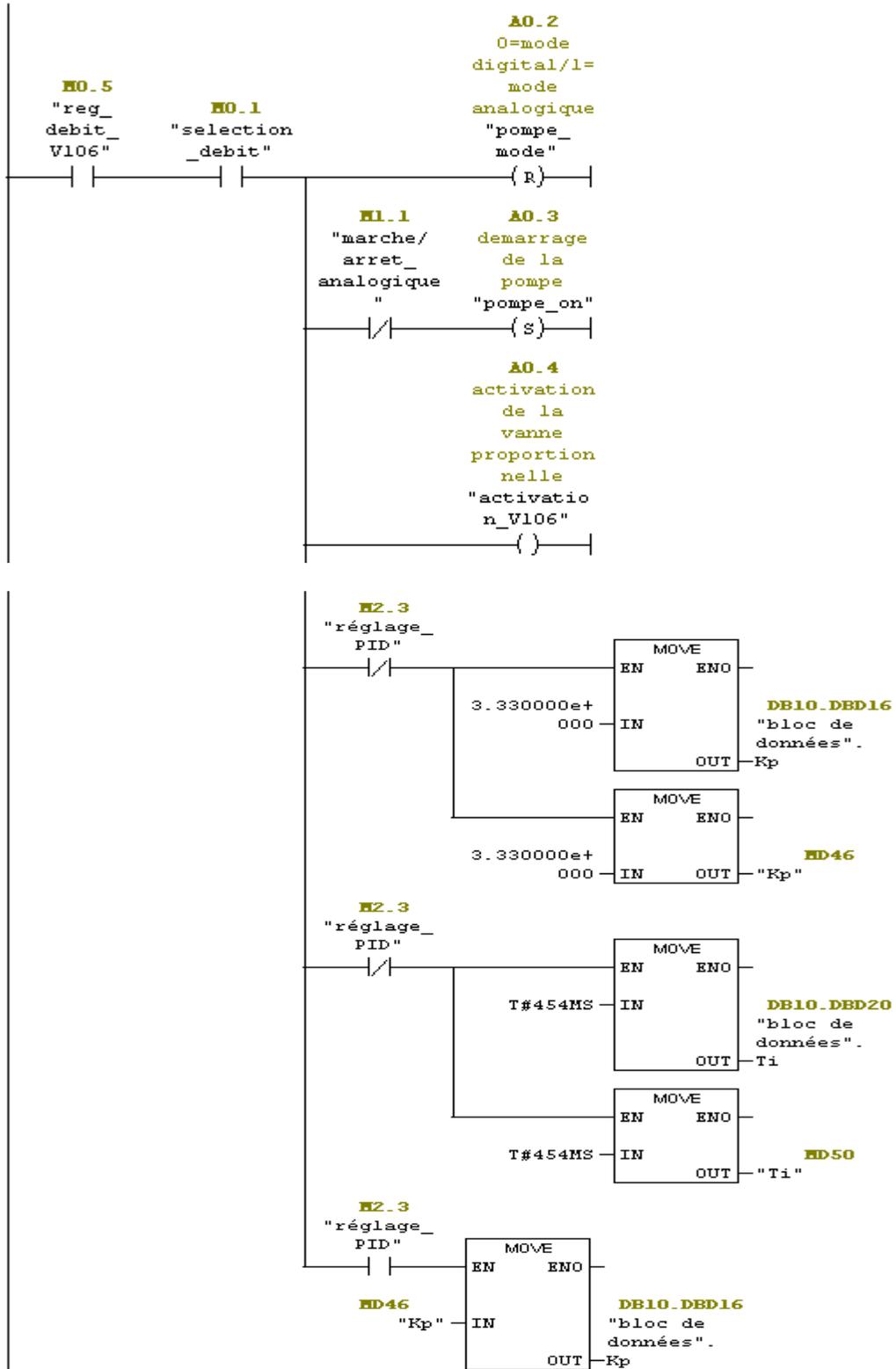
Commentaire :



Annexe F

Réseau 5 : selection du type de regulation de debit: vanne V106

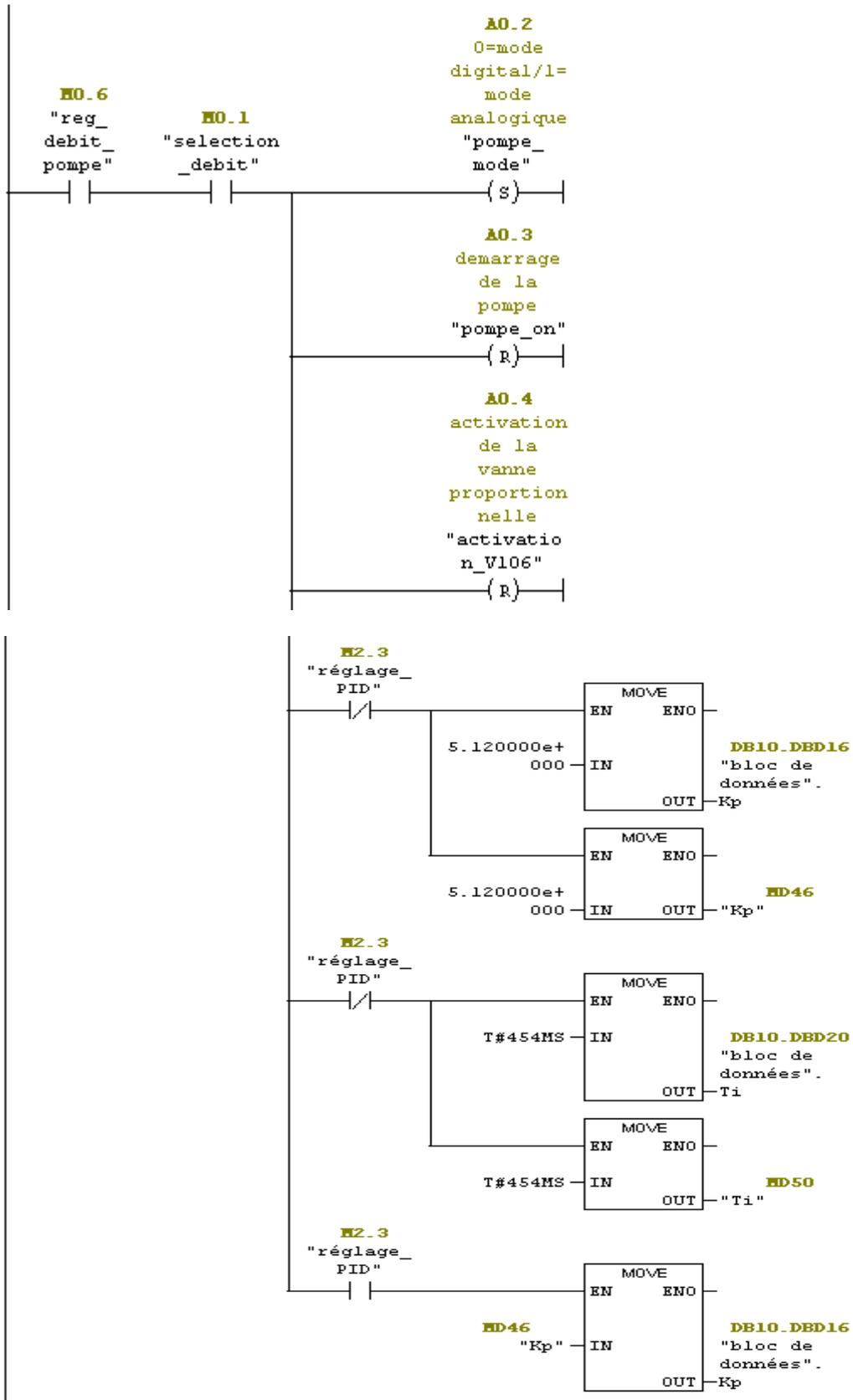
Commentaire :



