

République Algérienne Démocratique et Populaire  
Ministère de L'enseignement Supérieur et de la Recherche Scientifique  
**Université A/Mira de Béjaia**  
Faculté de technologie  
**Département électronique**  
**Option : Automatique**



# Mémoire de Fin d'études

En vue de l'obtention du diplôme Master recherche en Automatique

## THÈME

---

Conception d'une Régulation de Niveau avec un  
Automate Programmable

---

**Réalisé par :**

M<sup>r</sup> TENSAOUT Azouaou.  
M<sup>r</sup> YUCEF KHODJA Tarik.

**Encadreur :**

M<sup>me</sup> BELLAHSENE Nora  
M<sup>r</sup> MAHMOUDI Nadir.

PROMOTION 2014/2015

# REMERCIEMENTS

*Nous remercions Dieu, le tout puissant de nous avoir accordé santé, volonté, courage et patience qui nous ont été utiles tout le long de notre parcours.*

*Nous tenons à remercier chaleureux notre promotrice à l'Université A. Mira, Madame Bellahsene Nora de nous avoir encadré et pour ses conseils, son suivi et son orientation.*

*Nous tenons également à remercier vivement notre encadreur d'entreprise ; Monsieur Mahmoudi Nadir pour nous avoir permis d'user de son précieux temps et de nous avoir encouragé tout au long de ce projet, et prodigué ses directives précieuses et ses conseils.*

*Enfin nous tenons à remercier tout le personnel du service raffinerie de CEVITAL en particulier monsieur Harfi pour leur serviabilité et leur accueil chaleureux et pour l'excellente ambiance qui a régné durant toute la durée de notre stage.*

Dédicace

À mes très chers parents  
Que Dieu les garde  
A toute ma famille et mes amis  
A tous ceux qui sont proches de mon cœur et dont je n'ai pas cité les noms  
A mon binôme azouaou  
À mon pays  
Je dédie ce modeste travail.

**Tarik**

Je dédie ce travail :

A la lumière de mon chemin, mes parents qui m'ont donné la vie, le symbole de tendresse et l'école de mon enfance, qui ont veillé tout au long de ma vie à m'encourager à me donner de l'aide et à me protéger durant toutes les années d'études. Que dieu les gardes et les protège.

A mes frères et mes sœurs.

Et toute ma famille.

Au étoiles de ma vie : mes amis.

A tous ceux qui me sont chères.

A tous ce qui m'aiment.

A tous ce que j'aime.

A toute la promotion Automatique 2015.

A tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste travail.

*TENSAOUT Azouaou*

# Table des matières

<b>INTRODUCTION GÉNÉRALE</b>	<b>1</b>
<b>1 Description du processus de raffinage d'huile</b>	<b>2</b>
1.1 Introduction . . . . .	2
1.2 Description du processus de raffinage d'huile . . . . .	2
1.2.1 Définition du raffinage . . . . .	2
1.3 Processus de raffinage au niveau du complexe CEVITAL de Bejaia . . . . .	3
1.3.1 Présentation de la raffinerie . . . . .	3
1.3.2 Les différentes étapes . . . . .	3
1.4 Description du système actuel . . . . .	6
1.5 Cahier de charges . . . . .	7
1.5.1 Description du problème . . . . .	7
1.5.2 La solution proposée au problème . . . . .	7
1.5.3 Pour que cette pompe puisse démarrer : (condition démarrage de la pompe)	7
1.5.4 Pour que la pompe s'arrête :(arrêt de la pompe) . . . . .	7
1.6 Description des équipements du système existant . . . . .	8
1.6.1 La pompe . . . . .	8
1.6.2 La vanne manuelle . . . . .	9
1.6.3 Réservoir d'eau . . . . .	9
1.6.4 Les capteurs . . . . .	10
1.6.5 Description des capteurs du système proposé . . . . .	10
1.7 Conclusion . . . . .	12
<b>2 Généralités sur la Régulation</b>	<b>13</b>
2.1 Introduction . . . . .	13
2.2 Définition de la régulation automatique . . . . .	13
2.3 Objectifs de la régulation . . . . .	14
2.4 Notion de Boucle Ouverte/Fermée . . . . .	15
2.4.1 Système boucle ouverte . . . . .	15
2.4.2 Système boucle fermée . . . . .	15
2.5 Types de régulation automatique . . . . .	16
2.5.1 Régulation tout ou rien (TOR) . . . . .	16
2.5.2 Régulation analogique . . . . .	16
2.5.3 Régulation numérique . . . . .	16
2.6 Critères de performance d'une régulation . . . . .	16
2.7 Définition de la régulation PID . . . . .	17
2.8 Notion de correcteur PID . . . . .	17
2.8.1 Structure des régulateurs électroniques . . . . .	18
2.8.2 Le FB 41 CONT-C . . . . .	20
2.8.3 Utilisation . . . . .	20

2.8.4	Description . . . . .	20
2.8.5	États de fonctionnement . . . . .	20
2.8.6	Information d'erreur . . . . .	20
2.9	Conclusion . . . . .	21
<b>3</b>	<b>LES AUTOMATES PROGRAMMABLES</b>	<b>22</b>
3.1	Systèmes automatisés . . . . .	22
3.1.1	Objectif de l'automatisation : . . . . .	23
3.1.2	Structure d'un système automatisé : . . . . .	23
3.2	Les avantages et les inconvénients d'un système automatisé . . . . .	25
3.2.1	Les avantages : . . . . .	25
3.2.2	Les inconvénients : . . . . .	25
3.3	les automates programmables : . . . . .	25
3.4	Architecture des automates . . . . .	26
3.4.1	Aspect extérieur : . . . . .	26
3.4.2	Structure interne : . . . . .	27
3.5	Présentation de quelques gammes SIMATIC . . . . .	29
3.5.1	SIMATIC S7 : . . . . .	29
3.5.2	SIMATIC M7 : . . . . .	31
3.6	Critères de choix d'un automate programmable . . . . .	31
3.6.1	Quantification et choix d'automate . . . . .	32
3.7	Présentation de logiciel de programmation STEP7 . . . . .	32
3.7.1	Création d'un projet sous STEP7 . . . . .	33
3.8	Présentation de WinCC Flexible . . . . .	34
3.8.1	Introduction : . . . . .	34
3.8.2	Les tâches d'un système IHM : . . . . .	34
3.8.3	SIMATIC WinCC flexible . . . . .	34
3.8.4	Éléments de WinCC flexible : . . . . .	35
3.8.5	Intégration de WinCC flexible à STEP7 : . . . . .	36
3.9	Conclusion . . . . .	37
<b>4</b>	<b>Simulation et Résultats</b>	<b>38</b>
4.1	Introduction . . . . .	38
4.2	L'OB1 . . . . .	38
4.3	Les blocs de données d'instance (DB) . . . . .	39
4.3.1	Création d'un DB . . . . .	39
4.3.2	Contenu d'un DB . . . . .	39
4.4	LES FCS . . . . .	40
4.4.1	Création d'une FC . . . . .	40
4.5	Réalisation d'un PID sous STEP7 avec le bloc de régulation continue FB41 CONT-C	41
4.5.1	Régulateur PID, données du problème . . . . .	41
4.6	Table des mnémoniques . . . . .	43
4.7	Programme . . . . .	44
4.8	Les alarmes . . . . .	45
4.9	Simulation . . . . .	51
4.9.1	Simulation du programme sous STEP7 avec PLCSIM : . . . . .	51
4.9.2	Simulation du programme sous WinCC : . . . . .	52
4.10	Conclusion . . . . .	53
	<b>CONCLUSION Générale</b>	<b>54</b>

---

ANNEXE 1	i
ANNEXE 2	iii
ANNEXE 3	iv
Bibliographie	ix

# Table des figures

1.1	Les étapes de raffinage de l'huile . . . . .	5
1.2	Les différentes structures de la création du vide . . . . .	6
1.3	la pompe centrifuge . . . . .	8
1.4	la vanne manuelle . . . . .	9
1.5	réservoir d'eau . . . . .	9
1.6	Capteur . . . . .	10
1.7	Détecteur de niveau à lames vibrantes « Liquiphant T FTL260 » . . . . .	11
1.8	Modes de fonctionnement du Liquiphant T FTL260 . . . . .	12
2.1	Présentation d'un schéma fonctionnel d'un régulateur . . . . .	14
2.2	Système en BO . . . . .	15
2.3	Système en BF . . . . .	15
2.4	PID parallèle . . . . .	18
2.5	PID mixte . . . . .	19
2.6	PID série . . . . .	19
3.1	Structure d'un système automatisé . . . . .	23
3.2	Procédé automatisé . . . . .	24
3.3	Image réelle d'API . . . . .	26
3.4	Structure interne des automates . . . . .	27
3.5	API SIMENS S7-200 . . . . .	29
3.6	API SIMENS S7-300 . . . . .	30
3.7	API SIMENS S7-400 . . . . .	30
3.8	Organigramme pour la création de projets sous STEP7 . . . . .	33
3.9	Système d'automatisation avec un pupitre . . . . .	34
3.10	Fenêtre de WinCC flexible . . . . .	35
4.1	Exemple d'interruption de bloc par rapport à l'OB1 . . . . .	38
4.2	Création d'un DB . . . . .	39
4.3	Contenu d'un DB . . . . .	39
4.4	Création d'une FC . . . . .	40
4.5	Simulateur S7-PLCSIM . . . . .	51



# Liste des tableaux

- 2.1 Notion de correcteur PID . . . . . 17
- 2.2 Tableau d'influence . . . . . 18

---

## A

**API** Automates Programmable Industriel

**AP** Automates Programmable

## B

**BF** Boucle Fermée

**BO** Boucle Ouverte

## C

**CPU** Central Processing Unit

## G

**GM** Général motors

## K

**K<sub>p</sub>** Coefficient de Gain Proportionnel

## P

**P** Proportionnel

**PI** Proportionnel Intégral

**PID** Proportionnel Intégral Dérivé

**PLC** Programmable Logic Controllers

---

**S**  
**SPA** Système programmable automatisé

**T**  
**TOR** Tout Ou Rien

$T_i$  Temps d'intégration

$T_d$  Temps dérivation

# INTRODUCTION GÉNÉRALE

---

Au cours des dernières décennies, la régulation industrielle a connu un essor vertigineux grâce à un nouveau système, baptisé l'automate.

Ainsi, avec la mondialisation des économies et l'accentuation de la concurrence, le bien fondé des vertus de ce système pousse davantage les entreprises à automatiser leurs processus de production, afin d'assurer leur pérennité en s'offrant la compétitivité tout en améliorant les conditions de travail de leur personnel.

Dans cette course de survie, en Algérie, l'entreprise CEVITAL.SPA a fait un grand pas dans ce domaine, puisque la modernisation de ses installations et le travail selon les normes internationales fondent son image, ceci lui procure des économies d'échelle, lui permet de réaliser une productivité optimale et lui offre une meilleure combinaison : prix, qualité, quantité et délai.

Néanmoins, il y a encore certaines stations, dans cette méga-entreprise, non automatisées, entre autre, la fosse de relevage de la raffinerie d'huile dont à présent, tout ce fait manuellement (le contrôle de niveau des liquides et/ou de l'huile dans les différents bacs, l'ouverture et la fermeture des vannes et des pompes, la sélection des mécanismes à actionner ...). Cette opération mobilise un nombre important d'opérateurs.

Par conséquent, les dirigeants de CEVITAL ont commencé à étudier la possibilité de son automatisation. Cette opération vise à remplacer totalement l'opérateur humain par une commande à base d'automate programmable industriel API.

Le stage pratique que nous avons effectué au sein de cette société s'inscrit dans la même optique, ainsi, l'objectif de notre travail consiste à automatiser cette partie de l'usine en mettant au point un nouveau système de fonctionnement commandé par un automate programmable industriel API, en intégrant la technique de surveillance qui est la supervision.

Pour y répondre, nous avons réparti ce travail en quatre chapitres :

- \* Le premier chapitre, présente les différentes étapes de raffinage d'huile, ensuite, nous avons donné une description du système existant en faisant ressortir la problématique, le fonctionnement et expose une analyse fonctionnelle du système proposé, ainsi que l'instrumentation nécessaire.

- \* Le deuxième chapitre est consacré à Définition de la régulation automatique.

- \* Le troisième chapitre est consacré à la présentation de nouveau système de commande (automate programmable industriel).

- \* Nous avons terminé notre mémoire en exposant des différentes étapes de programmation de nouveau système proposé à l'aide du logiciel SYMATIC manager S7-300, et un aperçu sur le logiciel utilisé (WinCC flexible) pour la supervision et étale les interfaces qu'on a pu développer pour la supervision de notre installation.

Enfin, nous terminerons par une conclusion.

# Chapitre 1

## Description du processus de raffinage d'huile

### 1.1 Introduction

Dans ce premier chapitre nous donnons en premier temps une description du complexe agro-alimentaire CEVITAL et le processus de raffinage de l'huile ainsi que ces différentes étapes. Nous évoquons le processus de la raffinerie au complexe CEVITAL de Bejaia, en particulier la méthode manuelle de récupération d'huile issue des différentes étapes de raffinage d'huile brute, dite : fosse de relevage.

### 1.2 Description du processus de raffinage d'huile

#### 1.2.1 Définition du raffinage

Le raffinage est l'ensemble des opérations qui servent à transformer l'huile brute en un produit comestible en éliminant les impuretés qui le rendent impropres à la consommation en l'état. En effet, les huiles contiennent de nombreux composés : certains sont très utiles (vitamines, insaponifiables, etc), d'autres sont nuisibles à leurs qualités (gommes, acides gras libres, pigments, agents odorants, etc). Le raffinage consiste donc à éliminer au mieux ces composés afin d'obtenir une huile aux qualités organoleptiques et chimiques satisfaisantes. Il comprend plusieurs opérations :

- **Démucilagination (ou dégomme)** : elle permet de débarrasser les huiles des gommes après leur hydrolyse par un acide.
- **Neutralisation** : les acides gras libres sont les impuretés les plus représentées dans les huiles à raffiner. L'étape de neutralisation sert à éliminer ces composés susceptibles d'accélérer l'oxydation de l'huile. Elle se fait à l'aide de la soude caustique et suivie d'un lavage à l'eau et d'un séchage.
- **Décoloration** : elle sert à éliminer les pigments contenus dans l'huile.
- **Filtration** : cette étape permet d'obtenir une huile limpide après élimination de la terre décolorante.
- **Désodorisation** : cette étape permet de débarrasser l'huile de son odeur désagréable par distillation sous vide poussé à une température élevée (180C-200C).

## 1.3 Processus de raffinage au niveau du complexe CEVITAL de Bejaia

### 1.3.1 Présentation de la raffinerie

La capacité de production de la raffinerie d'huile du complexe CEVITAL de Bejaia est de 1800 tonnes par jour, qui se fait en trois lignes de production :

- Deux lignes symétriques (A et B) de capacité de 400 tonnes/jour chacune qui constitue 800 tonnes par jour.

- Une ligne C de capacité 1000 tonnes par jour.

pour plus d'informations sur le complexe CEVITAL voir l'annexe 1.

### 1.3.2 Les différentes étapes

#### Neutralisation et démulgation

L'huile brute provenant des bateaux arrive dans des bacs de stockage puis passe ensuite par un moteur à brosse afin d'éliminer les particules physiques qui pourra prendre lieu. Elle est ensuite soutirée à l'aide d'une pompe (PU), puis elle passe dans un échangeur de chaleur pour la chauffer de 25 C à 90 C, ensuite, dans un mélangeur (M1) en lui rajoutant un dosage de l'acide citrique afin d'éliminer les phospholipides. Afin d'accélérer la réaction chimique, le mélange (Acide citrique et huile brute) est transféré vers un réacteur R1 qui permet une agitation pendant 20 minutes (réaction de démulgation), en second lieu, on ajoute la soude caustique dans le mélangeur M2 afin d'éliminer les acides gras libre (AGL).

L'huile neutralisée et démulgée passe à travers des séparateurs qui permettent la séparation de l'huile des pates de neutralisation par centrifugation à 90 C. La phase lourde (pâte) passe vers les parois des séparateurs et la phase légère (l'huile) reste au centre.

Ensuite, l'huile séparée subira un lavage à l'eau chaude et d'un dosage d'acide citrique pour éliminer le reste des savons, puis on effectue un séchage sous vide afin d'éliminer l'humidité. A la fin de la neutralisation, l'huile est envoyée dans un bac pour la décoloration.

#### Décoloration

Dans un réacteur sous vide, on ajoute la terre décolorante à l'huile neutralisée en le chauffant avec de la vapeur. Le mélange est maintenu sous agitation durant 30 minutes pour que la terre absorbe les pigments. L'étape suivante consiste à faire passer le mélange à travers des filtres dite (filtre NIAGARA) qui vont piéger la terre contenant les pigments.

#### Désodorisation

Cette opération consiste à injecter de la vapeur sèche dans l'huile décolorée maintenue sous vide à une température de 225 C à 240 C. L'huile sort de l'économiseur à une température de 125 C puis passe vers un échangeur pour atteindre 110 C, enfin l'huile désodorisée est refroidie à 35 C en passant sur un économiseur, puis sur un refroidisseur.

**Remarque**

La figure ci-dessous représente toutes les étapes du raffinage d'huile, en précisant les auxiliaires principales de fabrication ainsi que les constituants indésirables éliminés. Notons que les déchets de la neutralisation vont être envoyés à la section de traitement des déchets qui s'effectue selon deux étapes : prétraitement et saponification.

Pendant l'étape de saponification, on réchauffe le mélange qui sera transféré dans un réacteur pour décomposition pendant une heure. Ce qui entraîne la formation de plusieurs phases, phase eau-acide, phase huile-acide et une phase intermédiaire non traité, elle sera envoyée à la fosse pour recyclage. Après cette étape de neutralisation, l'huile sera lavée, et l'eau de lavage est envoyée dans des canalisations entraînant une quantité d'huile dans la fosse.