Annexe F

Réseau 6 : selection du type de regulation de debit: pompe P101

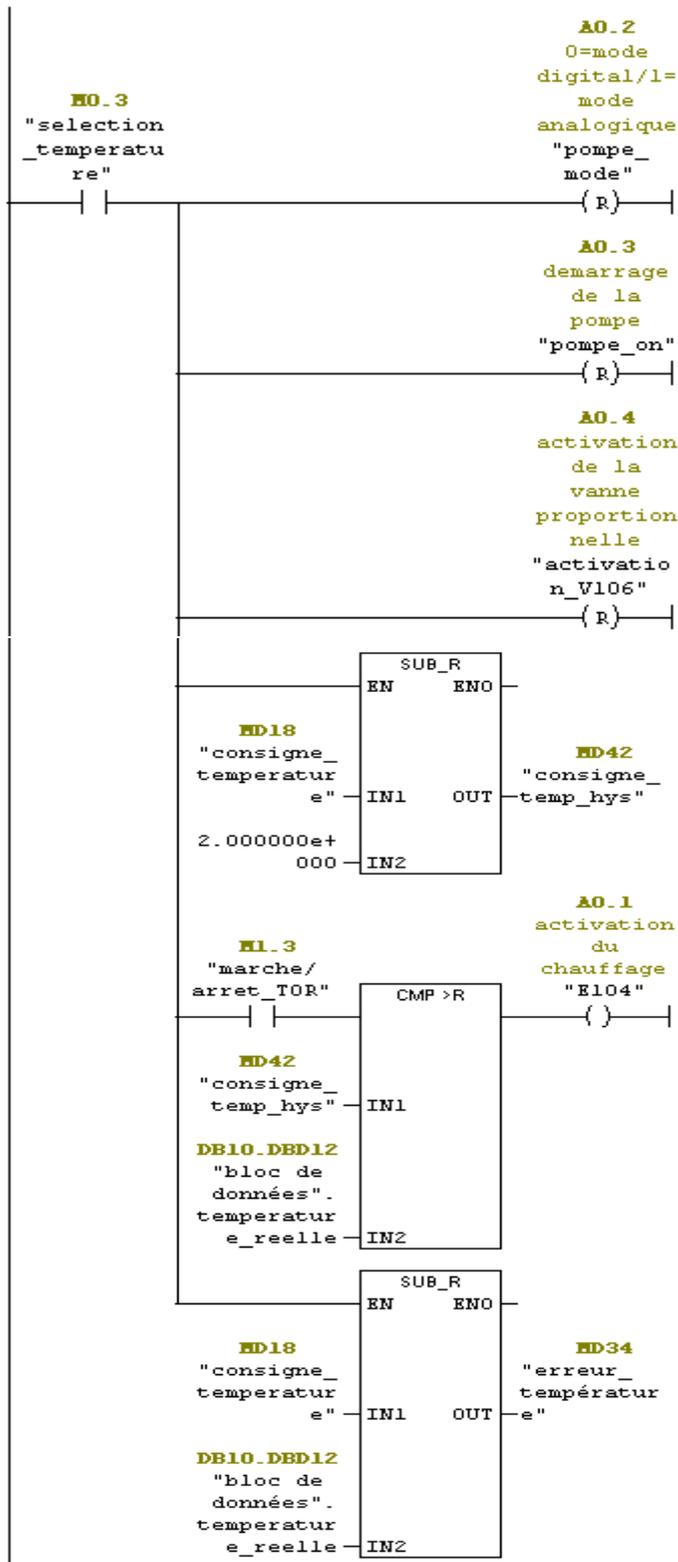
Commentaire :



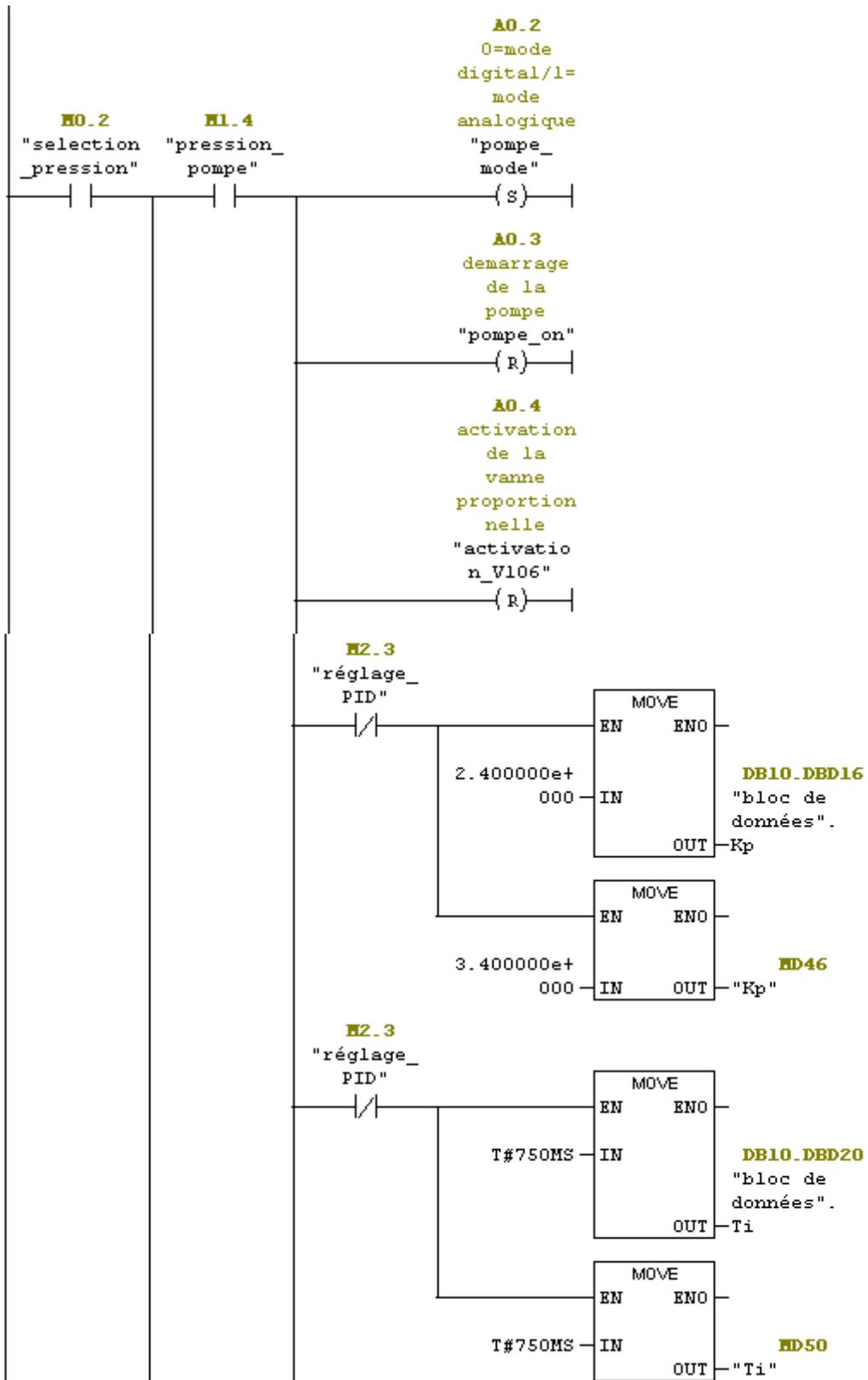
Annexe F

Réseau 7 : selection de la regulation de temperature TOR

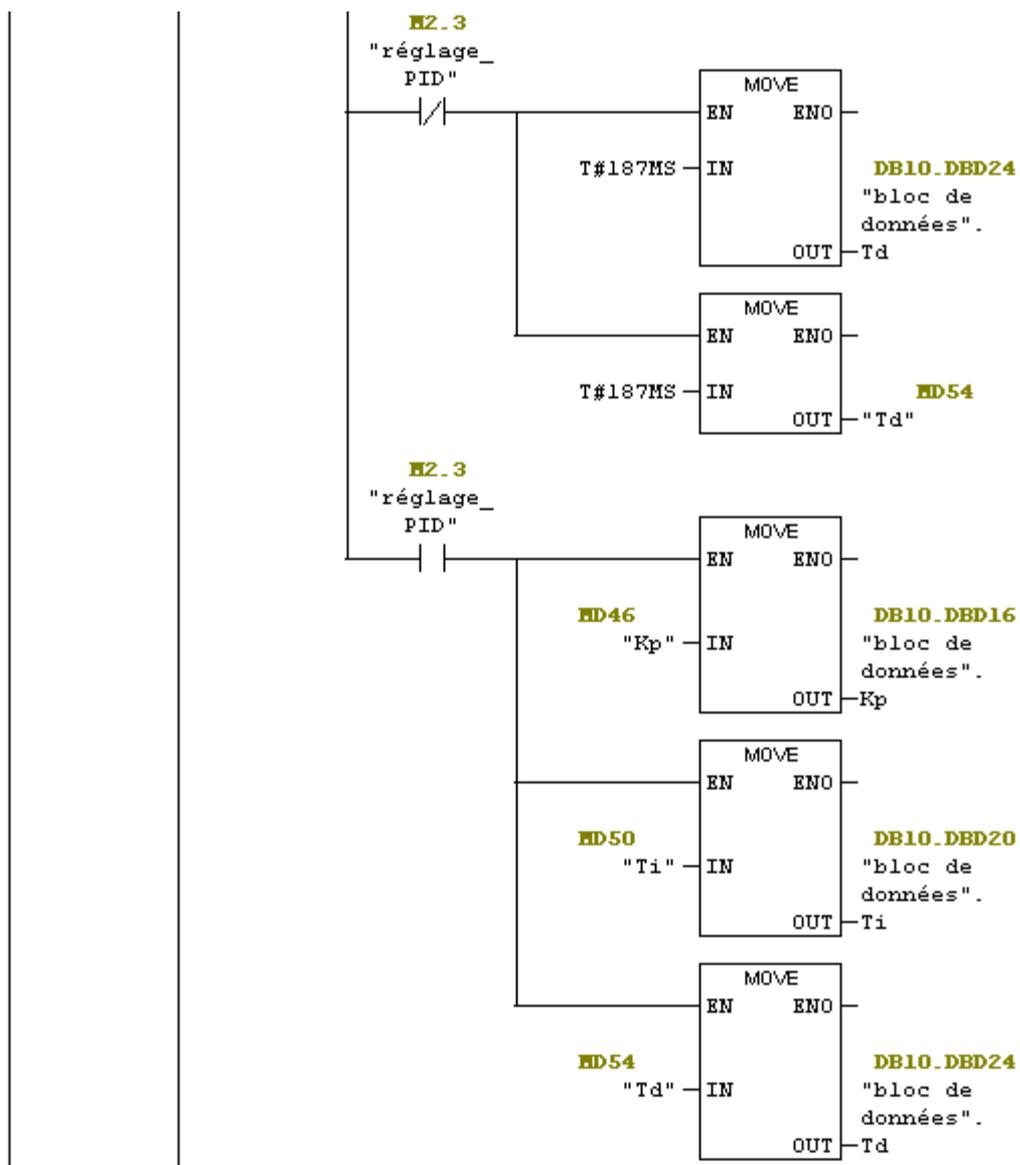
Commentaire :