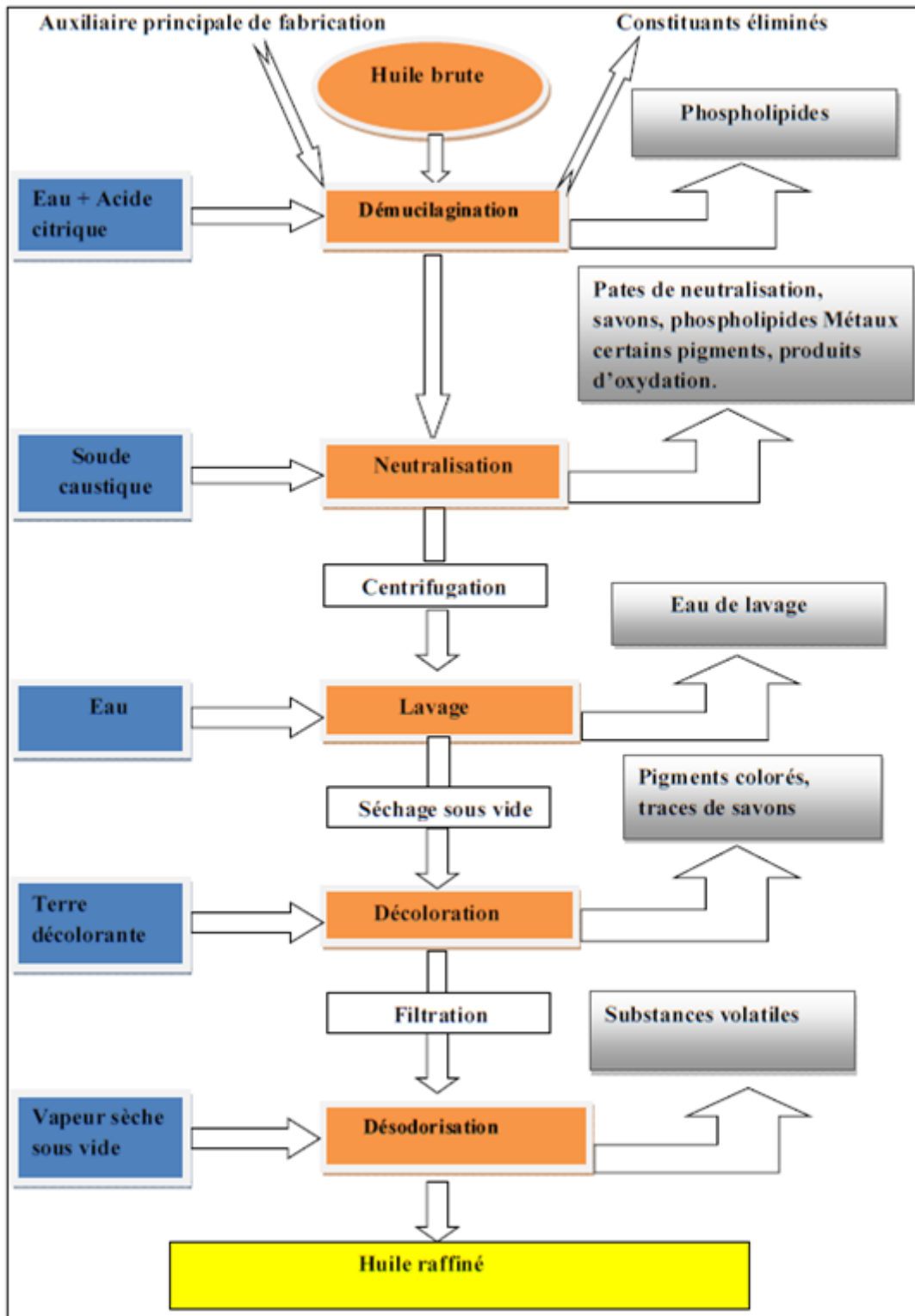


FIGURE 1.1 – Les étapes de raffinage de l'huile



## 1.4 Description du système actuel

La récupération d'huile basée sur le principe de décantation, issu des différentes étapes du raffinage d'huile brute (neutralisation et décoloration), sous forme de mélange eau-huile, ensuite, l'eau est séparée de l'huile en formant deux couches, une supérieure et une autre inférieure. La couche inférieure qui est dans le bac l'eau sale est envoyée vers deux sections via la pompe (Pompe2) dans le système de séparation (condenseur, éjecteurs) pour les deux sections :

- Section 600 pour la décoloration.
- Section 800 la désodorisation.

Le rôle des condenseurs sont la condensation de la vapeur qui existe au niveau des éjecteurs pour créer le vide pour séparer les huiles gros et les matières acides.

Ces particule grosses sont stockée ensuite dans le bac hot walle à l'air libre pour séparer l'eau d'huile.

De hot welle on envoie l'eau vers les refroidisseur pour refroidissement de l'eau avec la pompe (Pompe1), ensuite on envoie l'eau dans le bac d'eau sale.

Le schéma ci-dessous explique les différents structures qui consistent a la création du vide dans la récupération d'huile.

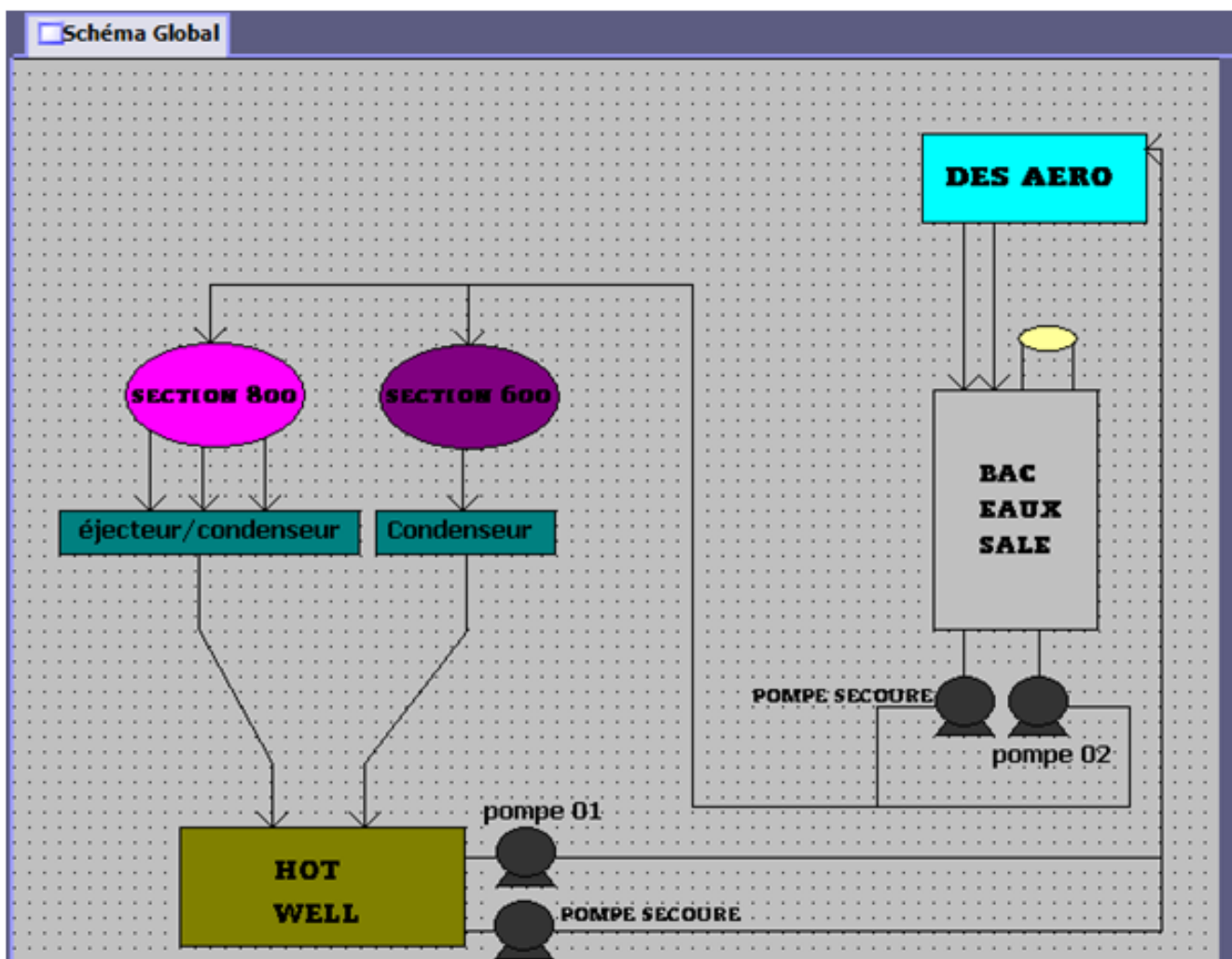


FIGURE 1.2 – Les différentes structures de la création du vide

## 1.5 Cahier de charges

- Nous avons élaboré ce cahier de charges afin de pouvoir commander cette pompe, c'est-à-dire démarrage et arrêt.
- A titre d'information, le bac est équipé de deux transmetteurs de niveau.
- Un variateur connecté à l'automate (ALTIVAR61)

### 1.5.1 Description du problème

Au départ notre système est commandé par une fonction tout ou rien, la pompe démarre à l'indication d'un niveau bas, et elle s'arrête à chaque niveau haut, ce système force le redémarrage et l'arrêt.

Par conséquent la pompe démarre et s'arrête environ 300 fois par heure ce qui fait subir des contraintes mécaniques et électriques sur la pompe et réduit sa durée de vie et entrave ainsi la production.

### 1.5.2 La solution proposée au problème

Alors pour remédier à ce problème c'est-à-dire avoir une solution qui permet le bon fonctionnement de la pompe sans augmenter le coût, nous avons opté pour une modification qui consiste à modifier la régulation tout ou rien par une régulation type proportionnel intégral dérivé PID à l'aide de l'automate Siemence.

Nous proposons une régulation à la pompe pour éviter les démarrages et les arrêts,

### 1.5.3 Pour que cette pompe puisse démarrer : (condition démarrage de la pompe)

- La pompe doit être prête (pas d'alarme).
- Le niveau du bac doit être inférieur à 80/100 du niveau maximal.
- Aucune alarme dans l'installation.
- Appui sur le boutons marche.

### 1.5.4 Pour que la pompe s'arrête :(arrêt de la pompe)

- Le niveau du bac doit être supérieur à 80/100.
- Appui sur le boutons arrêt.
- Une alarme dans l'installation.
- Défaut de niveau.

## 1.6 Description des équipements du système existant

Le système est composé des éléments suivants : bêche à eau, des vannes manuelles, deux pompes, deux capteurs de niveau.

### 1.6.1 La pompe

C'est un organe qui sert à aspirer et à refouler un fluide tel que l'eau et il est entraîné par un moteur asynchrone triphasé. Les pompes centrifuges sont des machines qui sont largement utilisées pour assurer le transfert de fluides. Ce sont les pompes les plus utilisées dans le domaine industriel à cause de leurs simplicités et faibles coûts.

#### **Principe de fonctionnement :**

Les pompes centrifuges sont composées d'une roue à aube qui tourne autour de son axe, d'un stator constitué au centre d'un distributeur qui dirige le fluide de manière adéquate à l'entrée de la roue, et d'un collecteur en forme de spirale disposé en sorte d'une roue appelée volute.

Le fluide arrivant par l'ouïe est dirigé vers la roue en rotation qui sous l'effet de la force centrifuge lui communique de l'énergie cinétique. Cette énergie cinétique est transformée en énergie de pression dans la volute.

Un diffuseur à la périphérie de la roue permet d'optimiser le flux sortant et de limiter les pertes d'énergie. La figure suivante représente la pompe utilisée dans l'installation.

pour voir le schéma fonctionnel et la plaque signalétique consulté l'annexe 2



FIGURE 1.3 – la pompe centrifuge

### 1.6.2 La vanne manuelle

C'est un dispositif qui permet l'ouverture ou la fermeture d'une conduite, soit le passage ou le blocage d'un débit et elle est commandée manuellement.



FIGURE 1.4 – la vanne manuelle

### 1.6.3 Réservoir d'eau

Ce sont des bacs en acier utilisés pour le stockage d'eau.



FIGURE 1.5 – réservoir d'eau

### 1.6.4 Les capteurs

#### Définition

Un capteur est un dispositif qui transforme une grandeur physique en une grandeur généralement électrique, qui peut être interprétée par un dispositif de commande.

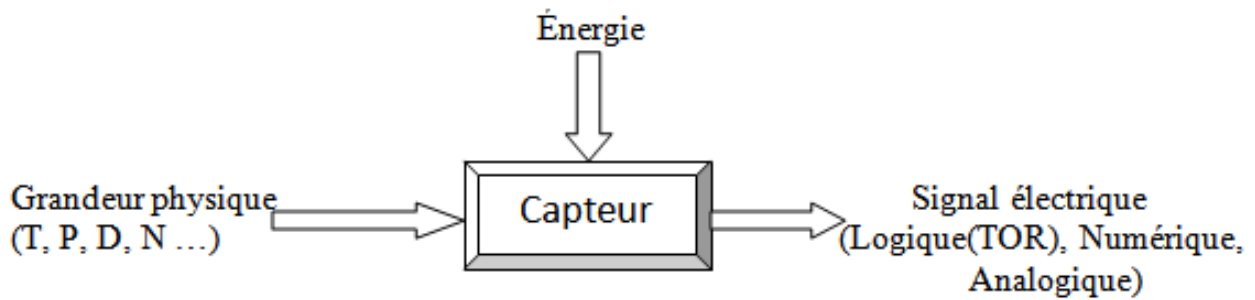


FIGURE 1.6 – Capteur

#### Caractéristiques d'un capteur

- Entendue de mesure : Valeurs extrêmes pouvant être mesurée par le capteur.
- Résolution : Plus petite variation de la grandeur mesurable par le capteur.
- Sensibilité : Variation du signal de sortie par rapport à la variation du signal d'entrée.
- Précision : Aptitude du capteur à donner une mesure proche de la valeur vraie.
- Rapidité : Temps de réaction du capteur. La rapidité est liée à la bande passante.

### 1.6.5 Description des capteurs du système proposé

#### Détecteur de niveau à lames vibrante pour liquides

##### A.Liquiphant T FTL260

##### A.1. Domaine d'application

Le Liquiphant T FTL260 est un détecteur de niveau pour liquides contenus dans des cuves de stockage et dans des réservoirs. Ce capteur peut fonctionner dans un environnement de température de -40 allant jusqu'à 150 C. Il est également employé là où les autres principes de mesure risquent d'échouer en raison de la viscosité du liquide, de la formation de dépôts, de turbulences, de courants et de bulles d'air....



FIGURE 1.7 – Détecteur de niveau à lames vibrantes « Liquiphant T FTL260 »

Le **Liquiphant T FTL260** est le capteur proposé pour la détection du niveau haut et bas des liquides dans le réservoir d'eau.

#### **A.2. Principaux avantages de Liquiphant T FTL260**

- Technologie éprouvée pour la détection des niveaux des liquides.
- Contrôle de fonctionnement sur site grâce à l'affichage lumineux externe.
- Boîtier robuste en inox.
- Forme compacte facilitant le montage.

#### **A.3. Principe de mesure**

Comme un diapason, la fourche du FTL260 est amenée à sa fréquence de résonance. Cette fréquence se modifie lorsque la fourche est recouverte de liquide.

L'électronique du FTL260 surveille la fréquence de résonance et indique si la fourche oscille librement ou si elle est recouverte de liquide.

#### **A.4. Raccordement**

Le FTL260 peut être raccordé selon deux modes de fonctionnement :

-MAX - Sécurité maximum

Le FTL260 maintient le contact électronique fermé aussi longtemps que le niveau de liquide ne recouvre pas la fourche. Exemple d'application : sécurité anti-débordement

-MIN - Sécurité minimum

Le FTL260 maintient le contact électronique fermé aussi longtemps que le niveau de liquide recouvre la fourche. Exemple d'application : protection contre la marche à vide de pompes.

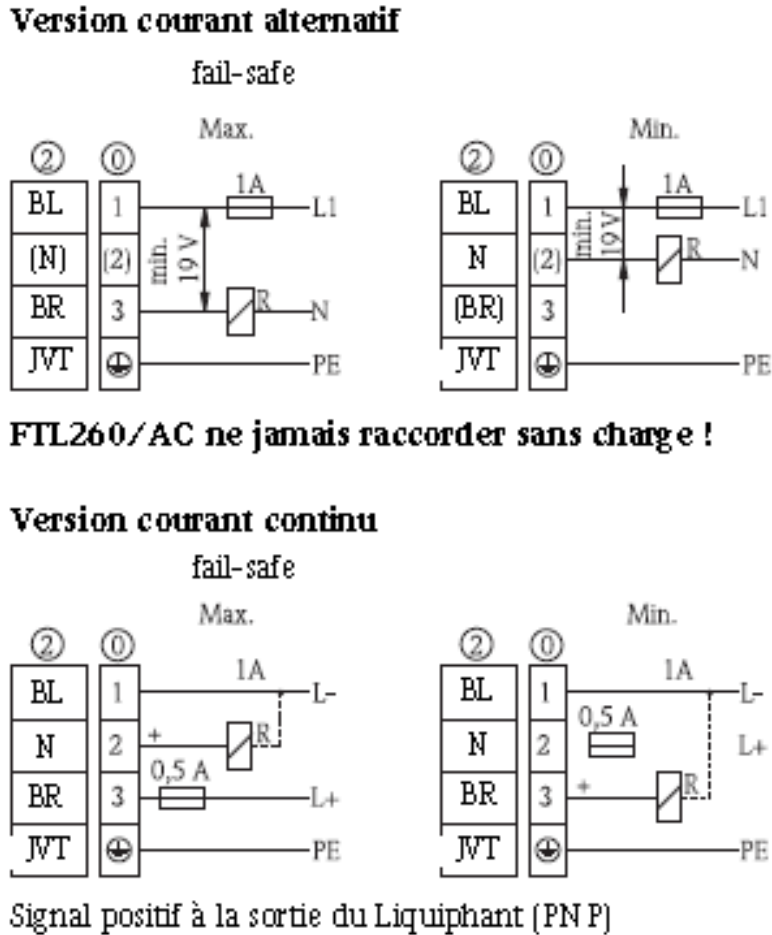


FIGURE 1.8 – Modes de fonctionnement du Liquiphant T FTL260

## 1.7 Conclusion

La description du processus de raffinage et le mode de fonctionnement de l'installation de la création du vide actuelle, nous a permis de bien comprendre son fonctionnement ainsi que le rôle de chaque constituant afin de remédier à la problématique, pour l'élaboration d'un nouveau système automatisé qui sera traité dans les chapitres suivant.

# Chapitre 2

## Généralités sur la Régulation

### 2.1 Introduction

La **régulation automatique**, actuellement rebaptisée «automatique» est noyée dans les techniques modernes de commande- robotique, productique etc., en raison surtout de l'apparition de l'électronique, puis vers les années 60 du microprocesseur et donc de l'informatique.

Un régulateur est un dispositif qui est capable, de maintenir une grandeur physique à une valeur la plus proche possible de la valeur désirée. Mais il est utile de souligner que les vieilles techniques de régulation classiques restent encore très utilisées dans l'industrie car la théorie en automatique avance bien plus vite que l'application et cela, parce que les moyens informatiques sont plus «performants» que la connaissance du système à traiter c'est à dire le modèle.

### 2.2 Définition de la régulation automatique

La régulation automatique regroupe l'ensemble des moyens matériels et techniques mis en œuvre pour maintenir automatiquement (pas d'intervention Manuelle) une ou plusieurs grandeurs physiques (installation de production, robot, alimentation électronique stabilisée, etc..) en vue d'en imposer le comportement.

Cette prise de contrôle s'effectue par l'intermédiaire de certains signaux (grandeurs physiques) qu'il est alors nécessaire de mesurer afin de déterminer l'action à entreprendre sur le système.

Lorsque des perturbations ou un changement de consigne se produisent, la régulation automatique provoque une action correctrice sur une autre grandeur physique, parmi les grandeurs d'entrée du procédé (grandeur réglante), afin de ramener la grandeur réglée vers sa consigne initiale (cas de perturbations) ou vers sa nouvelle consigne (cas de changement de consigne c'est-à-dire changement de point de fonctionnement).

Les méthodes de l'automatique offrent donc la possibilité de modifier le comportement statique et dynamique d'une grandeur physique, afin qu'elle évolue conformément aux exigences de l'application[1][2].