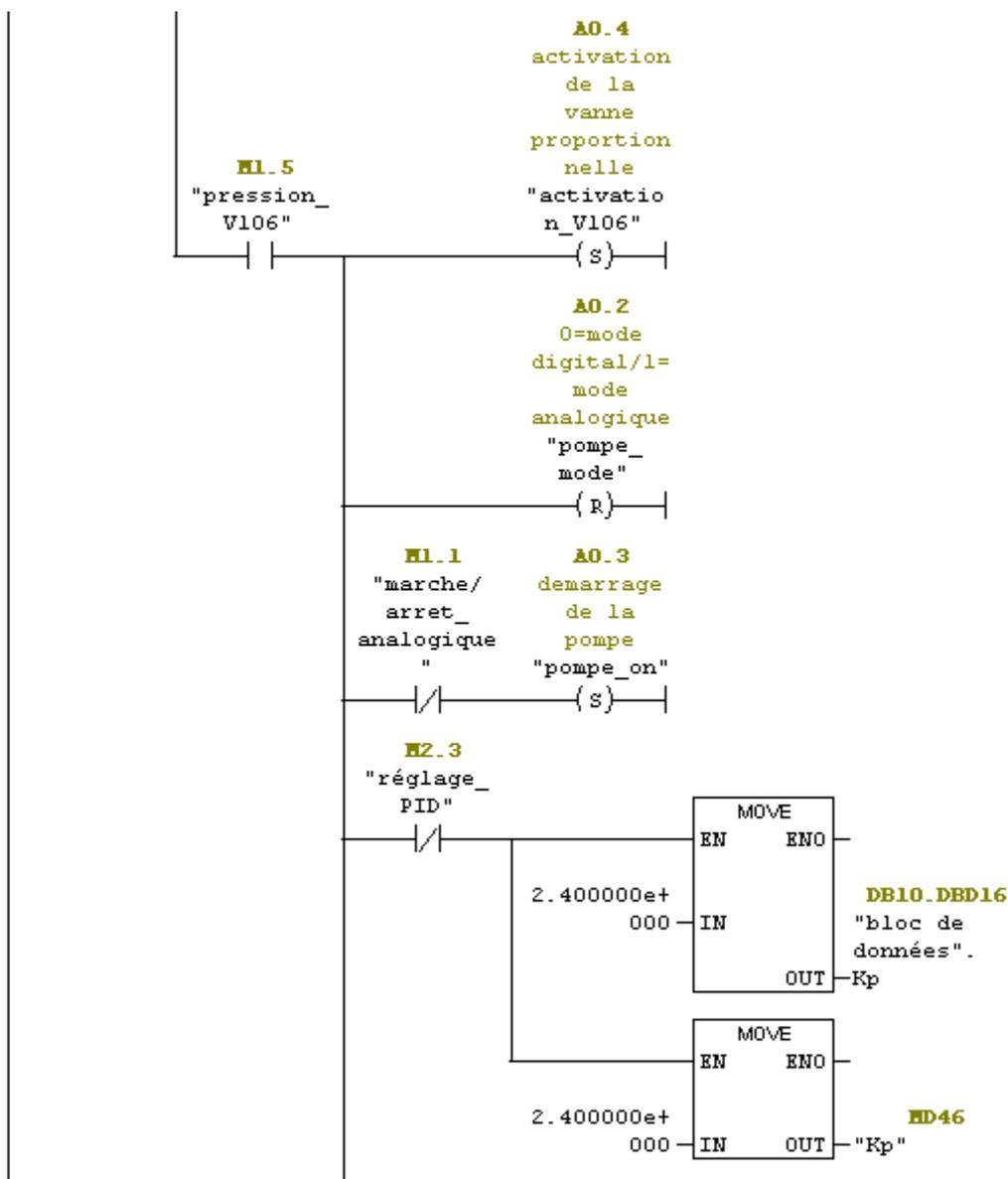
Annexe F



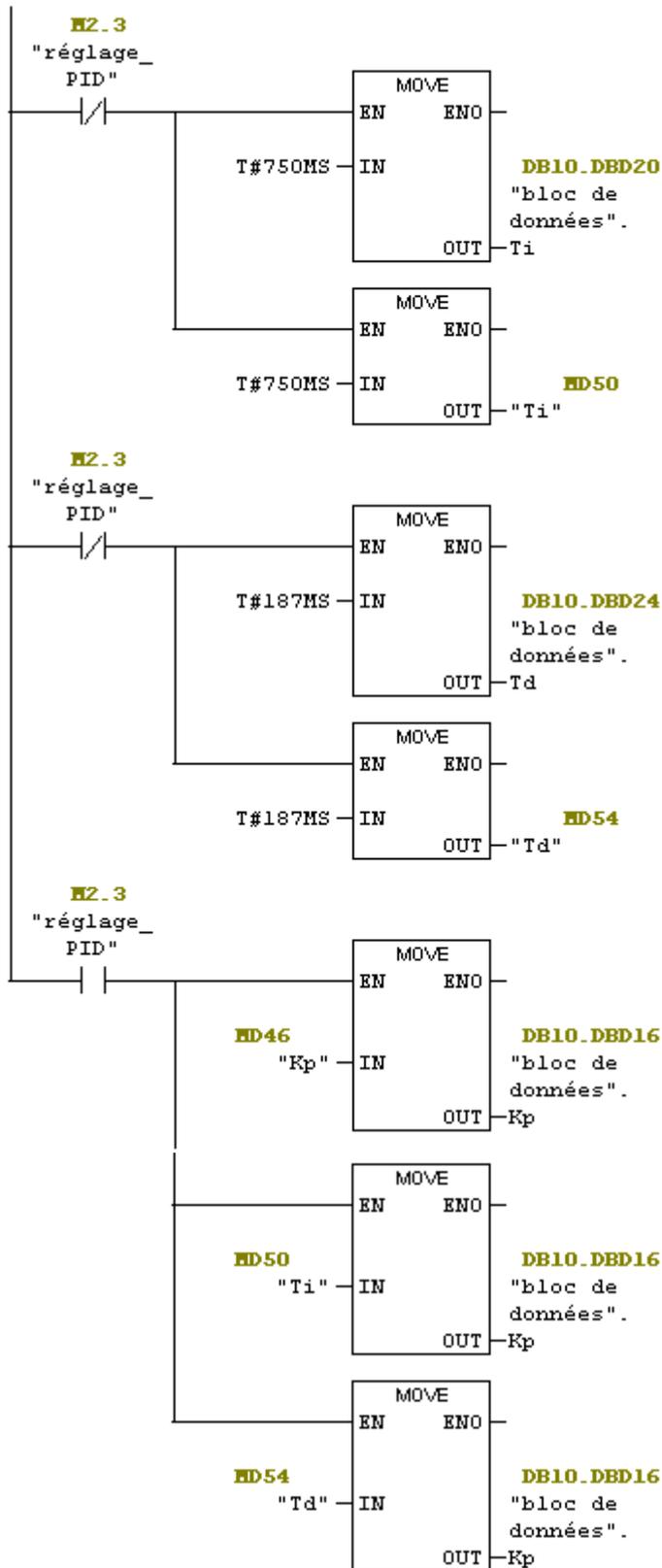
Annexe F



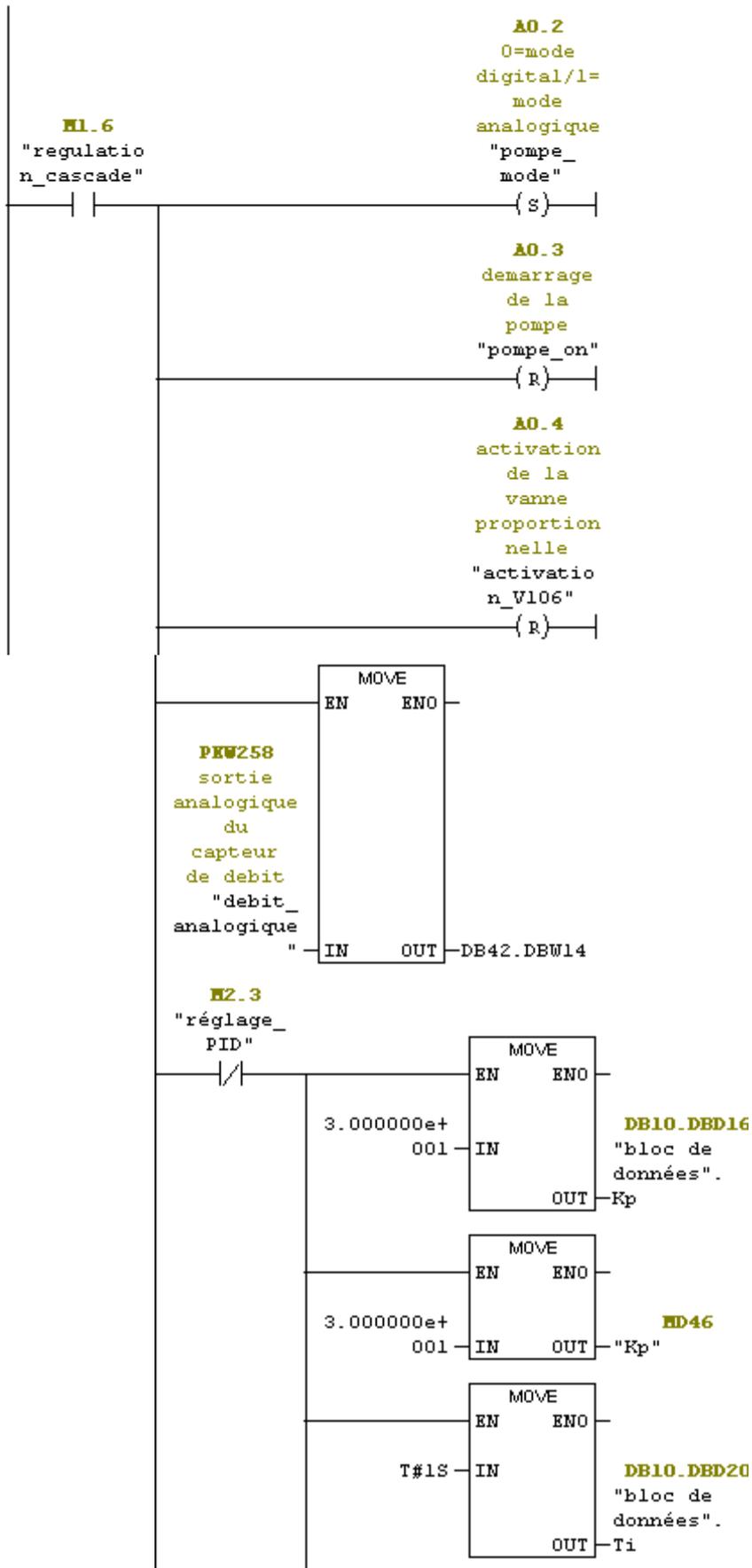
Annexe F