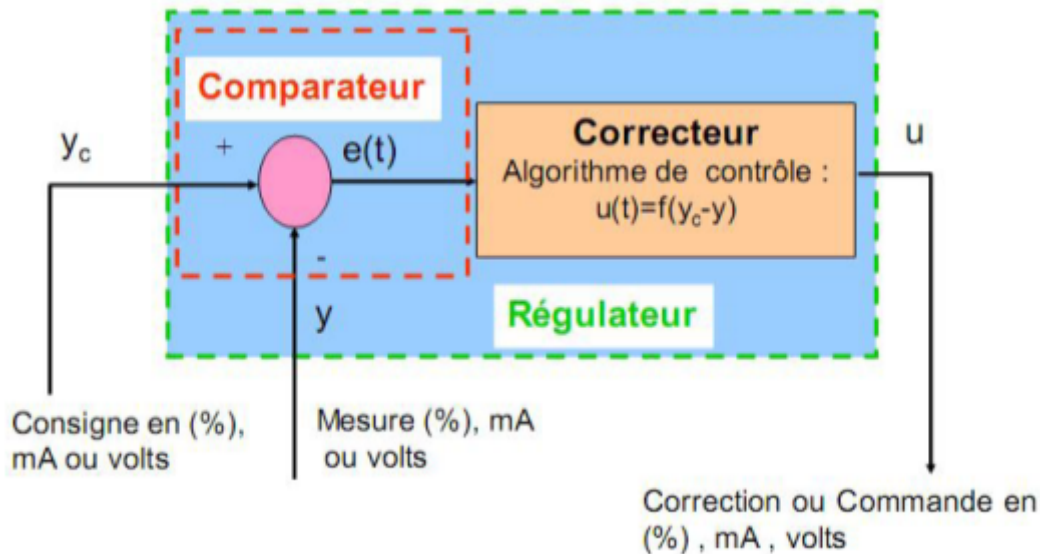


FIGURE 2.1 – Présentation d'un schéma fonctionnel d'un régulateur

## 2.3 Objectifs de la régulation

Réguler une grandeur, c'est obtenir d'elle un comportement donné, dans un environnement susceptible de présenter des variations. On ne peut pas parler de principe de régulation sans parler des lois de commandes.

En fait, les grandeurs physiques commandées varient dans le temps. Pour celles qui ne présentent que 2 états (système binaire ou TOR), tel les feux de signalisation, les commandes d'ascenseurs, de transfert de pièces par convoyeurs, etc..) on utilise une autre approche différente à la structure de boucle utilisée dans la plupart des systèmes y compris notre système (réservoir).

Les systèmes automatiques assurent en fait 2 types de fonctions :

- Maintenir la grandeur commandée, ou grandeur réglée, à une valeur de référence malgré les variations de conditions extérieures ; on parle de la régulation en sens strict.
- Répondre à des changements d'objectif, ou à un objectif variable tel que la poursuite de cible, on parle d'un fonctionnement d'asservissement[3].

Le rôle de l'automaticien (chargé d'obtenir un système régulé) sera multiple :

- Instrumenter le système : choisir les capteurs et actionneurs en fonction des besoins physiques, de coût et de performance demandés au système.
- Déterminer les relations entré-sorties du système, des capteurs et des actionneurs. On parlera des lors de :
  - Modéliser quand on s'attachera à déterminer la structure mathématique de ces relations.
  - Identifier quand on s'intéressera à calculer les coefficients du modèle.
  - Synthétiser une loi de commande (un correcteur) afin d'obtenir un système performant : précis, rapide et stable tout en s'affranchissant des influences néfastes des perturbations[4].

## 2.4 Notion de Boucle Ouverte/Fermée

### 2.4.1 Système boucle ouverte

On parle de fonctionnement en boucle ouverte quand c'est l'opérateur qui contrôle l'organe de réglage, ce n'est pas une régulation.

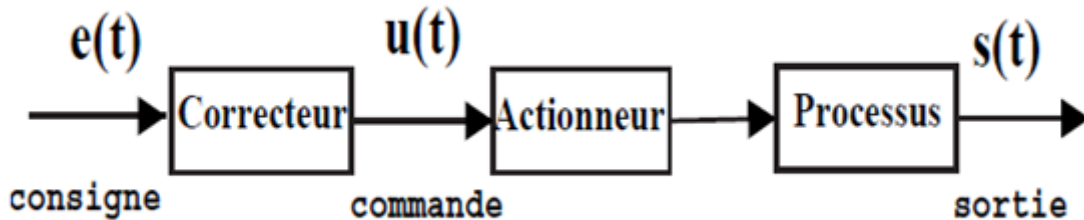


FIGURE 2.2 – Système en BO

### 2.4.2 Système boucle fermée

C'est le fonctionnement normal d'une régulation. Le régulateur compare la mesure de la grandeur réglée et la consigne et agit en conséquence pour s'en rapprocher [5].

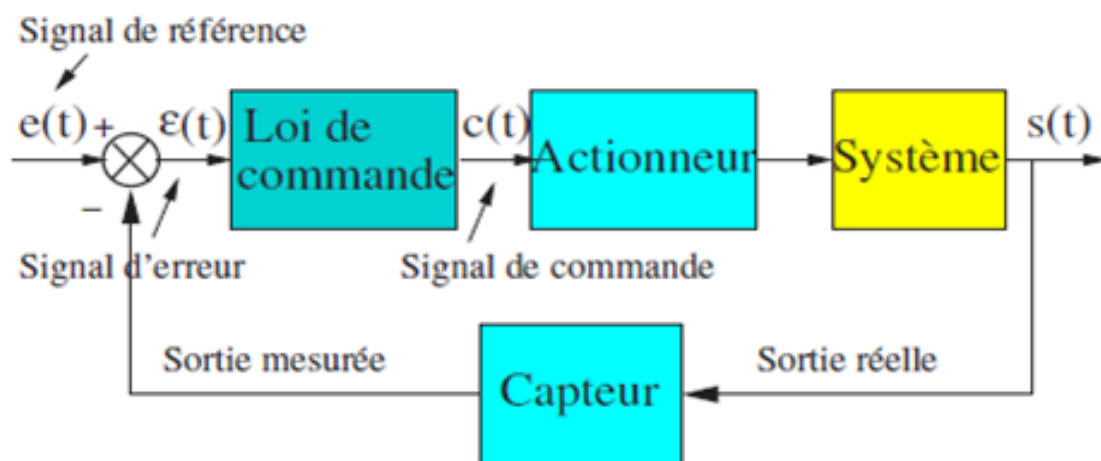


FIGURE 2.3 – Système en BF

## 2.5 Types de régulation automatique

### 2.5.1 Régulation tout ou rien (TOR)

Ce mode d'action est essentiellement discontinu. Sa réalisation impose de se fixer une limite inférieure et une limite supérieure lorsque la mesure atteint la limite inférieure l'actionneur prend une position particulier (ouverture ou fermeture pour une vanne) de façon analogue le fait d'atteindre la limite supérieure place l'actionneur dans la position contraire. La mesure oscille donc entre ces deux valeurs extrêmes et sa variation prend une allure en dents de scie. Ce réglage est simple ou le signal de commande ne prend que deux valeurs soit 0 ou 1 [6].

### 2.5.2 Régulation analogique

C'est le type de régulation où le signal du régulateur et la mesure variant d'une manière continue dans le temps. Le mode d'action analogique le plus simple est l'action proportionnelle qui est réalisée par un régulateur (p). Il convient en général aux installations ayant une grande inertie.

### 2.5.3 Régulation numérique

Le principe de la régulation numérique est que le régulateur prend la forme d'un algorithme programmé sur microprocesseur et exécuté en temps réel, i.e. impérativement à chaque période d'échantillonnage.

## 2.6 Critères de performance d'une régulation

Les performances d'une régulation peuvent se définir à partir de l'allure du signal de mesure suite à un échelon de consigne. Notons toutefois que les critères de performances classiques peuvent se résumer comme suit :

- **Stabilité** Cette condition est impérative mais avec un certain degré de stabilité (marge de sécurité). En général on impose une marge de gain de 2 à 2,5. L'utilisateur parle en termes de pompage.

- **Précision** L'exploitant demande à ce que le système possède une bonne précision en régime permanent d'où une nécessité de mettre un régulateur PI ou d'afficher un gain important dans le cas d'un régulateur P.

- **Rapidité** On demande en pratique que le système soit capable rapidement de compenser les perturbations et de bien suivre la consigne.

- **Dépassement** En général on recommande un système de régulation dont le régime transitoire soit bien amorti et dont le dépassement ne dépasse pas 5 à 10/100 de la valeur nominale[7] .

## 2.7 Définition de la régulation PID

Le régulateur standard le plus utilisé dans l'industrie est le régulateur PID (proportionnel intégral dérivé), car il permet de régler à l'aide de ses trois paramètres les performances (amortissement, temps de réponse, ...) d'un processus modélisé par un deuxième ordre. Nombreux sont les systèmes physiques qui, même en étant complexes, ont un comportement voisin de celui d'un deuxième ordre. Par conséquent, le régulateur PID est bien adapté à la plupart des processus de type industriel et est relativement robuste par rapport aux variations des paramètres du procédé.

Si la dynamique dominante du système est supérieure à un deuxième ordre, ou si le système contient un retard important ou plusieurs modes oscillants, le régulateur PID n'est plus adéquat et un régulateur plus complexe (avec plus de paramètres) doit être utilisé, au dépend de la sensibilité aux variations des paramètres du procédé[1].

\*Loi de commande du régulateur PID

$$u(t) = k_p * (e(t) + \frac{1}{T_i} * \int_{-\infty}^t e(\tau) * d\tau + T_d * \frac{de}{dt})$$

\*Fonction de transfert du régulateur PID :

$$G_c(P) = \frac{U(P)}{E(P)} = K_p * \frac{1 + P * T_i + P * T_i * T_d}{P * T_i}$$

## 2.8 Notion de correcteur PID

Sous forme d'un tableau récapitulatif, on résume les avantages et les limitations des actions de base des régulateurs PID :

Action	Points forts	Points faibles
<b>P</b>	Action instantanée	Ne permet pas d'annuler une erreur statique mais permet de la réduire
<b>I</b>	Annule l'erreur statique	Action lente Ralentit le système (effet déstabilisant)
<b>D</b>	Action très dynamique	Améliore la rapidité Apporte un effet stabilisant Sensibilité aux bruits Forte sollicitation de l'organe de commande

TABLE 2.1 – Notion de correcteur PID

## Tableau d'influence

Augmentation de	Stabilité	Précision	Rapidité
$K_p$	diminue	augmente	augmente
$T_i$	augmente	pas d'influence	diminue
$T_D$	diminue	pas d'influence	augmente

TABLE 2.2 – Tableau d'influence

## 2.8.1 Structure des régulateurs électroniques

Les différentes structures suivantes sont apparues grâce à l'emploi des amplificateurs opérationnels. On distingue trois structures pour les régulateurs PID :

La structure parallèle :

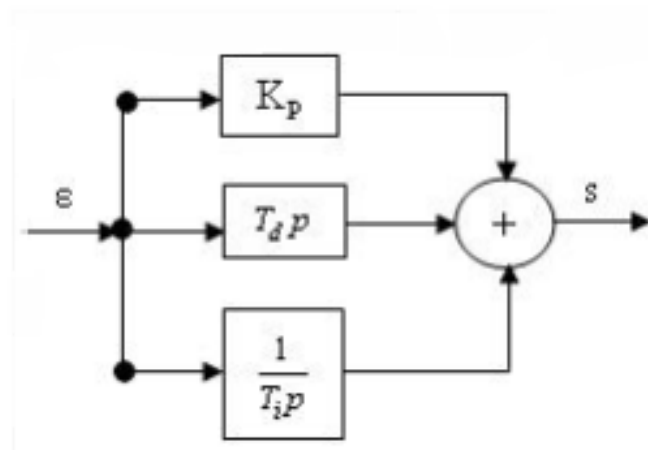


FIGURE 2.4 – PID parallèle

Fonction de transfert du régulateur PID parallèle est :

$$\frac{S(p)}{\varepsilon(p)} = K_p + T_d p + \frac{1}{T_i p}$$

La structure mixte :

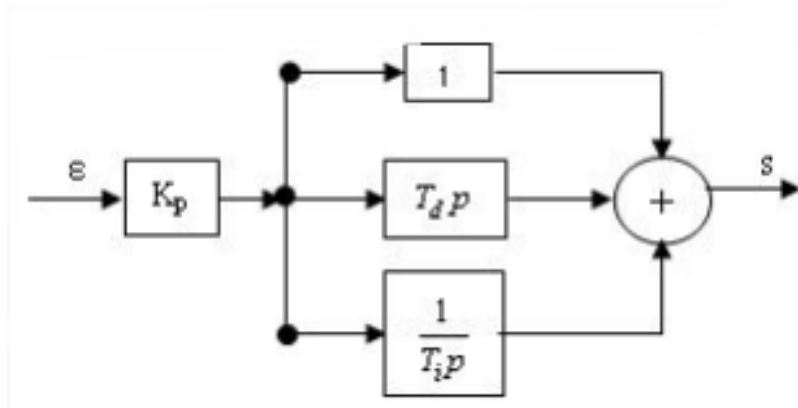


FIGURE 2.5 – PID mixte

Fonction de transfert du régulateur PID mixte est :

$$\frac{S(p)}{\varepsilon(p)} = K_p * \left(1 + \frac{1}{T_i p} + T_d p\right)$$

La structure série :

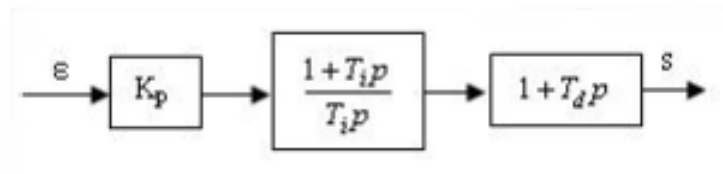


FIGURE 2.6 – PID série

Fonction de transfert du régulateur PID série est :

$$\frac{S(p)}{\varepsilon(p)} = K_p * \left(1 + \frac{1}{T_i p}\right) * (1 + T_d p)$$

Il existe un lien fort entre les paramètres puisque le réglage de l'action Proportionnelle  $K_p$  influe par exemple sur l'action Intégrale et que  $T_i$  influe sur l'action Proportionnelle[8].

## 2.8.2 Le FB 41 CONT-C

Le bloc FB 41 CONT-C (Continuos Controller) sert à réguler des processus industriels à grandeurs d'entrée et de sortie continues sur les automates programmables SIMATIC S7. Le paramétrage vous permet d'activer ou désactiver des fonctions partielles du régulateur PID et donc d'adapter ce dernier au système régulé[9].

## 2.8.3 Utilisation

Vous pouvez utiliser le régulateur comme régulateur PID de maintien mais régulations aussi comme régulateur en cascade, de mélange ou de rapport dans des à plusieurs boucles. Ses 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 et modulation de largeur d'impulsions pour régulations à deux ou trois échelons avec organes de réglage proportionnels[9].

## 2.8.4 Description

En plus des fonctions traitant la consigne et la mesure, le FB réalise un régulateur PID prêt à l'emploi avec sortie continue de la grandeur de réglage et possibilité d'influencer à la main la valeur de réglage.

**Selon le type de CPU, il sera mis en œuvre grâce au FB41 (pour les CPU 300) ou au FB (pour les CPU sans interface Profibus).** Il propose les fonctions partielles suivantes :

- Branche de consigne.
- Branche de mesure.
- Formation du signal d'erreur.
- Algorithme PID.
- Traitement de la valeur de réglage manuelle.
- Traitement de la valeur de réglage.
- Action anticipatrice

## 2.8.5 États de fonctionnement

**Démarrage et redémarrage :** Le bloc FB 41 CONT-C dispose d'un sous-programme de démarrage qui exécuté quand le paramètre d'entrée COM-RST=TRUE. A la mise en route, l'intégrateur est positionné de façon interne sur la valeur d'initialisation I-ITVAL.en cas d'appel dans un niveau d'alarme d'horloge, il continue à travailler à partir de cette valeur. Toutes les autres sorties sont positionnées sur leurs valeur par défaut[10].

## 2.8.6 Information d'erreur

Bloc ne procède pas à un contrôle interne d'erreur. Le mot indicateur d'erreur RET-VAL n'est pas employé.

Pour plus d'information sur le FB41 CONT-C consulter l'annexe 3.

## 2.9 Conclusion

Dans ce chapitre, nous avons présenté les différentes méthodes de la régulation en boucle ouverte et en boucle fermé, une description générale des systèmes industriels ainsi que quelque méthode de régulation numérique, qui se résumant dans l'utilisation des régulateurs PID, toute en essayant de répondre au cahier des charges élaboré.



# Chapitre 3

## LES AUTOMATES PROGRAMMABLES

### Historique :

Les automates programmables industriels sont apparus à la fin des années soixante, à la demande de l'industrie automobile américaine (GM), qui réclamait plus d'adaptabilité de leurs systèmes de commande.

Les coûts de l'électronique permettant alors de remplacer avantageusement les technologies actuelles.

**Avant :** utilisation de relais électromagnétiques et de systèmes pneumatiques pour la réalisation des parties commandes===*Logique câblée*

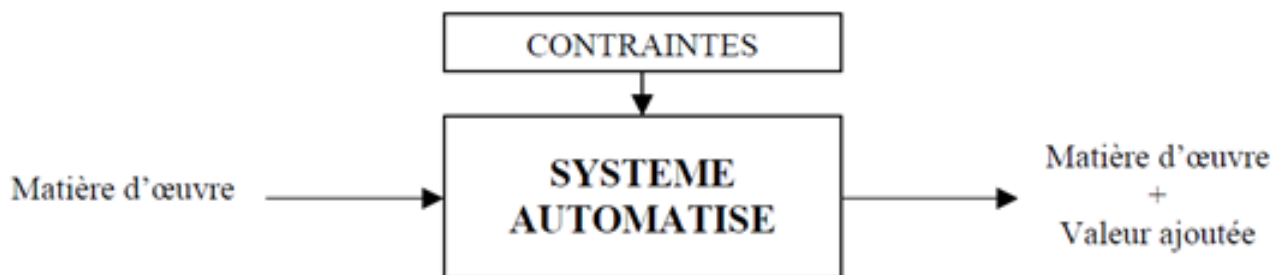
Inconvénients : cher, pas de flexibilité, pas de communication possible.

**Solution :** utilisation de systèmes à base de microprocesseurs permettant une modification aisée des systèmes automatisés===*logique programmée*[1].

### 3.1 Systèmes automatisés

Un système automatisé est un ensemble d'éléments en interaction, et organisés dans un but Précis : agir sur une **matière d'œuvre** afin de lui donner une **valeur ajoutée**.

Le système automatisé est soumis à des **contraintes** : énergétiques, de configuration, de réglage et d'exploitation qui interviennent dans tous les modes de marche et d'arrêt du système[1].



### 3.1.1 Objectif de l'automatisation :

L'objectif de l'automatisation des systèmes est de produire, en ayant recours le moins possible à l'homme, des produits de qualité et ce pour un coût le plus faible possible, augmentation de la sécurité, s'adapter a des contextes particulier.

### 3.1.2 Structure d'un système automatisé :

Tout système automatisé peut se décomposer selon le schéma ci-dessous :

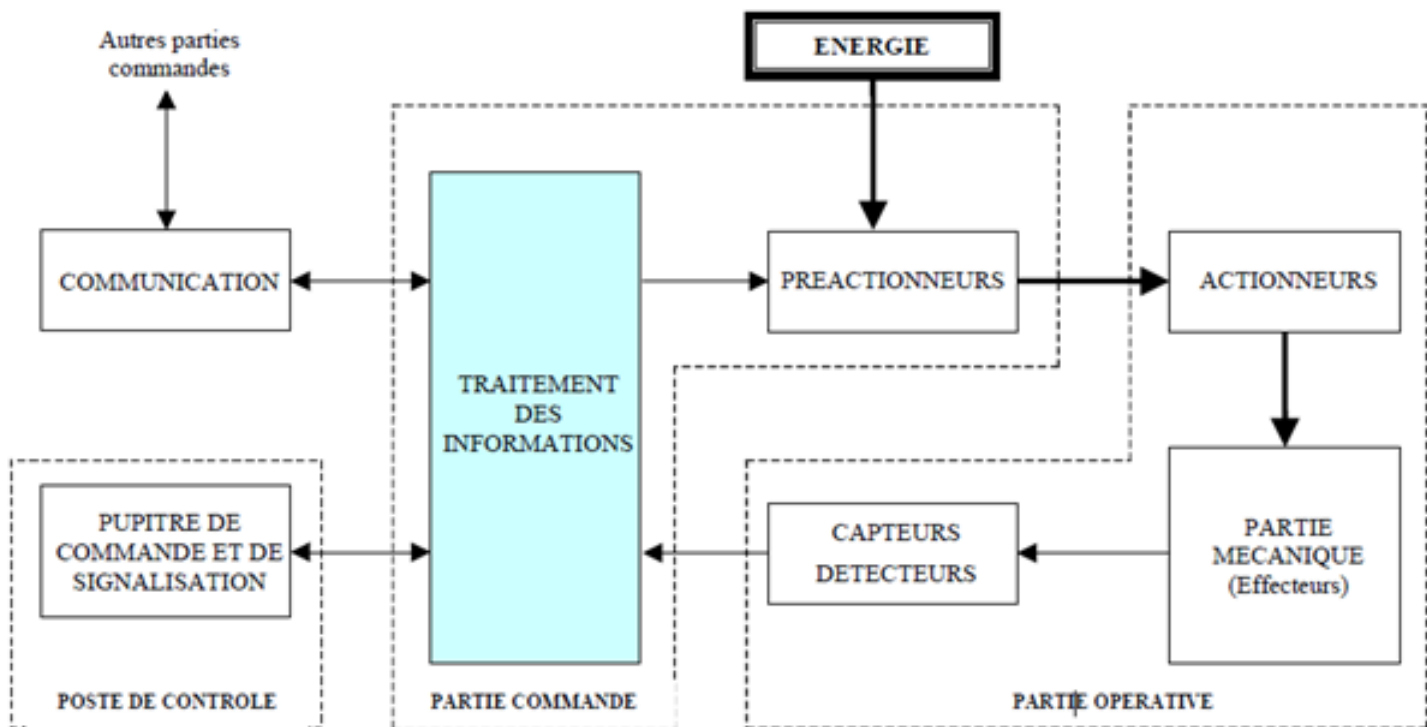


FIGURE 3.1 – Structure d'un système automatisé

Pendant le fonctionnement, un dialogue continu s'établit dans le système, permettant ainsi le déroulement correct du cycle défini dans le cahier des charges[6].

#### Partie opérative :

Elle agit sur la matière d'œuvre afin de lui donner sa valeur ajoutée.

**Les actionneurs** (moteurs, vérins) agissent sur la partie mécanique du système qui agit à son tour sur la matière d'œuvre.

**Les capteurs / détecteurs** permettent d'acquérir les divers états du système.

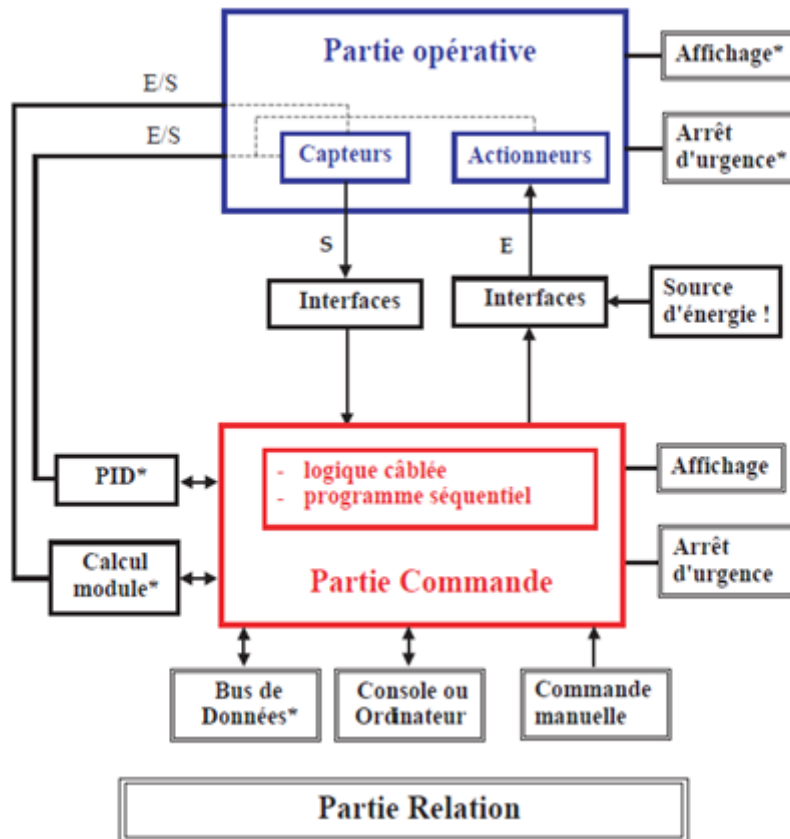


FIGURE 3.2 – Procédé automatisé

\* En option

**Partie commande :**

Elle donne les ordres de fonctionnement à la partie opérative.

**Les pré-actionneurs** permettent de commander les actionneurs ; ils assurent le **transfert d'énergie** entre la source de puissance (réseau électrique, pneumatique ...) et les actionneurs. Exemple : contacteur, distributeur ...

Ces pré-actionneurs sont commandés à leur tour par le bloc traitement des informations.

Celui-ci reçoit les consignes du pupitre de commande (opérateur) et les informations de la partie opérative transmises par les capteurs / détecteurs.

En fonction de ces consignes et de son programme de gestion des tâches (implanté dans un automate programmable ou réalisé par des relais (on parle de logique câblée)), elle va commander les pré-actionneurs et renvoyer des informations au pupitre de signalisation ou à d'autres systèmes de commande et/ou de supervision en utilisant un réseau et un protocole de communication.

**La partie relation :**

Sa complexité dépend de l'importance du système. Elle regroupe les différentes commandes nécessaires au bon fonctionnement du procédé, c'est à dire marche/arrêt[11].

## 3.2 Les avantages et les inconvénients d'un système automatisé

### 3.2.1 Les avantages :

Ils sont nombreux et citons principalement :

- Accélération des processus de production, dans tous les domaines industriels, en gardant un produit de qualité.
- La suppression de certaines tâches fatigantes, répétitives ou nocives pour l'homme.
- Les SPA s'adaptent facilement (commande et puissance) à tous les milieux de production (industries de l'automobile, du verre, du bois, du papier ainsi que le tri et l'emballage).
- La création de métiers nouveaux.
- La souplesse d'utilisation qu'ils présentent peut répondre aux problèmes simples comme aux extrêmement complexes.

### 3.2.2 Les inconvénients :

Ils existent, et sont à prendre en considération comme :

- Le coût élevé du matériel, principalement dans les systèmes hydrauliques.
- La maintenance doit être parfaitement structurée et réalisée par un personnel spécialisé (électroniciens ou automaticiens).
- Il faut, cependant, noter que les systèmes automatisés peuvent être la cause de suppression d'emplois[12].

## 3.3 les automates programmables :

Les Automates Programmables Industriels (API) également appelés Programmable logic controllers (PLC) sont des semi-conducteurs, membres de la famille de l'ordinateur, en utilisant des circuits intégrés au lieu de dispositifs électromécaniques pour mettre en œuvre le contrôle des fonctions d'automatisme, pour assurer la commande des pré-actionneurs et d'actionneurs à partir d'informations logique, analogique ou numérique. Ils sont capables de stocker des instructions, telles que le séquençage, le calendrier, le comptage, l'arithmétique, les données de manipulation et de communication, pour contrôler des machines et des procédés industriels.

Toutefois, en termes simples les automates programmables industriels sont des contrôleurs industriels prêts, avec leurs racines de conception basées sur les principes d'application simple et pratique, avec une architecture spécialement conçus comme appareils de terrain, et à la fois avec leurs unités centrales (l'automate lui-même) et leur interfaçage des circuits pour des connexions d'entrée / sortie vers le monde réel [13].

## 3.4 Architecture des automates

### 3.4.1 Aspect extérieur :

Les automates peuvent être de type compact ou modulaire.

**A) type compact**, on distinguera les modules de programmation (LOGO de Siemens, ZELIO de Schneider, MILLENIUM de Crouzet ...) des micro-automates.

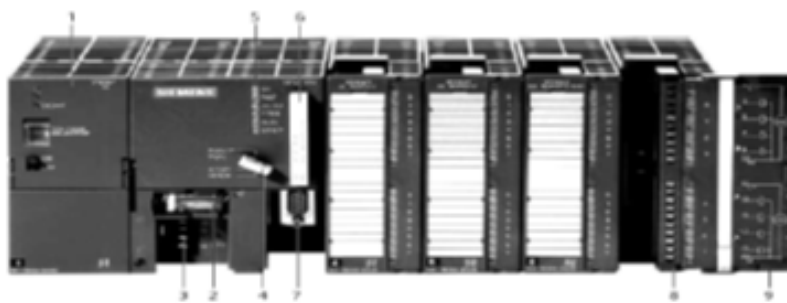
Il intègre le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage rapide, E/S analogiques ...) et recevoir des extensions en nombre limité. Ces automates, de fonctionnement simple, sont généralement destinés à la commande de petits automatismes.

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

Ces automates sont intégrés dans les automatismes complexes où puissance, capacité de traitement et flexibilité sont nécessaires.



Automate compact (Allen-Bradley)



Automate modulaire (Siemens)

FIGURE 3.3 – Image réelle d'API

### 3.4.2 Structure interne :

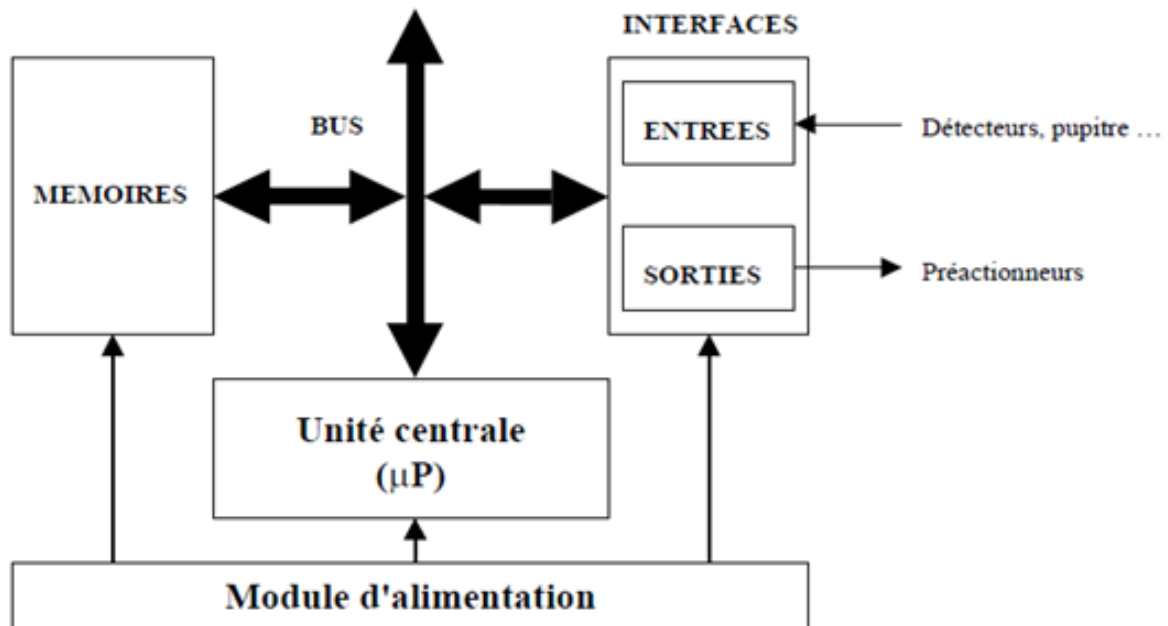


FIGURE 3.4 – Structure interne des automates

#### 1. Module d'alimentation :

L'alimentation du système joue un rôle majeur dans le fonctionnement total du système. En fait, il peut être considéré comme le "gestionnaire de premier niveau" de la fiabilité et l'intégrité du système. Sa fonction n'est pas seulement de fournir des tensions continues internes pour les composants du système, mais aussi pour surveiller et réguler les tensions fournies et prévenir la CPU si quelque chose ne va pas. Le bloc d'alimentation a pour fonction de fournir une puissance bien régulée et de protection pour les autres composants du système.

#### 2. Unité centrale ou CPU :

L'unité centrale est le regroupement du processeur et de la mémoire centrale, elle commande l'interprétation et l'exécution des instructions programmées. Elle est aussi chargée de détecter les pannes de communication, ainsi que d'autres défaillances, qui peuvent survenir pendant le fonctionnement du système, Il doit alerter l'opérateur ou le système en cas de dysfonctionnement.

À base de microprocesseur, l'unité centrale réalise toutes les fonctions logiques, arithmétiques et de traitement numérique (transfert, comptage, temporisation ...). Ce module se compose essentiellement de :

##### a. Microprocesseur :

Il constitue le cœur de la CPU, Son rôle consiste d'une part à organiser les différentes relations entre la zone mémoire et les interfaces d'E/S et d'autre part à gérer les instructions du programme.

**b. Le bus :**

Le Bus est un ensemble de pistes conductrices (pistes en cuivre) par lesquelles s'achemine une information binaire (suite de 0 ou 1), c'est à dire ensemble de fils autorisant le passage des informations entre les quatre secteurs (l'alimentation, la mémoire, le processeur et l'interface E/S) de l'automate. L'unité centrale dispose de trois bus :

- Le bus de données.
- Le bus d'adresses.
- Le bus de commandes.

**c. Mémoires :**

Elle est conçue pour recevoir, gérer et stocker des informations issues des différents secteurs du système qui sont :

- Le terminal de programmation.
- Le processeur, qui lui gère et exécute le programme. Elle reçoit également des informations en provenance des capteurs. Il existe dans les automates plusieurs types de mémoires qui remplissent des fonctions différentes :
- La conception et l'élaboration du programme font appel à la RAM et l'EEPROM.
- La conservation du programme pendant l'exécution de celui-ci fait appel à une EPROM.

**d. Interfaces d'entrées / sorties** Les interfaces d'entrées / sorties permettent à l'unité centrale de communiquer avec l'environnement ou les périphériques [13].

**3. Modules d'entrées/sorties :**

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

**3.1. Modules d'entrées et sorties TOR (Tout Ou Rien) :****a. Modules d'entrées TOR (Tout Ou Rien)**

L'automate reçoit ses informations sur le processus via les capteurs de signaux reliés aux entrées. Les modules d'entrée TOR permettent de recevoir les signaux des différents capteurs logiques qui peuvent être des détecteurs qui reconnaîtront si la pièce d'usinage se trouve à une position donnée (détecteurs des niveaux -haut et bas-, Cellules photoélectrique ...) ou de simples commutateur ou interrupteur qui peuvent être fermés ou ouverts. Ce qui fait que l'information délivrée par ces capteurs et qui sera traitée par la CPU ne peut prendre que deux valeurs 0 ou 1.

**b. Modules de sorties TOR (Tout Ou Rien)**

Ces modules permettent de délivrer des signaux qui permettent à l'automate d'agir sur les pré-actionneurs du système à commander tels que (Vanne Electromagnétique, Electrovannes, Contacteurs, Pompes, Voyants...).

### 3.2. Modules d'entrées et de sorties Analogiques :

#### a. Modules d'entrées Analogiques :

L'information traitée est continue et prend une valeur qui évolue dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (débit, niveau, pression, interface, température...).

#### b. Modules de sorties Analogiques :

Ils émettent un signal analogique qui représente l'état que doit prendre un actionneur entre deux limites. Ce module est muni d'un convertisseur analogique-numérique[6].

## 3.5 Présentation de quelques gammes SIMATIC

Il existe plusieurs gammes de SIMATIC. On trouve le SIMATIC S7 et le SIMATIC M7 :

### 3.5.1 SIMATIC S7 :

Dans la gamme S7 on distingue trois grandes familles d'automates programmables industriels décrites dans ces paragraphes qui suivent :

#### SIMATIC S7-200

La famille S7-200 est constituée de micro-automates programmables utilisables dans des applications d'automatisations variées. La Figure ci-dessous présente un micro-automate S7 -200. Son dessin compact, ses possibilités d'expansion, son faible prix et son important jeu d'opérations en font une solution idéale pour la commande de petites applications. On outre, le large choix de tailles et de tensions de CPU offre la souplesse nécessaire pour résoudre un problème d'automatisation.



FIGURE 3.5 – API SIMENS S7-200

Un automate programmable S7-200 consiste en une CPU S7-200 seule ou complétée de divers modules d'extension facultatifs connectés à cette dernière à l'aide d'un connecteur de bus fourni avec ce module d'extension[12].



### SIMATIC S7-300

La famille S7-300 est constituée d'automates programmables de conception modulaire utilisés pour des automatismes de gamme, et peuvent être connectés entre eux au moyen d'un câble-bus PROFIBUS.

Un automate S7-300 consiste en une CPU, un module d'alimentation PS, un module de signaux FM, un module de signaux SM et un processeur de communication. Comme indiqué dans la figure suivante[12].