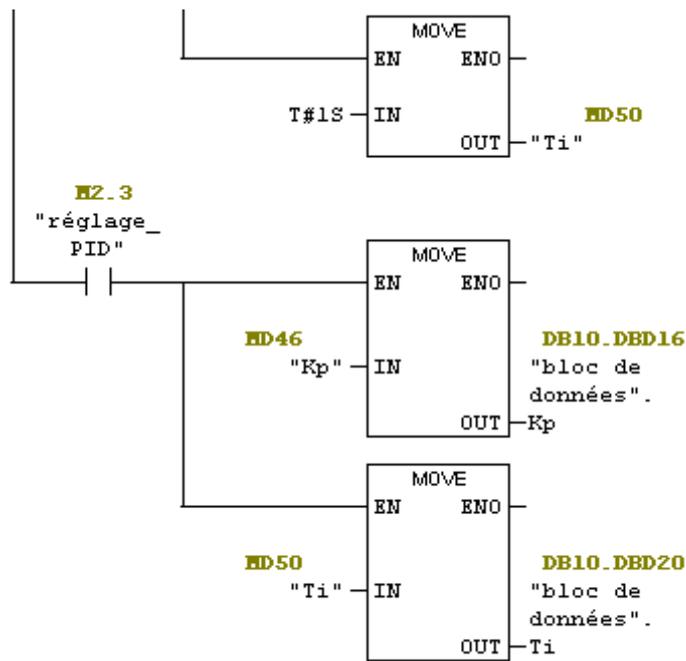
Annexe F



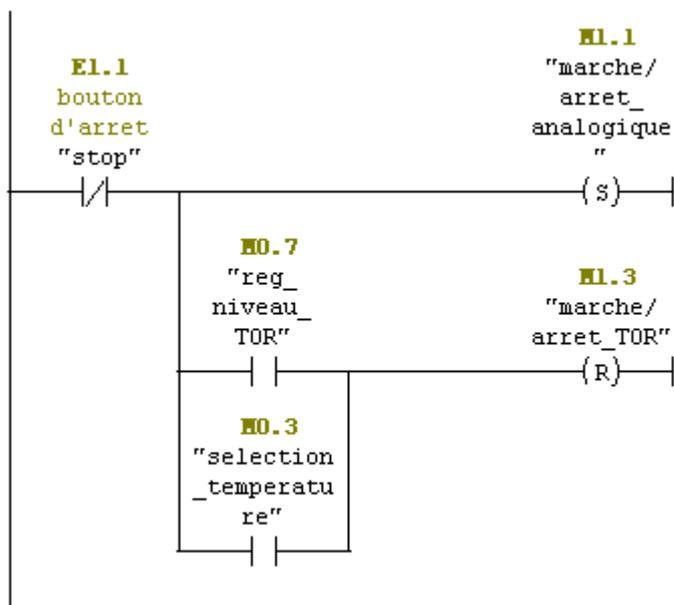
Annexe F



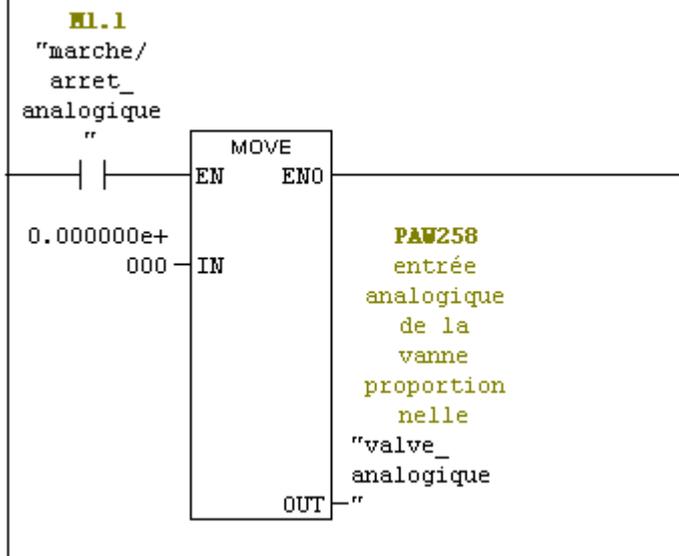
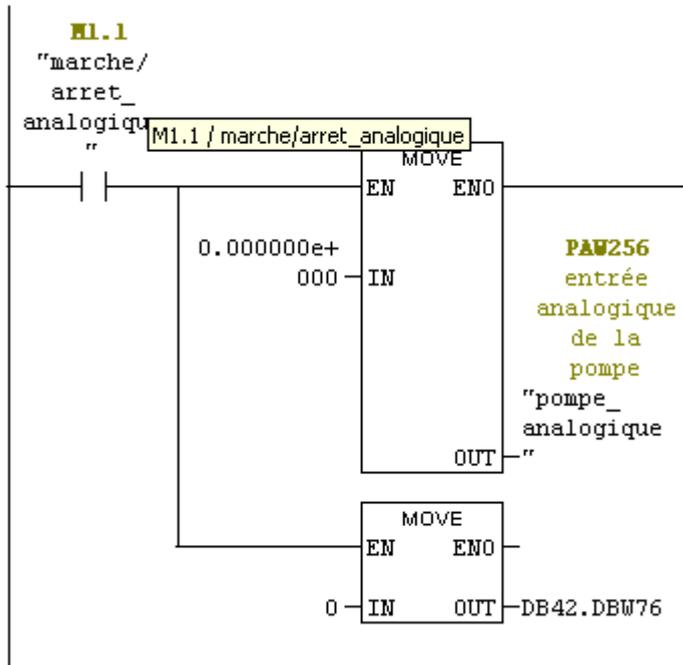
Annexe F