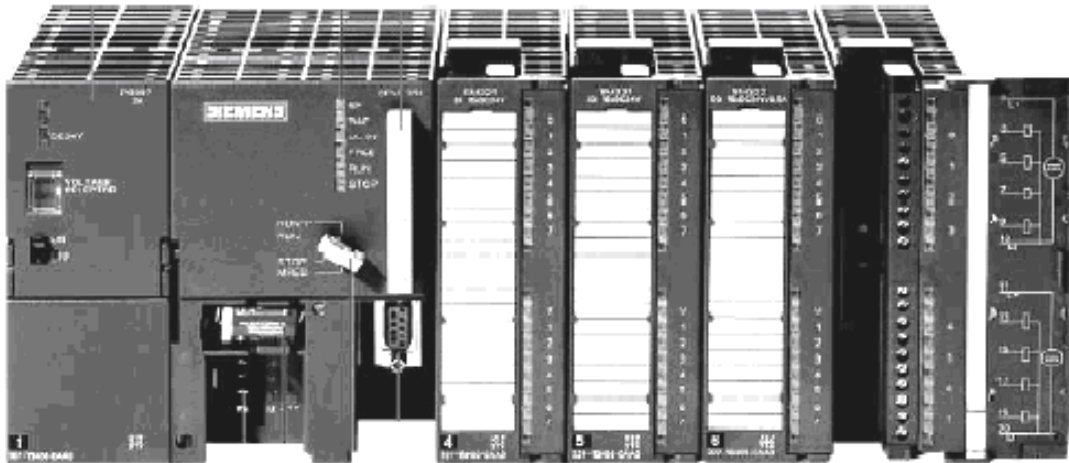


FIGURE 3.6 – API SIMENS S7-300

### SIMATIC S7-400

La famille S7-400 est aussi constituée d'automates programmables de conception modulaire. Pratiquement chaque tâche d'automatisation peut être résolue par un choix approprié des constituants de S7-400 et avec la possibilité d'expansion de plusieurs modules.

Les modules se présentent sous forme de boîtiers que l'on adapte sur un châssis [12].



FIGURE 3.7 – API SIMENS S7-400

### 3.5.2 SIMATIC M7 :

Les SIMATIC M7 sont des calculateurs industriels compatibles PC. Il s'agit d'un système modulaire sous boîtier, construit dans la technique des automates SIMATIC S7. Il peut être intégré dans un automate S7 300/400 ou être utilisé comme système autonome avec une périphérie choisie dans la gamme S7. En raison de son architecture PC standard, les contrôleurs M7-300 et M7-400 constituent une extension programmable de la plate-forme d'automatisation SIMATIC. Les programmes utilisateur pour SIMATIC M7 peuvent être écrits dans un langage de haut niveau comme C et C++, les services fournis par le système d'exploitation sont extrêmement importants. Le système d'exploitation effectue les tâches suivantes pour ces applications :

- Accès au matériel.
- Gestion des ressources.
- Système de liaison.
- Communiquer avec les autres composants du système[12].

## 3.6 Critères de choix d'un automate programmable

Le choix d'un automate programmable est en premier lieu le choix d'une société ou d'un groupe où les contacts commerciaux et expériences vécues sont déjà un point de départ.

Les grandes sociétés privilégieront deux fabricants pour faire jouer la concurrence et pouvoir "se retourner" en cas de "perte de vitesse" de l'une d'entre elles.

Le personnel de maintenance doit toutefois être formé sur ces matériels et une trop grande diversité des matériels peut avoir de graves répercussions. Un automate utilisant des langages de programmation de type GRAFCET est également préférable pour assurer les mises au point et dépannages dans les meilleures conditions. La possession d'un logiciel de programmation est aussi source d'économies (achat du logiciel et formation du personnel). Des outils permettant une simulation des programmes sont également souhaitables.

Il faut ensuite quantifier les besoins :

- Nombre d'entrées / sorties : le nombre de cartes peut avoir une incidence sur le nombre de racks dès que le nombre d'entrées / sorties nécessaires devient élevé.
- Type de processeur : la taille mémoire, la vitesse de traitement et les fonctions spéciales offertes par le processeur permettront le choix dans la gamme souvent très étendue.
- Fonctions ou modules spéciaux.
- Fonctions de communication : l'automate doit pouvoir communiquer avec les autres systèmes de commande (API, supervision ...) et offrir des possibilités de communication avec des standards normalisés (Profibus ...).

### 3.6.1 Quantification et choix d'automate

Dans notre cas, le choix est porté sur l'automate programmable SIEMENS S7-300 CPU 314, et cela pour les raisons suivantes :

#### 1. Les compétences/expériences de l'équipe d'automaticiens en mise en œuvre et en programmation de la gamme d'automate :

- matériels et logiciels conçus pour une utilisation facile par les électriciens et les techniciens.
- Familiarisation (compétences et expériences) de l'équipe d'automaticiens travaillant au complexe CEVITAL-SPA.

#### 2. Qualité du service après vente :

SIEMENS est une marque de renommée mondiale sur le marché des API. Une entreprise d'une telle importance n'acceptera jamais de ternir son image, ce qui fait qu'elle offre à ces clients un service après vente des plus compétent. [12]

#### 3. Type d'entrées/sorties :

L'automate programmable S7-300 dispose de module d'entrées/sorties de type logique analogique.

#### 4. Nombre d'entrées/sorties :

Le châssis de S7-300 peut prendre 8 modules de signaux de communication (analogique ou TOR) ce qui est largement suffisant pour notre application.

## 3.7 Présentation de logiciel de programmation STEP7

STEP7 fait partie de l'industrie logicielle SIMATIC. Le logiciel STEP 7 permet de concevoir, configurer, programmer, tester, mettre en service et maintenir les systèmes d'automatisation SIMATIC.

Les tâches de bases qu'il offre à son utilisateur lors de la création d'une solution d'automatisation sont :

- La création et gestion de projets.
- La configuration et le paramétrage du matériel et de la communication.
- La gestion des mnémoniques.
- La création des programmes.
- Le chargement de programmes dans les systèmes cibles.
- Le test de l'installation d'automatisation.
- Le diagnostic lors des perturbations dans l'installation[12].

### 3.7.1 Création d'un projet sous STEP7

La figure ci-dessous nous montre les étapes à suivre pour concevoir un projet avec STEP7.

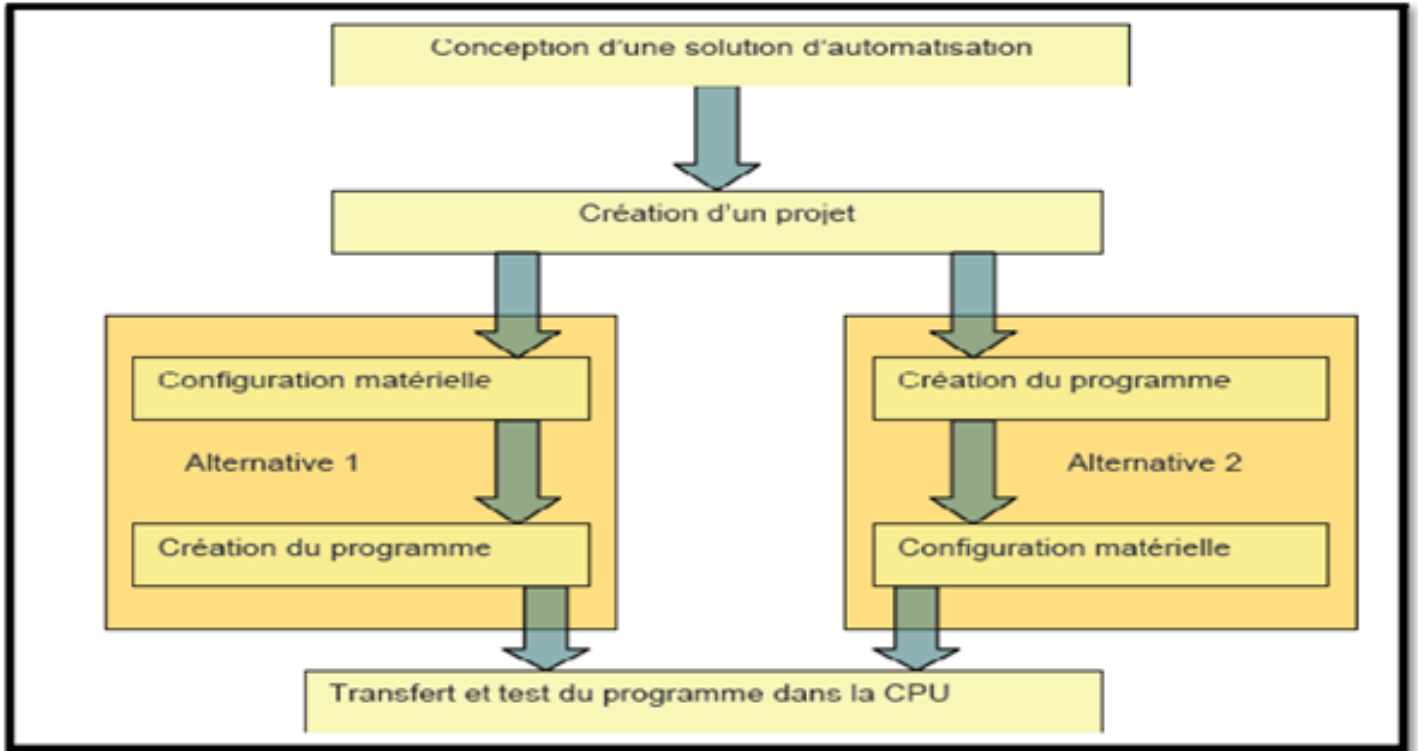


FIGURE 3.8 – Organigramme pour la création de projets sous STEP7

Pour créer un projet avec Step7 on peut lancer l'assistant de création de projet Step7, ou créer directement un projet que l'on configurera soi même[14].

## 3.8 Présentation de WinCC Flexible

### 3.8.1 Introduction :

Lorsque la complexité des processus augmente et que les machines et installations doivent répondre à des spécifications de fonctionnalité toujours plus sévères, l'opérateur a besoin d'un maximum de transparence. Cette transparence s'obtient au moyen de l'Interface Homme-Machine (IHM). Un système IHM constitue l'interface entre l'homme (opérateur) et le processus (machine/installation). Le contrôle proprement dit du processus est assuré par le système d'automatisation. Il existe par conséquent une interface entre l'opérateur et WinCC flexible (sur le pupitre opérateur) et une interface entre WinCC flexible et le système d'automatisation[15].

### 3.8.2 Les tâches d'un système IHM :

- Représentation du processus
- Commande du processus
- Vue des alarmes
- Archivage de valeurs processus et d'alarmes
- Gestion des paramètres de processus et de machine[15].

### 3.8.3 SIMATIC WinCC flexible

WinCC flexible est le logiciel IHM pour la réalisation, par des moyens d'ingénierie simples et efficaces, de concepts d'automatisation évolutifs, au niveau machine. Et pour réaliser ces concepts, on utilise les composants suivants :

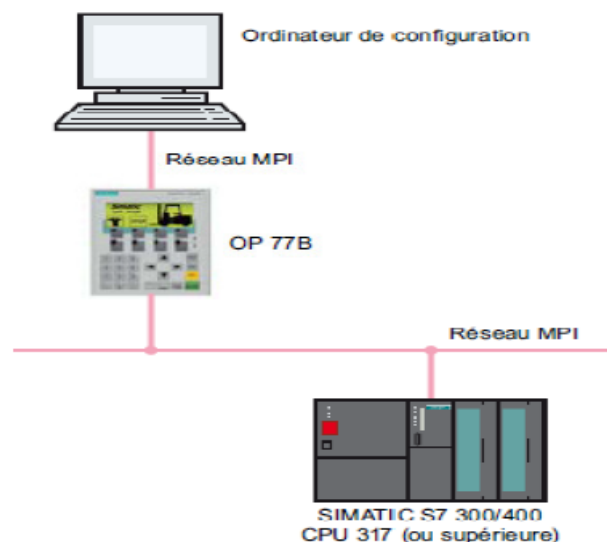


FIGURE 3.9 – Système d'automatisation avec un pupitre

Un ou plusieurs pupitres peuvent être reliés à un ou plusieurs systèmes d'automatisation via un bus système (p. ex. PROFIBUS ou Ethernet)[16].

### 3.8.4 Eléments de WinCC flexible :

#### WinCC flexible Engineering Système :

WinCC flexible Engineering System est le logiciel avec lequel vous réalisez toutes les tâches de configuration requises. L'édition WinCC flexible détermine les pupitres opérateurs de la gamme SIMATIC HMI pouvant être configurés[15].

#### Principe :

Lorsque vous créez ou ouvrez un projet sous WinCC flexible, l'écran de l'ordinateur de configuration affiche WinCC flexible Workbench. La fenêtre de projet affiche la structure du projet et permet de gérer celui-ci[15].

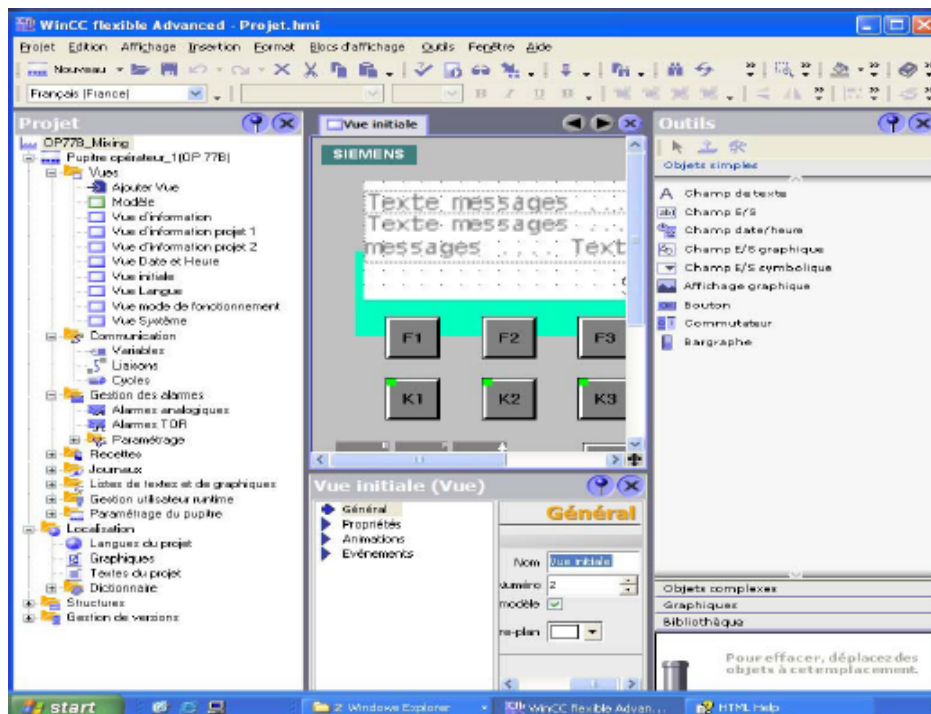


FIGURE 3.10 – Fenêtre de WinCC flexible

\* La zone de travail sert à éditer les objets du projet. Tous les éléments de WinCC flexible sont disposés autour de la zone de travail. A l'exception de la zone de travail, vous pouvez disposer et configurer, déplacer ou masquer par exemple tous les éléments comme vous le souhaitez.

Dans la fenêtre du projet tous les éléments et tous les éditeurs disponibles d'un Projet sont affichés dans l'arborescence et peuvent y être ouverts. Dans la fenêtre de projet, vous pouvez de plus accéder aux propriétés du projet et au paramétrage du pupitre utilisateur.

Dans la fenêtre des propriétés vous éditez les propriétés des objets, par exemple la Couleur des objets graphiques.

La fenêtre d'outils vous propose une sélection d'objets que vous pouvez insérer Dans vos vues, par exemple des objets graphiques et des éléments de commande. La fenêtre d'outils contient en outre des bibliothèques d'objets et collections de blocs d'affichage prêts à l'emploi.

**WinCC flexible Runtime :**

WinCC flexible Runtime est le logiciel de visualisation de process. Dans Runtime, vous exécutez le projet en mode process[16].

**1- Principe :**

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

- Communication avec les automates.
- Affichage des vues à l'écran.
- Commande du processus, par exemple spécification de consignes ou ouverture et fermeture de vannes.
- Archivage des données de Runtime actuelles, des valeurs processus et événements d'alarme par exemple[15].

**2- Variable :**

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

- Les variables externes permettent de communiquer et d'échanger des données entre les composants d'un processus automatisé, entre un pupitre opérateur et un automate.
- Les variables internes ne possèdent aucun lien avec l'automate. Elles sont enregistrées dans la mémoire du pupitre[16].

**3.8.5 Intégration de WinCC flexible à STEP7 :**

Pour intégrer un projet WinCC existant dans un projet STEP 7, procédez comme suit :

1. Ouvrez la configuration WinCC flexible.
2. Sélectionnez le menu "Projet > Intégrer dans le projet STEP 7..." Le dialogue "Intégrer dans les projets STEP 7" s'ouvre.
3. Sélectionnez dans le dialogue le projet STEP 7 correspondant.

Si le projet souhaité ne se trouve pas dans la liste, naviguez par le champ "Rechercher dans" vers le dossier qui contient le projet STEP 7.

Après la sélection du projet STEP 7, l'intégration est exécutée[14].

## 3.9 Conclusion

L'analyse du problème mène à une déduction d'une base des entrées et des sorties qui nous permettra par la suite de réfléchir au choix de la gamme automate programmable, de la CPU et les types de modules qui vont être utilisés. Nous avons présenté les caractéristiques de l'API S7-300 de la firme SEMENS qui est l'automate choisit, pour réguler l'installation de niveau d'eau.



# Chapitre 4

## Simulation et Résultats

### 4.1 Introduction

Dans ce chapitre nous allons exposer les résultats que nous avons trouvés après la programmation pour l'automatisation de notre système et les résultats de la simulation.

### 4.2 L'OB1

L'OB1 gère le programme en fonctionnement normal, on peut utiliser l'OB1 pour programmer directement à l'intérieur de celui-ci des ordres ou alors appeler des Blocks (FCs ou FBs) qui permettront de structurer le programme. Ainsi on pourra créer un Block pour une pompe avec tous ses défauts, ses status (HMI), ses modes opératoires (Auto/Manu), ces commandes (Marche/Arrêt), ceci est valable pour des vannes où autres.

100ms			ex : 15ms	100ms			ex : 15ms
OB1	OB1	OB1	OB35	OB1	OB1	OB1	OB35

FIGURE 4.1 – Exemple d'interruption de bloc par rapport à l'OB1

Comme on peut le voir sur le tableau ci-dessus, l'OB35B est appelé cycliquement toutes les 100ms (on prendra comme exemple que le programme à l'intérieur de l'OB35 s'exécute en 15ms) et vient interrompre l'appel de l'OB1.

**Remarque** : L'OB1 à la plus basse priorité, il peut être interrompu par tous les autres OBs.

### 4.3 Les blocs de données d'instance (DB)

Les blocs de données d'instance sont affectés au FB/SFB au moment de l'appel du bloc. Ils sont générés automatiquement lors de la compilation. Les DB d'instances, servent à sauvegarder le contexte d'un FB entre 2 appels, les paramètres sauvegardés sont : IN, OUT, IN-OUT et STATIC, les paramètres TEMP ne sont pas sauvegardés.

#### 4.3.1 Création d'un DB

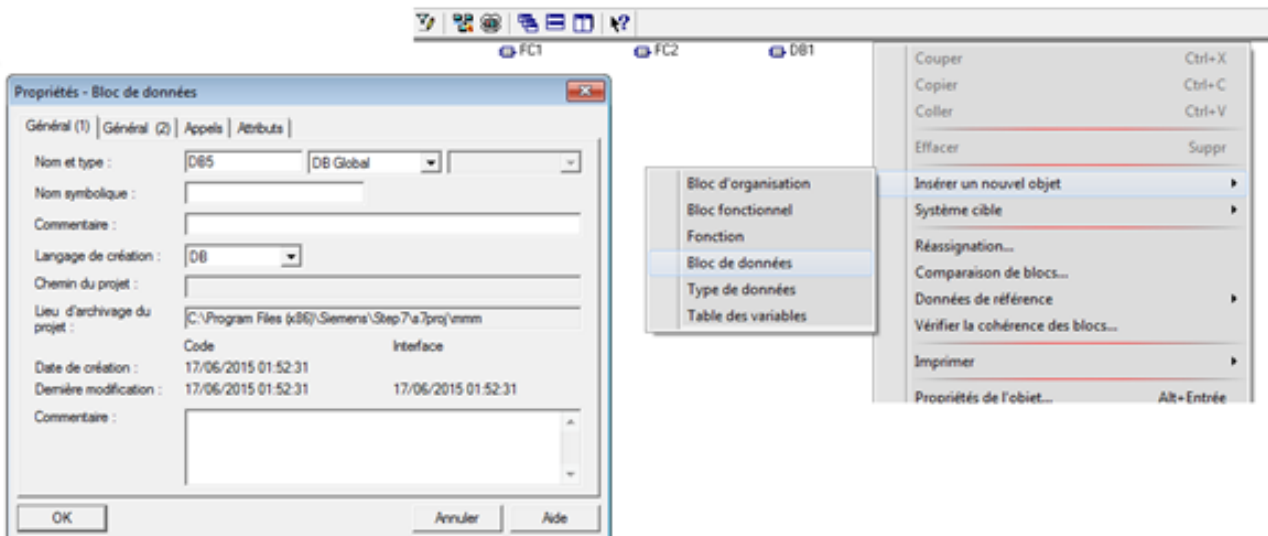


FIGURE 4.2 – Création d'un DB

#### 4.3.2 Contenu d'un DB

Adresse	Nom	Type	Valeur initiale	Commentaire
0.0		STRUCT		
+0.0	cmprst	BOOL	FALSE	
+0.1	manon	BOOL	FALSE	
+0.2	pvperon	BOOL	FALSE	
+0.3	psel	BOOL	FALSE	
+0.4	isel	BOOL	FALSE	
+0.5	dsel	BOOL	FALSE	
+2.0	sp	REAL	0.000000e+000	
+6.0	MAN	REAL	0.000000e+000	
+10.0	GAIN	REAL	3.000000e+000	
+14.0	Ti	TIME	T#15S	
+18.0	Td	TIME	T#15MS	
=22.0		END_STRUCT		

FIGURE 4.3 – Contenu d'un DB

On obtient le type de fenêtre ci-dessus à l'ouverture d'un DB. Dans la barre d'outils on peut choisir dans "View" soit d'être en "Data View" ou en "Declaration View". La première option permet de voir et de modifier les données en état actuel et la seconde permet de modifier les valeurs à l'état initial.

## 4.4 LES FCS

Les FCs sont des fonctions permettant la programmation de sous-programmes. Le plus souvent on les appellera par l'intermédiaire de l'OB1.

Le bloc utilise des variables temporaires à l'intérieur du bloc pour ces calculs et aussi pour l'affectation de ces E/S. En appelant ce FC dans un autre endroit du programme (FC, OB, ...), alors celui-ci exécutera son contenu à l'aide de ses paramètres d'entrées et modifiera des valeurs de sorties. On raccordera aux E/S du FC les variables faisant partie de notre table de mnémonique.

### 4.4.1 Création d'une FC

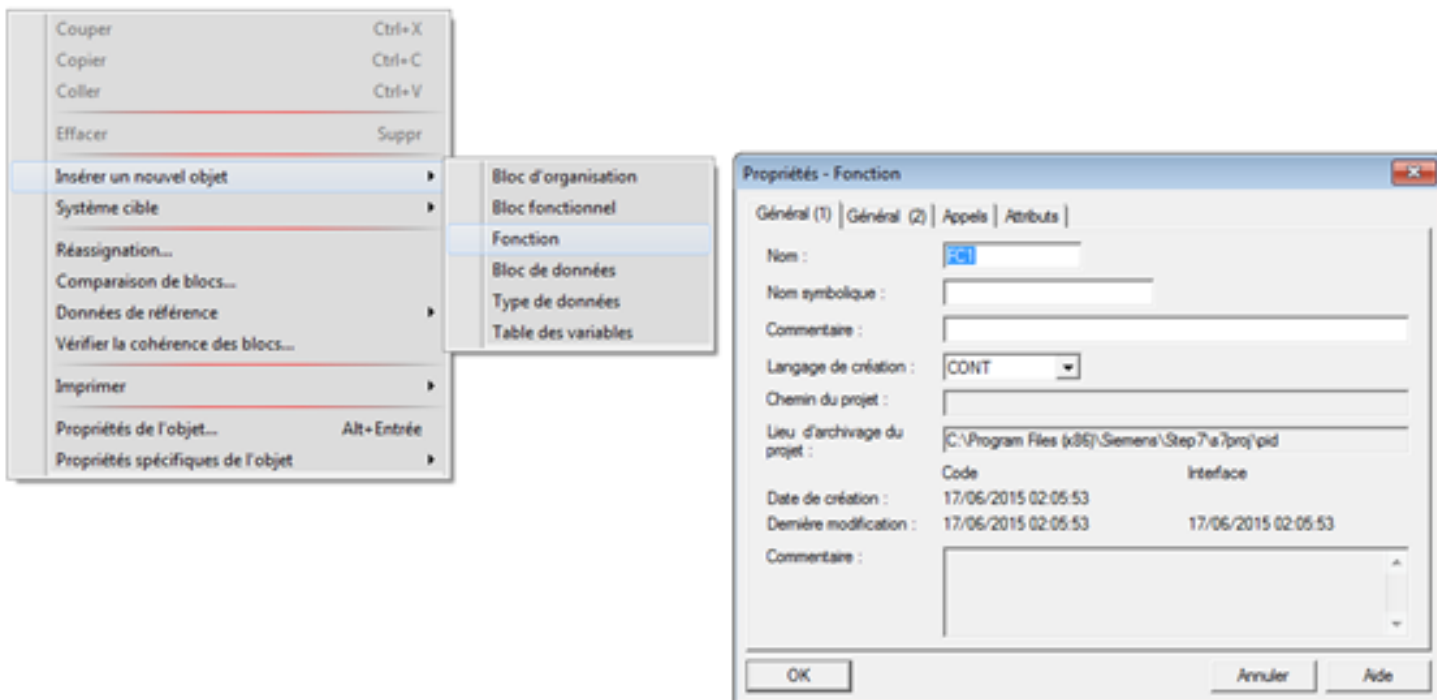
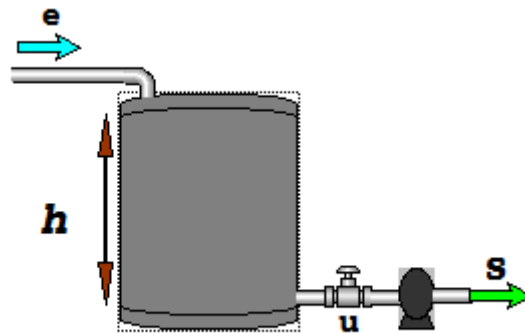


FIGURE 4.4 – Création d'une FC

## 4.5 Réalisation d'un PID sous STEP7 avec le bloc de régulation continue FB41 CONT-C

### 4.5.1 Régulateur PID, données du problème



avec (e) et (s) c'est un débit. Ce module explique la réalisation pas à pas d'un régulateur PID avec un système SIMATIC S7-300. Dans notre exemple, la tension de sortie du système régulé devra être maintenue constante grâce à une régulation continue. En fonction des réglages appliqués au système nous simulerons une régulation de type P et PI. Le facteur de transfert  $K_p$  ainsi que les constantes de temps sont paramétrables.

L'automate programmable est relié au système régulé avec des entrées/ sorties analogiques pour influencer sur celui-ci.

Dans cet exemple, le (S) FB 41 de l'AP SIMATIC S7-300 va être utilisé en tant que régulateur continu numérique. Son rôle sera de calculer une valeur d'ajustement  $y$  en fonction de l'erreur (différence mesure/consigne)  $e=w-x$  selon l'algorithme d'un régulateur PID, et de livrer cette grandeur d'ajustement  $y$  sur sa sortie analogique.

Pour cela, il est nécessaire de définir les paramètres de régulation suivante :

$K_p$  : coefficient de gain proportionnel

$T_i$  : temps d'intégration

$T_d$  : temps de dérivation

Dans notre travail on va identifier les paramètres de régulateur PID :

On a

Les équations temporelles de notre système :

$$C \frac{dh}{dt} = e(t) - s(t)$$

$$s(t) = \frac{1}{R}(h - u) \Rightarrow s(t) = h$$

$$\text{Avec : } R=1 ; U=0$$

E : entrée ; S : sortie ; h : hauteur ; U : niveau initial ; R : conduite parfaite ; C = volume du bac

La fonction de transfert :

$$\begin{aligned} C * H(p) &= E(p) - S(p) \quad \text{Tq } S(p) = H(p) \\ \Rightarrow S(p) &= (Cp + 1) * H(p) \\ \frac{H(p)}{S(p)} &= \frac{1}{Cp + 1} = G(p) \end{aligned}$$

$$C = 15m^3$$

La fonction de transfert en boucle fermée avec un correcteur :

$$\begin{aligned} \frac{S(p)}{E(p)} &= \frac{C(p)G(p)}{C(p)G(p) + 1} \\ \Rightarrow \frac{S(p)}{E(p)} &= \frac{C(p)}{C(p) + 15p + 1} \end{aligned}$$

On désire avoir la fonction de transfert :  $\frac{1}{sP+1}$

$$\begin{aligned} \Rightarrow C(p) &= \frac{15p + 1}{5p} \\ \Rightarrow \tau &= 5s \end{aligned}$$

En remplaçant C dans la boucle ouverte et avec l'identification avec la forme générale on trouve :  
 $K_p = 3 ; T_i = 15s ; T_d = 0s$

## 4.6 Table des mnémoniques

En programmant sur step7, nous travaillons avec des opérands tels qu'E/S, mémoires, compteurs, temporisateurs, données, fonctions. Nous pouvons les adresser de manière absolue dans le programme, mais nous pouvons aussi améliorer considérablement la lisibilité et la clarté d'un programme en utilisant des mnémoniques à la place des adresses absolues.

Programme S7(1) (Mnémoniques) -- pid\SIMATIC 300(1)\CPU 313					
	Etat	Mnémonique /	Opérande	Type de do	Commentaire
1		affichage_AL_NB	A 124.5	BOOL	
2		affichage_AL_NH	A 124.6	BOOL	
3		affichage_AL_P1	A 124.4	BOOL	
4		affichage_AL_P2	A 124.7	BOOL	
5		alarme_NB	M 1.2	BOOL	
6		alarme_NH	M 1.4	BOOL	
7		alarme_P1	M 1.0	BOOL	
8		alarme_P2	M 0.5	BOOL	
9		ArretP1	M 0.0	BOOL	
10		ArretP2	M 0.1	BOOL	
11		CONT_C	FB 41	FB 41	Continuous Control
12		CYC_INT5	OB 35	OB 35	Cyclic Interrupt 5
13		DB de FB41	DB 5	DB 5	regulateur de cuve
14		defaut_niveau	M 1.3	BOOL	
15		MarcheP1	E 124.1	BOOL	
16		MarcheP2	E 124.4	BOOL	
17		NB	E 124.3	BOOL	
18		NH	E 124.2	BOOL	
19		pompe 2	FC 2	FC 2	
20		pompe_1	A 124.0	BOOL	
21		Pompe2	A 124.1	BOOL	

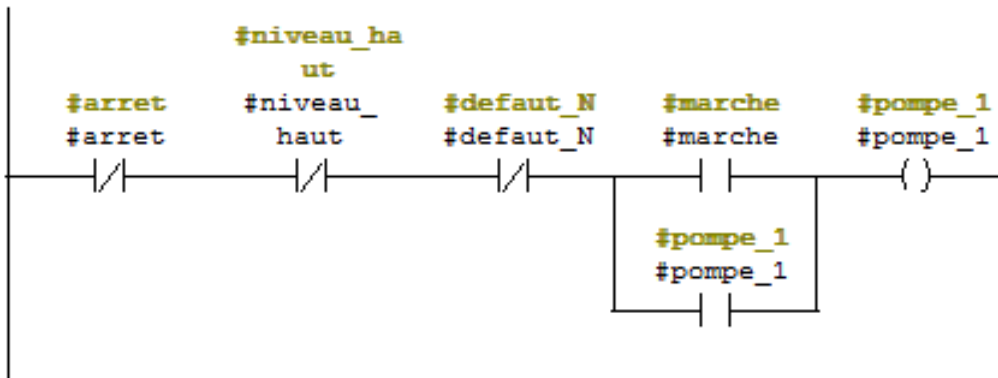
## 4.7 Programme

Pour introduire la consigne de régulation on a programmé comme suite :  
Les conditions marche/arrêt de la pompe 1, cette dernière que on cherche à réguler.

on as programmes c'est deux programme pour les deux pompe :

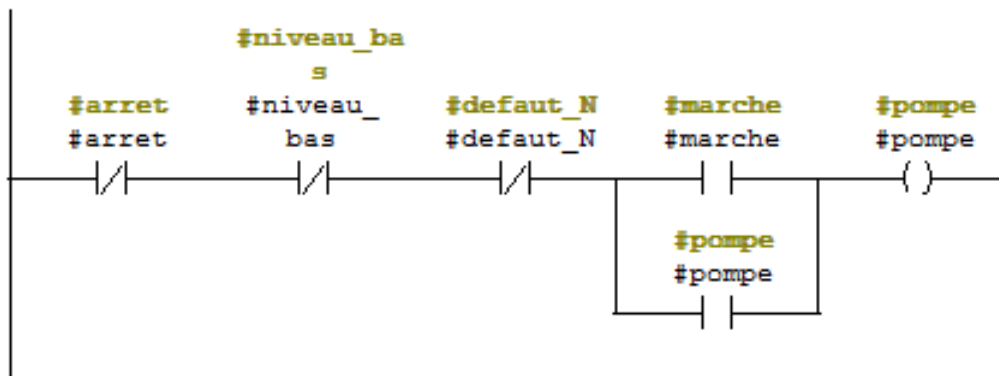
**pompe 1 :**

```
FC1 : fonction pompel
Réseau 1 : marche pompel
```



**pompe 2 :**

```
FC2 : Pompe_2
Réseau 1 : Marche/arrêt_p2
```



## 4.8 Les alarmes

On a programmé la sécurité de la pompe 1 et pompe 2 avec :

- Les alarmes de niveau haut.
- les alarmes Niveau bas.
- Le défaut de niveau.
- les alarmes des deux pompes.

Alarme de niveau :

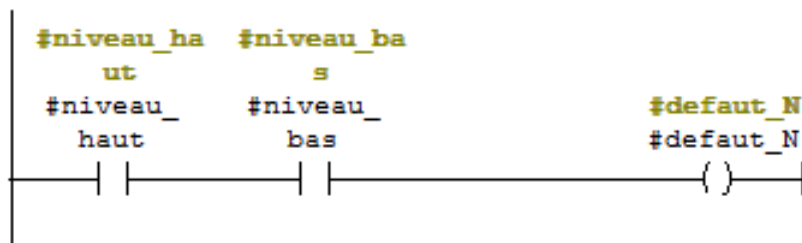
Réseau 2 : alarme niveau bas



Réseau 5 : alarme niveau haut



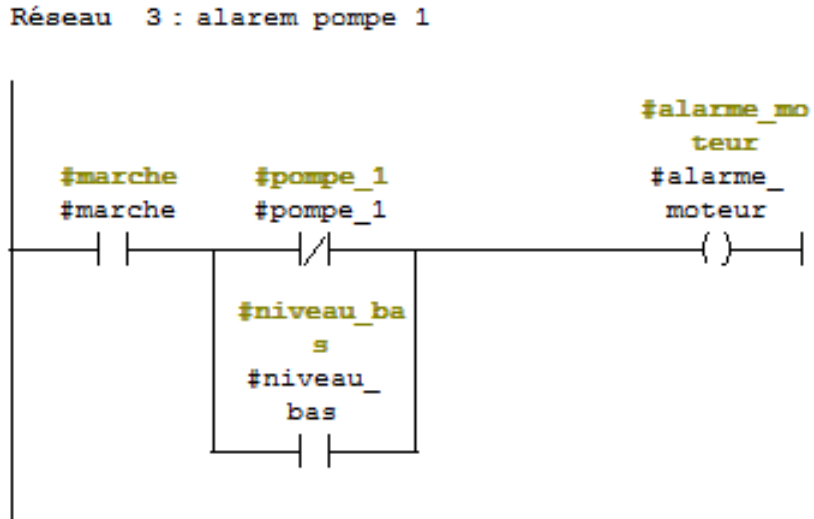
Réseau 2 : défaut\_niveau



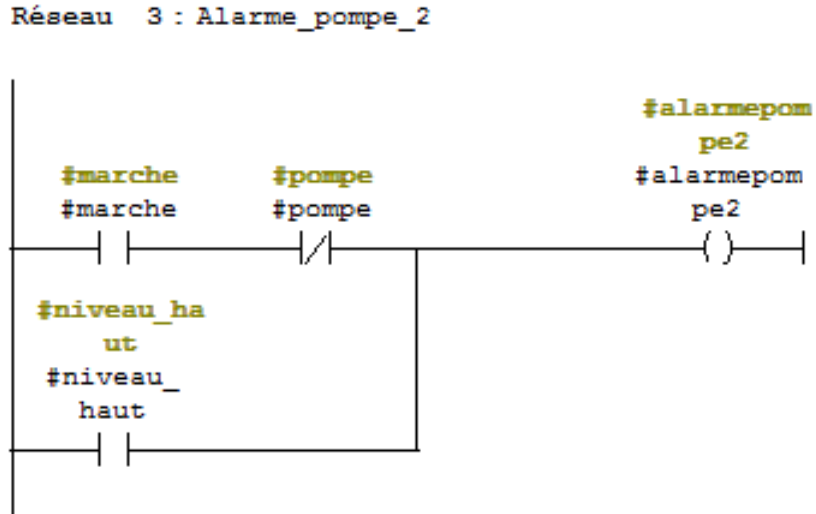


les alarmes des pompes :

Alarme pompe 1 :

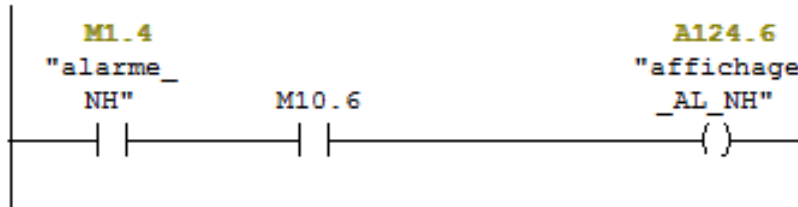


Alarme pompe 2 :

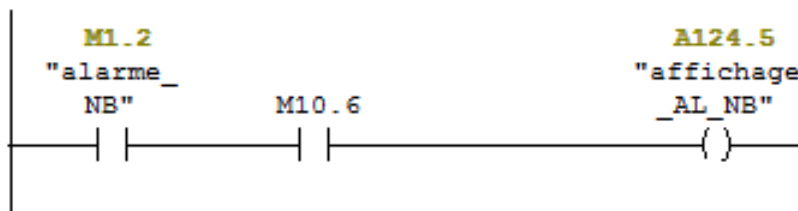


Pour le bon affichage sur Wincc (supervisions) on a réalisé le programme clignotement des alarmes à l'aide des mémento de cadences :