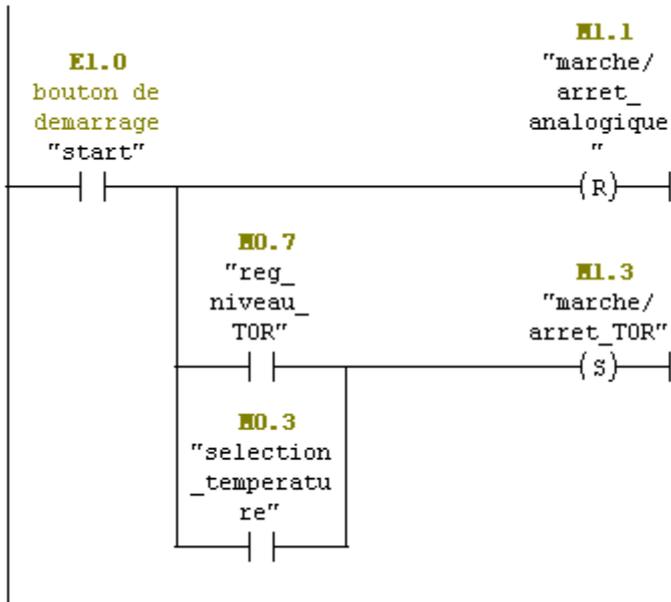
Le bloc fonction FB12 :



Annexe F



Annexe F

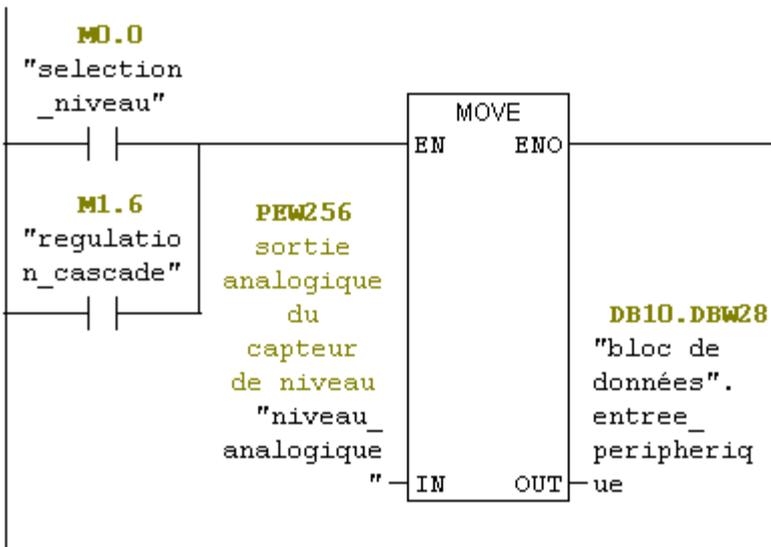


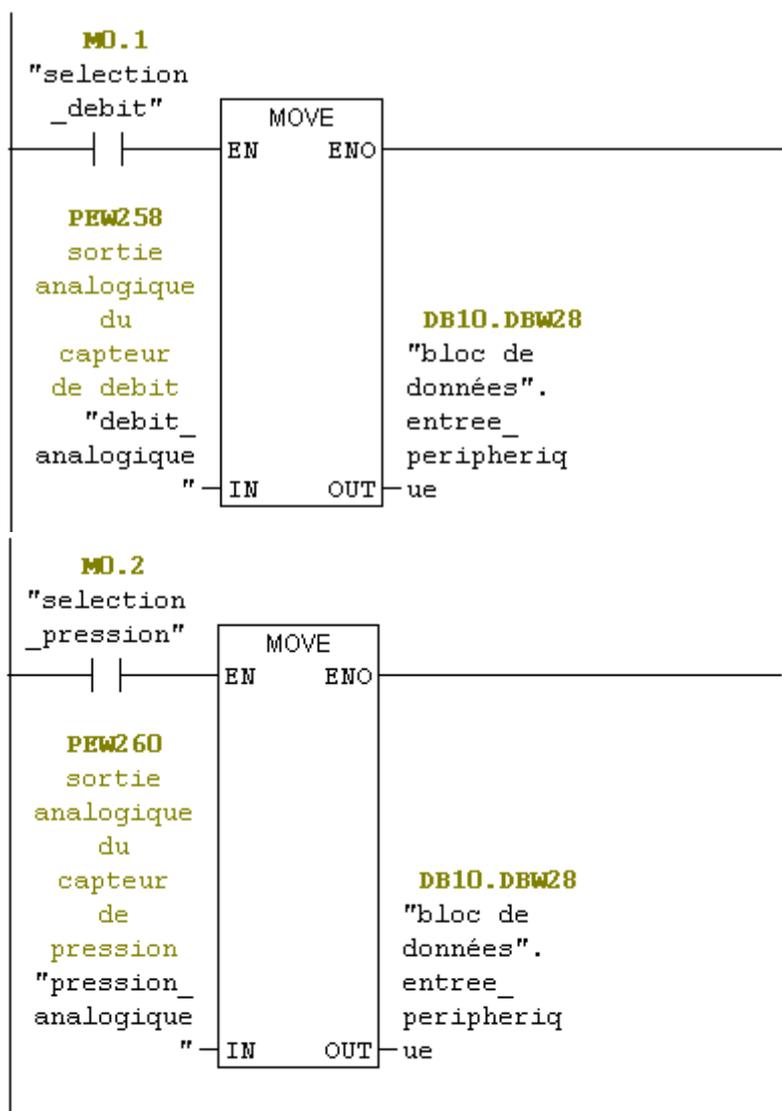
Le bloc d'organisation OB35 :