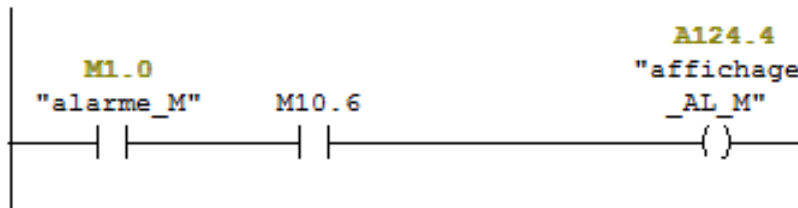
Réseau 2 : affichage alarme niveau haut



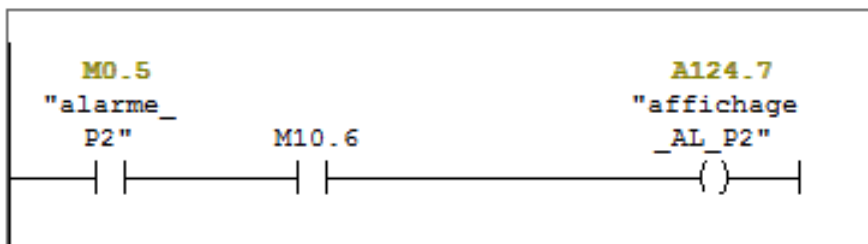
Réseau 3 : affichage alarme niveau bas



Réseau 4 : affichage alarmme POMPE 1

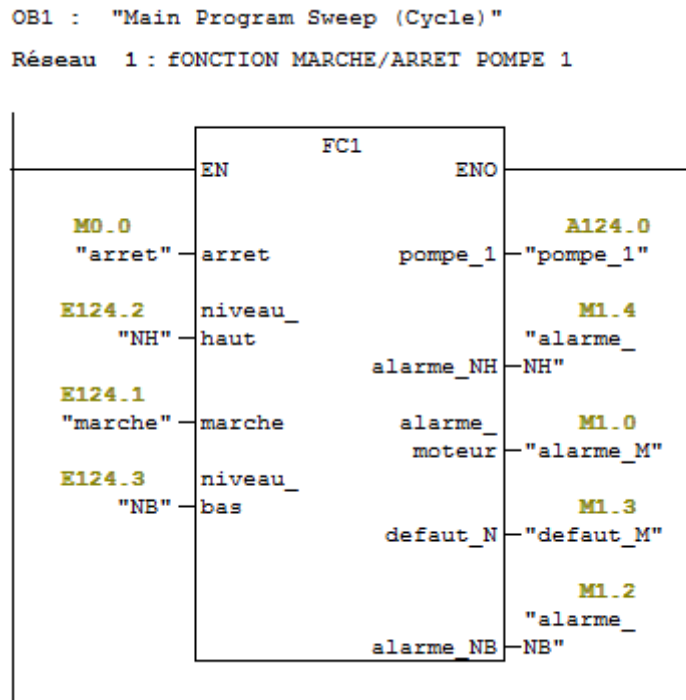


Réseau 5 : alarme pompe 2

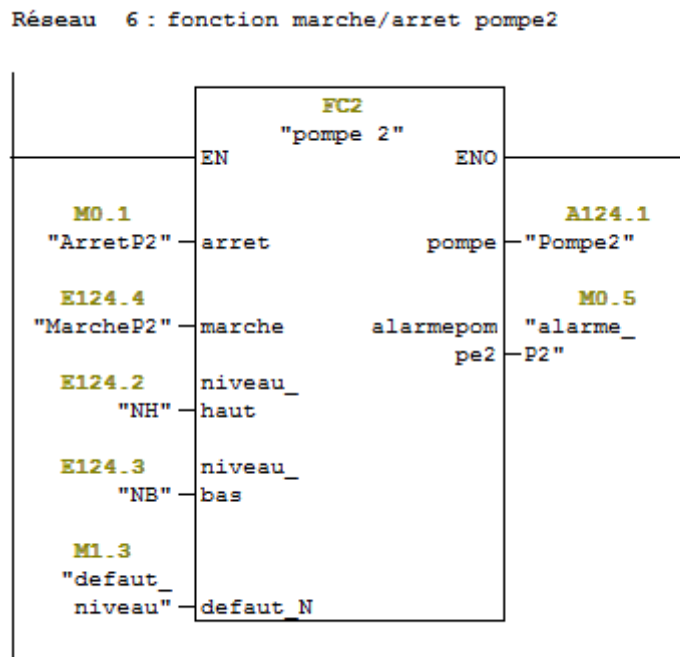


**Les fonctions :**

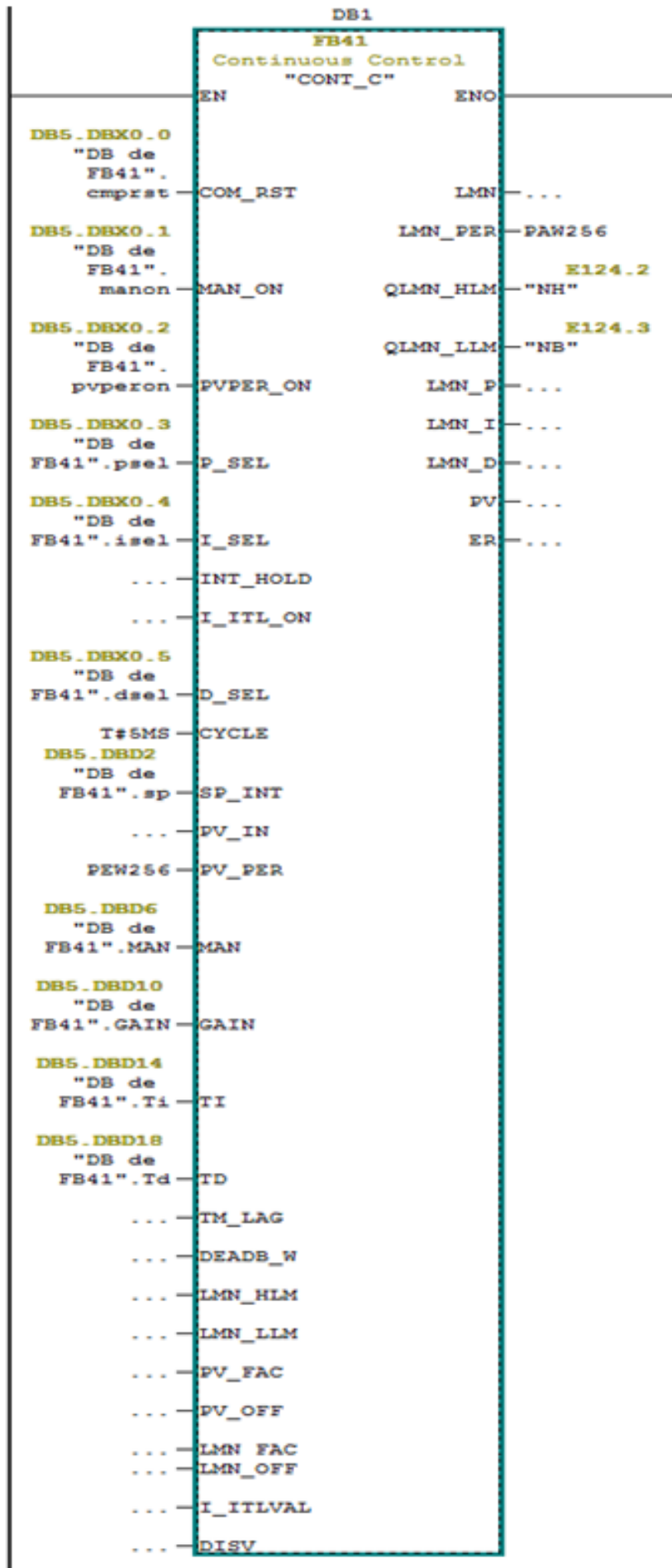
On a programmé cette fonction pour le démarrage et arrête de la pompe 1 :



On a programmé cette fonction pour le démarrage et arrête de la pompe 2 :



Après la sécurité de notre système et après le démarrage des pompes nous avons programme cette fonction FB41 du régulateur PID qui va surveillé le niveaux du bac.



Le contenu du bloc de donné d'instance pour le FB41 :

Adresse	Nom	Type	Valeur initiale	Commentaire
0.0		STRUCT		
+0.0	cmprst	BOOL	FALSE	
+0.1	manon	BOOL	FALSE	
+0.2	pvperon	BOOL	FALSE	
+0.3	psel	BOOL	FALSE	
+0.4	isel	BOOL	FALSE	
+0.5	dsel	BOOL	FALSE	
+2.0	sp	REAL	0.000000e+000	
+6.0	MAN	REAL	0.000000e+000	
+10.0	GAIN	REAL	0.000000e+000	
+14.0	Ti	TIME	T#0MS	
+18.0	Td	TIME	T#0MS	
=22.0		END_STRUCT		

Le bloc de donné en temps réel de la fonction FB41 :

	Adresse	Décl	Nom	Type	Valeur initial	Valeur en c	Commentaire
1	0.0	in	COM_...	BOOL	FALSE	FALSE	complete restart
2	0.1	in	MAN_...	BOOL	TRUE	TRUE	manual value on
3	0.2	in	PVPE...	BOOL	FALSE	FALSE	process variable peripherie on
4	0.3	in	P_SEL	BOOL	TRUE	TRUE	proportional action on
5	0.4	in	I_SEL	BOOL	TRUE	TRUE	integral action on
6	0.5	in	INT_H...	BOOL	FALSE	FALSE	integral action hold
7	0.6	in	I_ITL...	BOOL	FALSE	FALSE	initialization of the integral action
8	0.7	in	D_SEL	BOOL	FALSE	FALSE	derivative action on
9	2.0	in	CYCLE	TIME	T#1S	T#1S	sample time
10	6.0	in	SP_INT	REAL	0.000000e...	0.000000e...	internal setpoint
11	10.0	in	PV_IN	REAL	0.000000e...	0.000000e...	process variable in
12	14.0	in	PV_PER	WORD	W#16#0	W#16#0	process variable peripherie
13	16.0	in	MAN	REAL	0.000000e...	0.000000e...	manual value
14	20.0	in	GAIN	REAL	2.000000e...	2.000000e...	proportional gain
15	24.0	in	TI	TIME	T#20S	T#20S	reset time
16	28.0	in	TD	TIME	T#10S	T#10S	derivative time
17	32.0	in	TM_L...	TIME	T#2S	T#2S	time lag of the derivative action
18	36.0	in	DEAD...	REAL	0.000000e...	0.000000e...	dead band width
19	40.0	in	LMN_...	REAL	1.000000e...	1.000000e...	manipulated value high limit
20	44.0	in	LMN_...	REAL	0.000000e...	0.000000e...	manipulated value low limit
21	48.0	in	PV_F...	REAL	1.000000e...	1.000000e...	process variable factor
22	52.0	in	PV_OFF	REAL	0.000000e...	0.000000e...	process variable offset
23	56.0	in	LMN_...	REAL	1.000000e...	1.000000e...	manipulated value factor
24	60.0	in	LMN_...	REAL	0.000000e...	0.000000e...	manipulated value offset
25	64.0	in	I_ITLV...	REAL	0.000000e...	0.000000e...	initialization value of the integral action
26	68.0	in	DISV	REAL	0.000000e...	0.000000e...	disturbance variable
27	72.0	out	LMN	REAL	0.000000e...	0.000000e...	manipulated value
28	76.0	out	LMN_...	WORD	W#16#0	W#16#0	manipulated value peripherie

## 4.9 Simulation

### 4.9.1 Simulation du programme sous STEP7 avec PLCSIM :

L'application de simulation de modules S7-PLCSIM permet d'exécuter et de tester le programme dans un Automate Programmable (API) qu'on simule dans un ordinateur ou dans une console de programmation. La simulation étant complètement réalisée au sein du logiciel STEP7, il n'est pas nécessaire qu'une liaison soit établie avec un matériel S7 quelconque (CPU ou module de signaux). S7-PLCSIM dispose d'une interface simple permettant de visualiser et de forcer les différents paramètres utilisés par le programme. Et de ce fait on peut corriger des erreurs qui peuvent exister sans risquer de causer des dommages sur la machine.

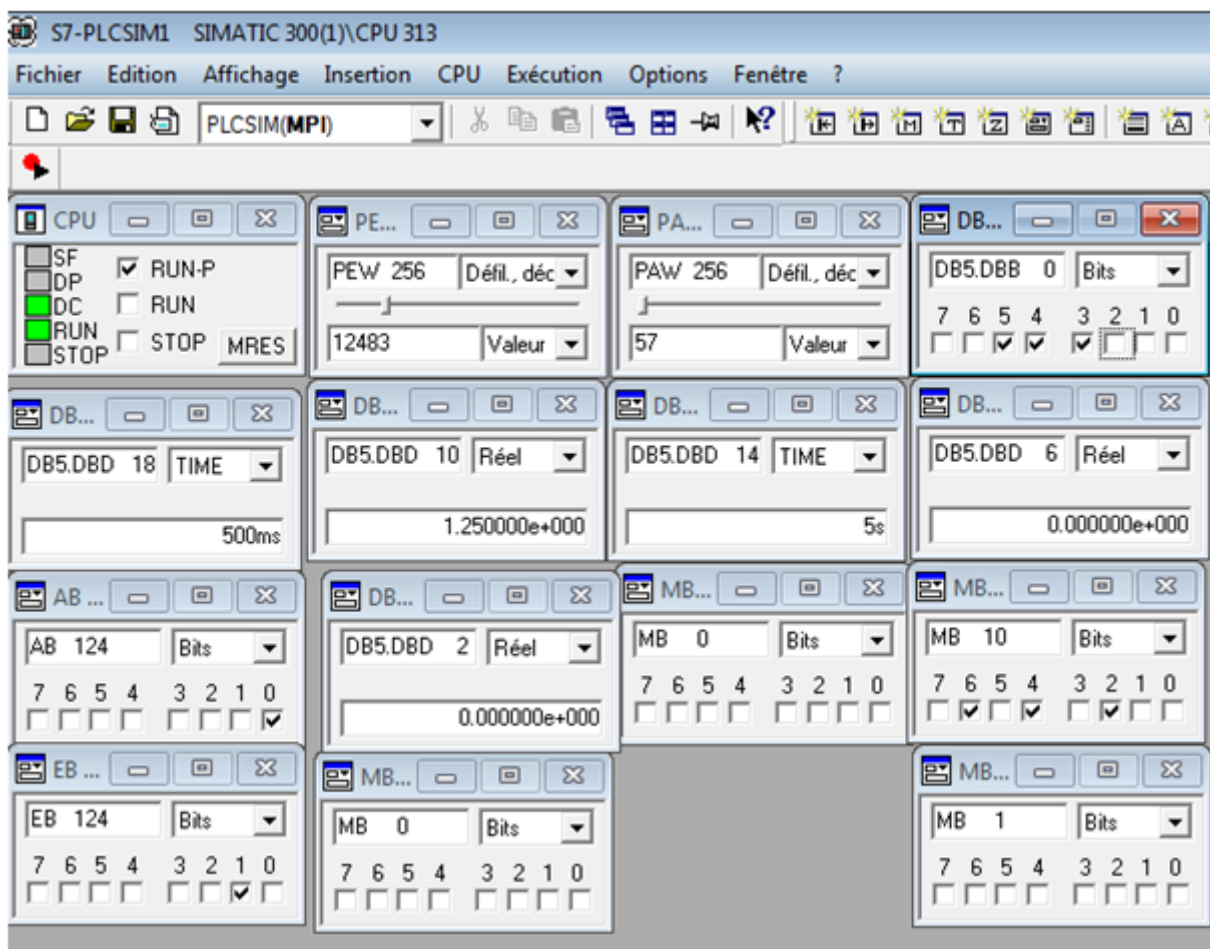
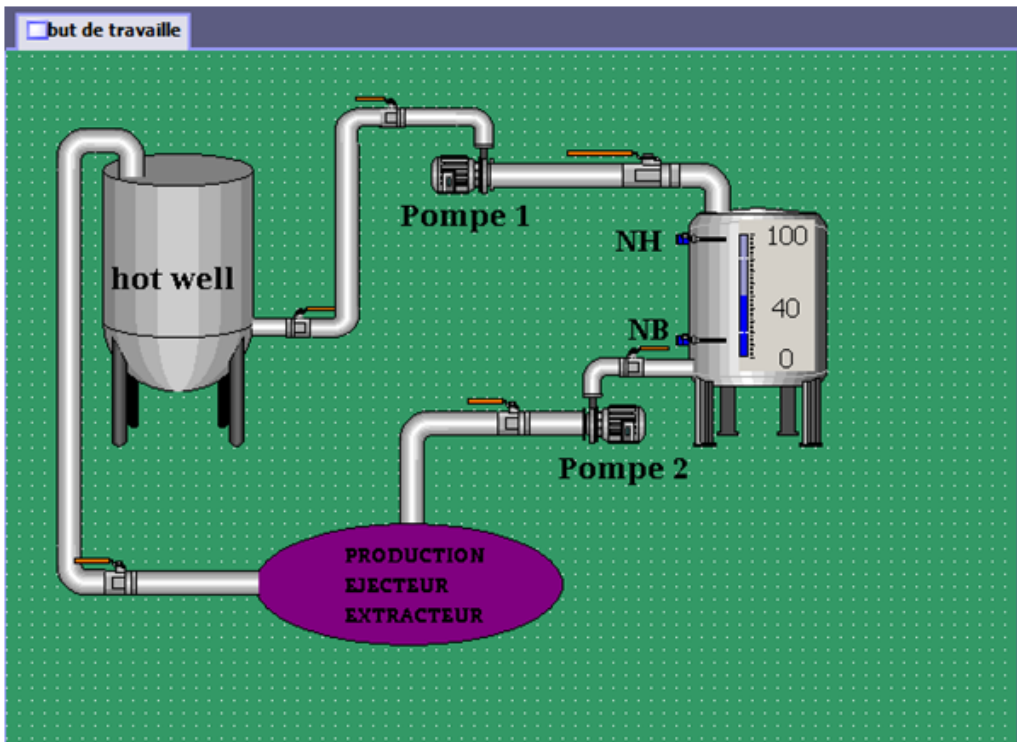


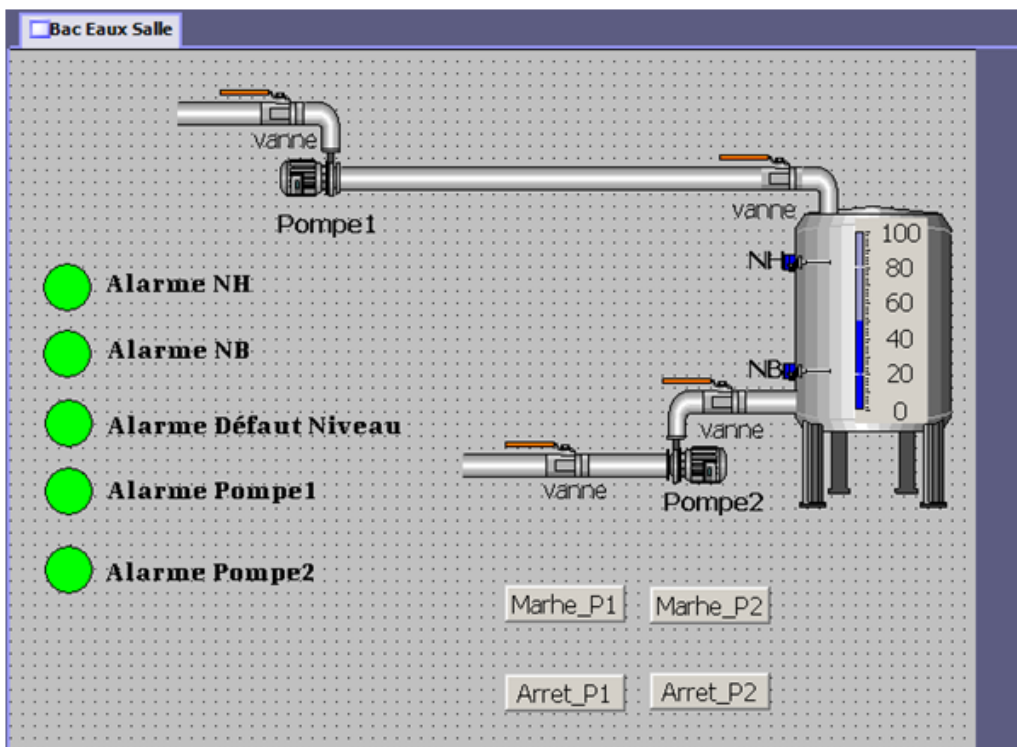
FIGURE 4.5 – Simulateur S7-PLCSIM

4.9.2 Simulation du programme sous WinCC :

Système global :



Vue du système :



## 4.10 Conclusion

Dans ce chapitre nous avons présenté la solution apportée en deux parties, la partie matérielle et la partie logicielle. A la fin on a présenté la simulation de ce système sous formes d'images dans le logiciel WinCC.