Réseau 2 : transfert de donnée peripherique niveau

Commentaire :

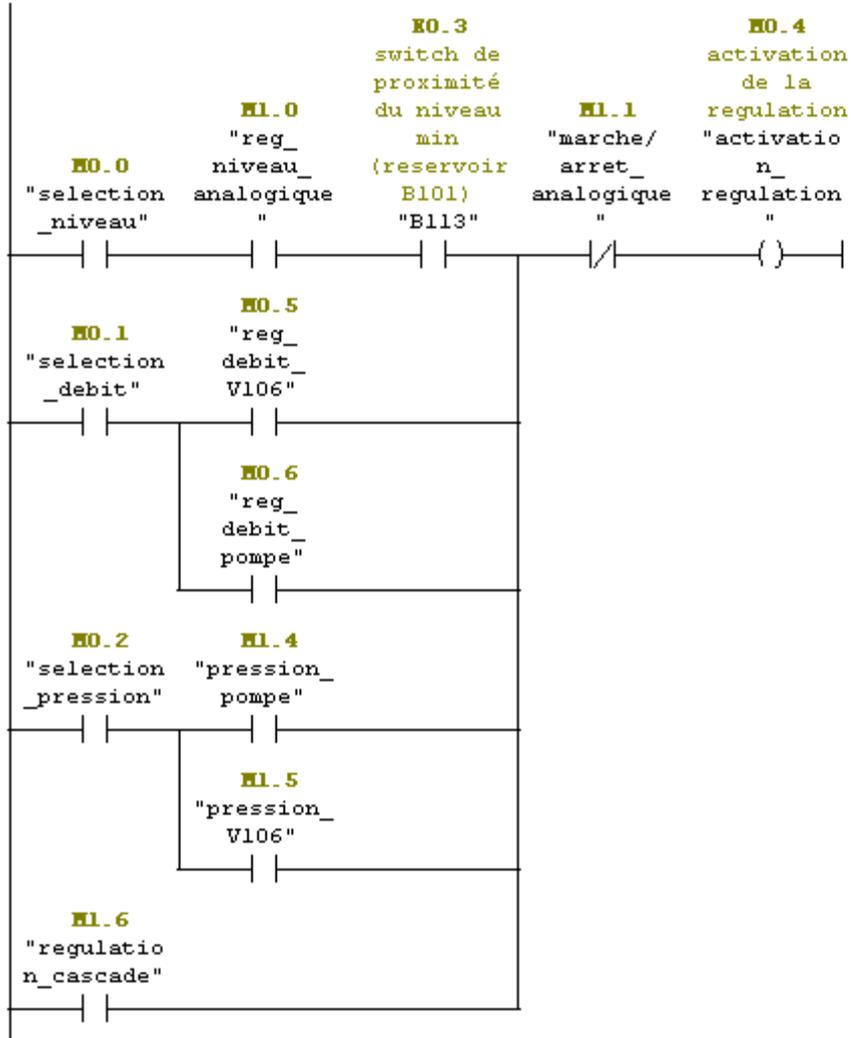




Annexe F

Réseau 5 : activation de lecture peripherique

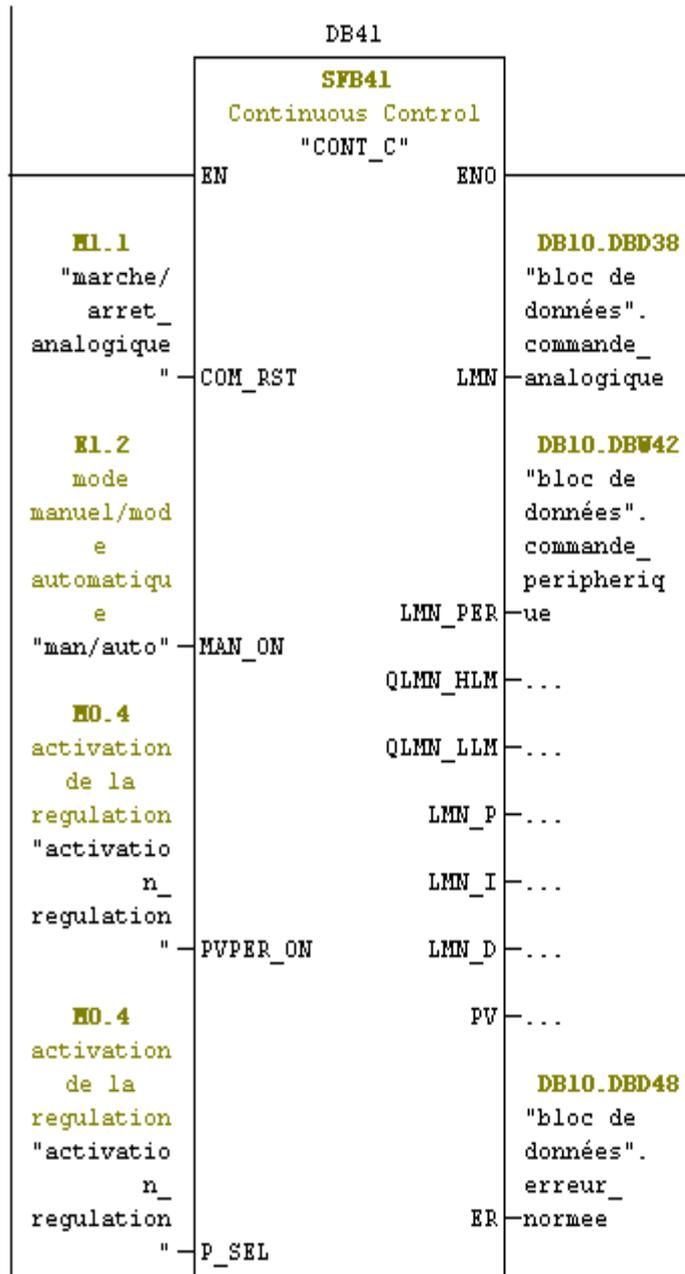
Commentaire :



Annexe F

Réseau 6 : appel du bloc de regulation PID

Commentaire :



Annexe F

HO.4 activation de la regulation "activation n_ regulation " I_SEL
... INT_HOLD
... I_ITL_ON
HO.2 "selection _pression" D_SEL
T#100MS CYCLE
DB10.DBD30 "bloc de donnees". consigne_ normee SP_INT
... PV_IN
DB10.DBW28 "bloc de donnees". entree_ peripheriq ue PV_PER
DB10.DBD44 "bloc de donnees". valeur_ reglage MAN
DB10.DBD16 "bloc de donnees". Kp GAIN
DB10.DBD20 "bloc de donnees". Ti TI
DB10.DBD24 "bloc de donnees". Td TD
... TM_LAG
... DEADE_W
... LMN_HLM
... LMN_LLM
... PV_FAC
... PV_OFF
... LMN_FAC
... LMN_OFF
... I_ITLVAL
... DISV

Annexe F

Réseau 7: bloc PID de régulation en cascade

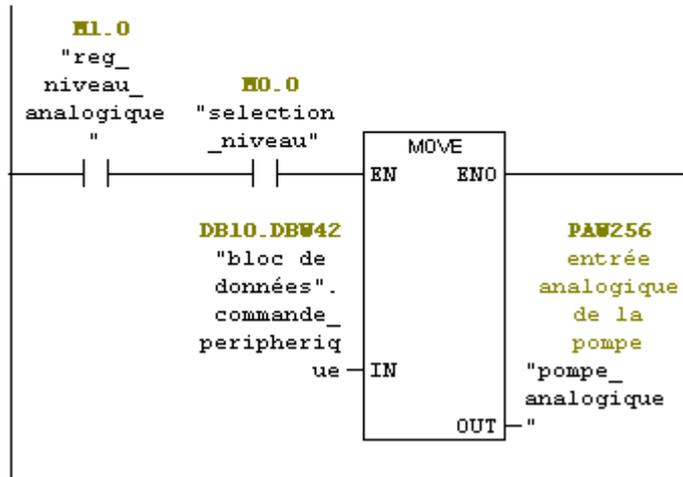
Commentaire :

```
U      "man/auto"           E1.2
=      L      20.1
BLD    103
U      "regulation_cascade" M1.6
=      L      20.2
BLD    103
U      "regulation_cascade" M1.6
=      L      20.3
BLD    103
U      "regulation_cascade" M1.6
=      L      20.4
BLD    103
UN     "marche/arret_analogique" M1.1
U      "regulation_cascade" M1.6
SPBNB _004
CALL  "CONT_C" , DB42      SFB41
COM_RST :=
MAN_ON  :=L20.1
PVPER_ON:=L20.2
P_SEL   :=L20.3
I_SEL   :=L20.4
INT_HOLD:=
I_ITL_ON:=
D_SEL   :=
CYCLE   :=T#100MS
SP_INT  :="bloc de données".commande_analogique DB10.DBD38
PV_IN   :=
PV_PER  :=
MAN     :=
GAIN    :=3.330000e+000
TI      :=T#454MS
TD      :=
TM_LAG  :=
DEADB_W :=
LMN_HLM :=
LMN_LLM :=
PV_FAC  :=
PV_OFF  :=
LMN_FAC :=
LMN_OFF :=
I_ITLVAL:=
DISV    :=
LMN     :=
LMN_PER :=
QLMN_HLM:=
QLMN_LLM:=
LMN_P   :=
LMN_I   :=
LMN_D   :=
PV      :=
ER      :=
004: NOP  0
```

Annexe F

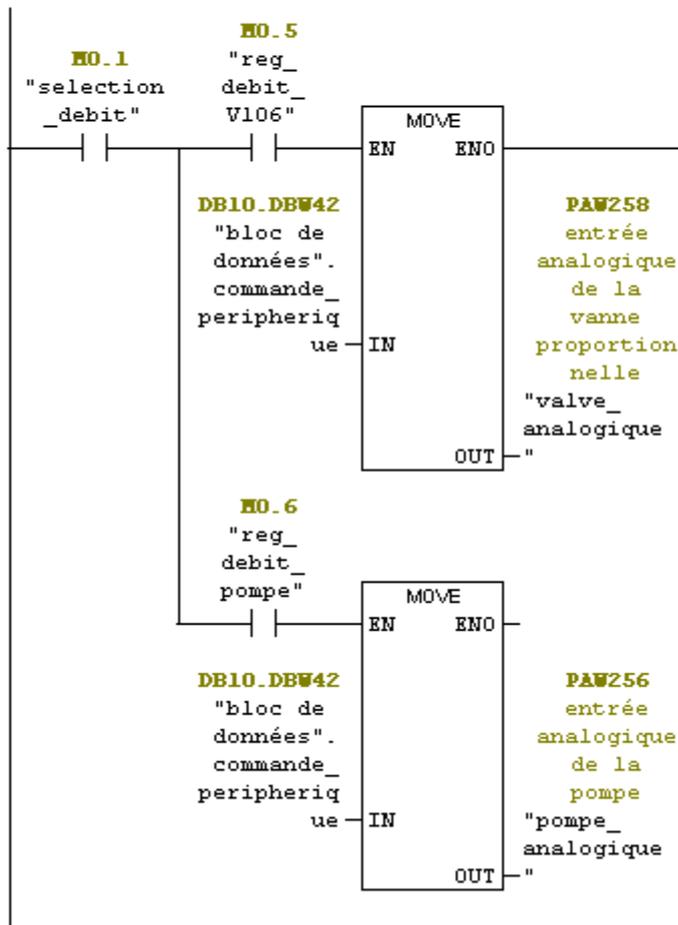
Réseau 8 : transmission de la commande peripherique niveau

Commentaire :



Réseau 9 : transmission de la commande peripherique debit

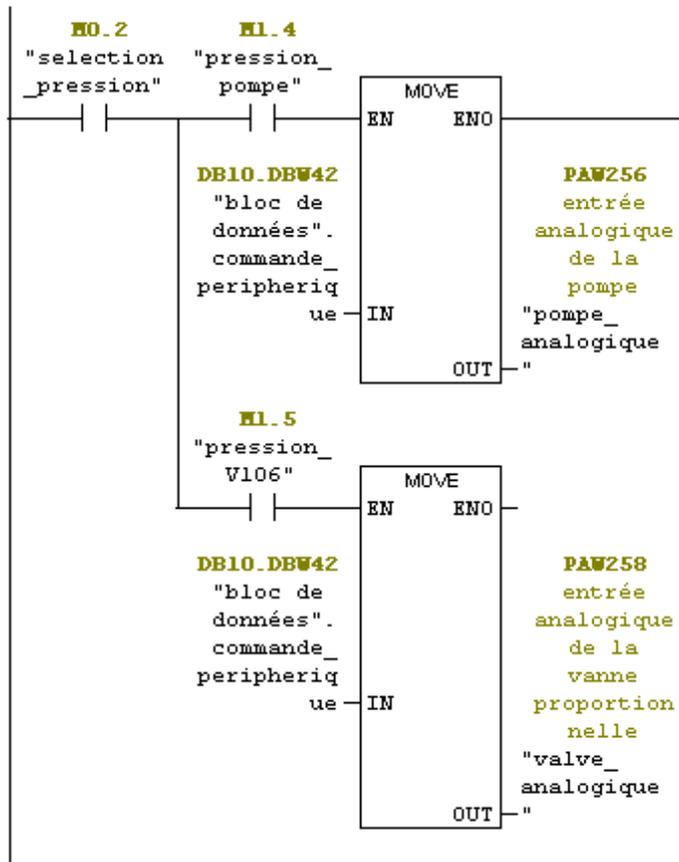
Commentaire :



Annexe F

Réseau 10 : transmission de la commande périphérique

Commentaire :



Réseau 11 : Titre :

Commentaire :