Les résultats que nous avons trouvés sont concluant, à savoir une meilleur régulation de niveaux évitant ainsi tous les inconvénients de la régulation tout ou rien TOR.



# CONCLUSION GÉNÉRALE

---

Le stage effectué au sein du complexe CEVITAL nous a permis de mettre en évidence l'aspect pratique de l'ensemble des connaissances théoriques acquises tout au long de notre cursus, ainsi enrichir nos connaissances et notre savoir-faire dans le domaine de la conception électronique et l'automatisation des systèmes industriels et tirer profit de l'expérience des personnes du domaine et d'autre part, d'apprendre une méthodologie rationnelle à suivre pour l'élaboration des projets d'automatisation.

Nous avons présenté notre projet qui consiste en l'automatisation et la régulation de niveau d'eau d'un bac. Pour ce faire ; nous avons procédé de la manière suivante :  
Tout d'abord, nous avons fait l'étude du système déjà existant afin de déterminer les problèmes et proposer ensuite les solutions.  
En effet la pompe fonctionnait en semi-automatique, avec une automatisation de quelques parties basée sur une logique câblée.

Après, et suivant les informations récoltées de la première étape, nous avons proposés la nouvelle solution d'automatisation du système. Cette solution est basée autour d'un automate programmable SIEMENCE de type S7-313. Le choix de la CPU ainsi que les différents modules a été effectué selon les besoin de la machine. La prochaine étape c'est d'élaborer un programme pour le variateur qui commande la pompe centrifuge. Ce programme a été réalisé a l'aide du logiciel Step7 en langages CONT-C.

La validation de ce programme a été faite sous forme de simulations sur un automate virtuel.les résultats des simulations ont été concluants et ont montrés que le programme est bien fonctionnel et peut être appliqué sur le système réel.  
Nous espérons avoir apporté une solution au problème posé tout en étant conscients que des améliorations peuvent être apportées par les promotions a venir.

# ANNEXE 1

---

## Présentation du complexe CEVITAL de Bejaia Historique

CEVITAL est parmi les entreprises algériennes qui ont vu le jour dès l'entrée de notre pays en économie du marché. Elle a été créée par des fonds privés en 1998. CEVITAL contribue largement au développement de l'industrie agroalimentaire nationale, elle vise à satisfaire le marché national et à exporter le surplus, en offrant une large gamme de produits de qualité. En effet, les besoins du marché national sont de 1200 T/J d'huile ou l'équivalent de 12 litres par personne et par an. Les capacités actuelles de CEVITAL sont de 1800 T/j, soit un excédent commercial de 600T/J. Les nouvelles données économiques nationales dans le marché de l'agroalimentaire font que les meilleurs sont ceux qui maîtrisent d'une façon efficace et optimale les coûts, les charges et ceux qui offrent le meilleur rapport qualité prix. Ceci est nécessaire pour s'imposer sur le marché que CEVITAL négocie avec les grandes sociétés commerciales internationales, ces produits se vendent dans différentes villes africaines (Lagos, Niamey, Bamako, Tunis, Tripoli...)[17].

### Situation géographique

CEVITAL est un complexe de production qui se situe au niveau du nouveau quai du port de Bejaia à 3 Km du sud-ouest de cette ville et s'étend sur une superficie de 45000 m<sup>2</sup>, à proximité de la route nationale N 09. Cette situation géographique de l'entreprise lui a beaucoup profité étant donné qu'elle lui confère l'avantage de proximité économique. Elle se trouve proche du port et de l'aéroport.

### Activité et objectif du CEVITAL

L'entreprise produit essentiellement des huiles végétales et du sucre, elle assure le conditionnement de ses produits selon des normes internationales. Malgré la qualité de ses produits CEVITAL les propose en qualité suffisante à des prix compétitifs.

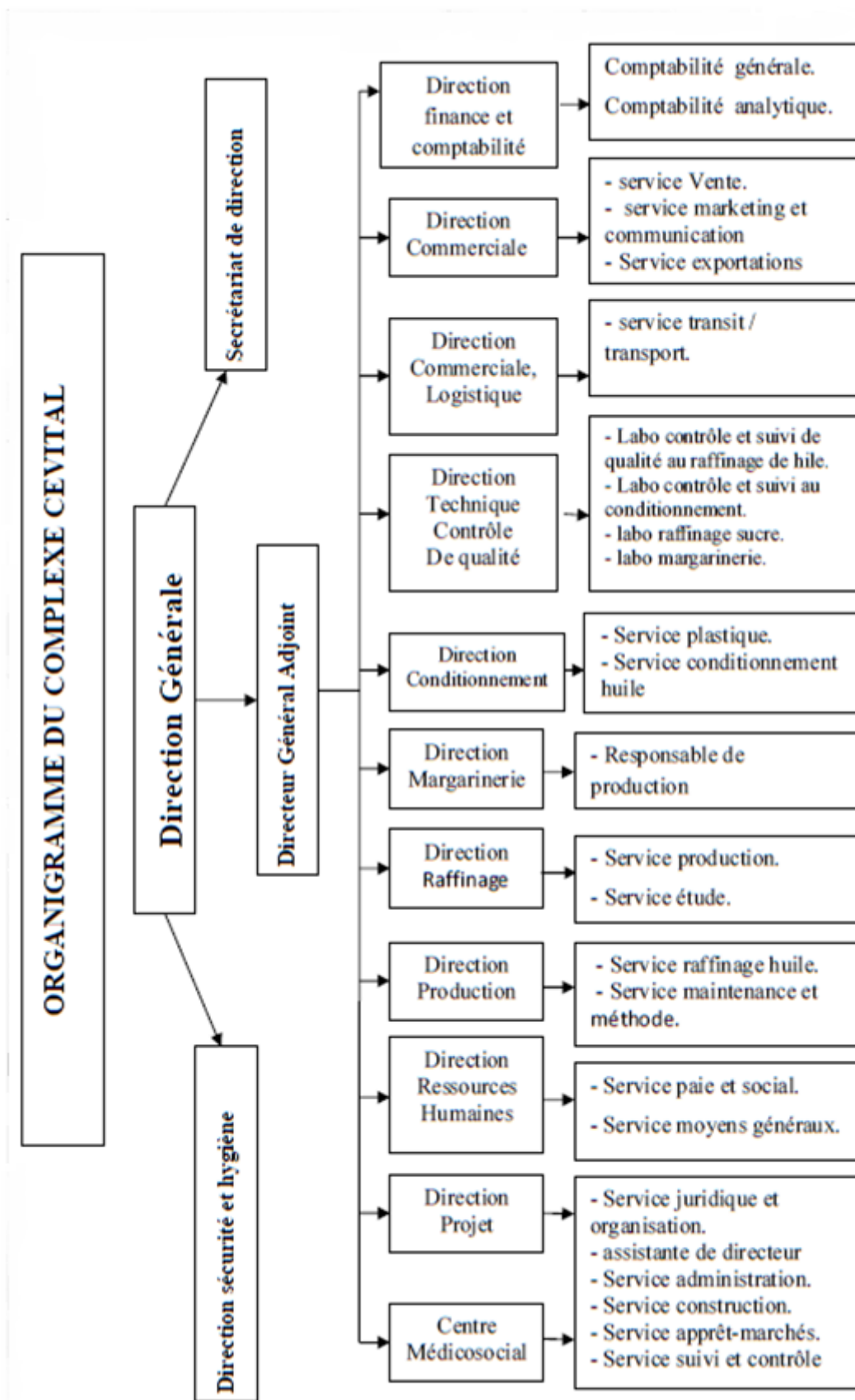
Sa production par jour est capitulée comme suite :

- Raffinage d'huiles (1800 tonnes /jour)
- Conditionnement d'huile (1400 tonnes/jour)
- Production de la margarine (600 tonnes/jour)
- Fabrication d'emballage (PET) : poly-Ethylène-Téréphtalate (9600 unités/heure)
- Raffinage du sucre (1600 tonnes/jour et 3000 tonnes/jour)
- Stockage des céréales (120000 tonnes)
- La cogénération (une capacité de production arrive jusqu'à 64MW)
- Minoterie et savonnerie en cour d'étude[17]

### Structure du complexe CEVITAL

Pour présenter le complexe agroalimentaire CEVITAL d'une manière générale, la figure ci-dessous relate la structure hiérarchique du complexe ainsi que les différentes directions des services[17].

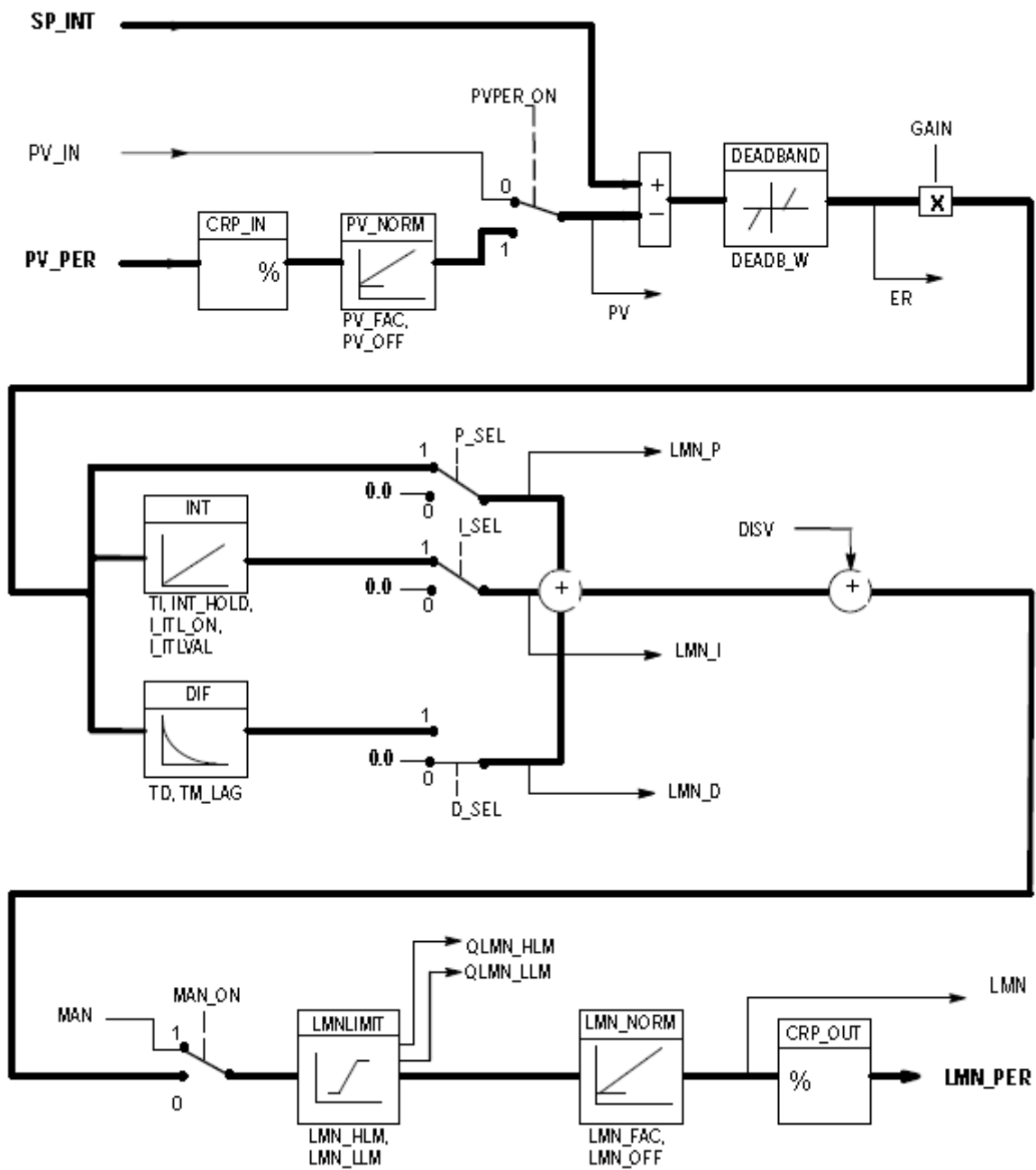
# ANNEXE 1





# ANNEXE 3

Le schéma fonctionnel d'un PID :[7]



## ANNEXE 3

paramètres d'entrée (INPUT) du bloc FB 41 [7]

Paramètre	Type de données	Valeurs admises	Par défaut	Description
COM_RST	BOOL		FALSE	COMPLETE RESTART / Démarrage Le bloc renferme un sous-programme de démarrage qui est exécuté quand cette entrée est à 1.
MAN_ON	BOOL		TRUE	MANUAL VALUE ON / Activation du mode manuel Quand cette entrée est à 1, la boucle de régulation est interrompue. La valeur de réglage manuelle est sortie comme grandeur de réglage.
PVPER_ON	BOOL		FALSE	PROCESS VARIABLE PERIPHERY ON / Activation de la mesure de périphérie Pour que la mesure soit lue en périphérie, il faut relier l'entrée PV_PER à la périphérie et mettre à 1 l'entrée PVPER_ON.
P_SEL	BOOL		TRUE	PROPORTIONAL ACTION ON / Activation de l'action proportionnelle Dans l'algorithme PID, il est possible d'activer et de désactiver séparément chacune des actions. L'action P est active quand cette entrée est à 1.
I_SEL	BOOL		TRUE	INTEGRAL ACTION ON / Activation de l'action par intégration Dans l'algorithme PID, il est possible d'activer et de désactiver séparément chacune des actions. L'action I est active quand cette entrée est à 1.
INT_HOLD	BOOL		FALSE	INTEGRAL ACTION HOLD / Gel de l'action par intégration La sortie de l'intégrateur peut être gelée. Pour cela, il faut mettre à 1 cette entrée.
I_ITL_ON	BOOL		FALSE	INITIALIZATION OF THE INTEGRAL ACTION / Initialisation de l'action par intégration La sortie de l'intégrateur peut être positionnée sur la valeur initiale I_ITL_VAL. Pour cela, il faut mettre à 1 cette entrée.
D_SEL	BOOL		FALSE	DERIVATIVE ACTION ON / Activation de l'action par dérivation Dans l'algorithme PID, il est possible d'activer et de désactiver séparément chacune des actions. L'action D est active quand cette entrée est à 1.
CYCLE	TIME	$\geq 1$ ms	T#1s	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	-100,0 à 100,0 (%) ou grandeur physique <sup>1</sup>	0,0	INTERNAL SETPOINT / Consigne interne Cette entrée sert à introduire une valeur de consigne.
PV_IN	REAL	-100,0 à 100,0 (%) ou grandeur physique <sup>1</sup>	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.

## ANNEXE 3

paramètres d'entrée (INPUT) du bloc FB 41 (suite) [7]

Paramètre	Type de données	Valeurs admises	Par défaut	Description
PV_PER	WORD		W#16#0000	PROCESS VARIABLE PERIPHERIE / Mesure de périphérie La mesure en format de périphérie est appliquée au régulateur par cette entrée.
MAN	REAL	-100,0 à 100,0 (%) ou grandeur physique <sup>2</sup>	0,0	MANUAL VALUE / Valeur de réglage manuelle Cette entrée sert à introduire une valeur de réglage manuelle moyennant des fonctions de contrôle-commande.
GAIN	REAL		2,0	PROPORTIONAL GAIN / Coefficient d'action proportionnelle Cette entrée indique le gain du régulateur.
TI	TIME	>= CYCLE	T#20 s	RESET TIME / Temps d'intégration Cette entrée détermine la réponse temporelle de l'intégrateur.
TD	TIME	>= CYCLE	T#10 s	DERIVATIVE TIME / Temps de dérivation Cette entrée détermine la réponse temporelle de l'unité de dérivation.
TM_LAG	TIME	>= CYCLE/2	T#2 s	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	>= 0,0 (%) ou grandeur physique <sup>1</sup>	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 100,0 (%) ou grandeur physique <sup>2</sup>	100,0	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	-100,0 à LMN_HLM (%) ou grandeur physique <sup>2</sup>	0,0	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		1,0	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		0,0	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		1,0	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.

## ANNEXE 3

*paramètres d'entrée (INPUT) du bloc FB 41 (suite) [7]*

Paramètre	Type de données	Valeurs admises	Par défaut	Description
LMN_OFF	REAL		0,0	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	-100,0 à 100,0 (%) ou grandeur physique <sup>2</sup>	0,0	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	-100,0 à 100,0 (%) ou grandeur physique <sup>2</sup>	0,0	DISTURBANCE VARIABLE / Grandeur perturbatrice La grandeur perturbatrice est appliquée à cette entrée pour l'action anticipatrice.

*Paramètres de sortie (OUTPUT) du bloc FB 41 [7]*

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.



## ANNEXE 3

---

*paramètres de sortie (output) du bloc FB 41 (suite) [7]*

Paramètre	Type de données	Valeurs admises	Par défaut	Description
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 Cette sortie donne le signal d'erreur effectif.

# Bibliographie

### BIBLIOGRAPHIE

[1] SEKHSOKH Soumya, OUKILI Kawthar « ÉTUDE D'UNE BOUCLE DE RÉGULATION DE NIVEAU : IMPLÉMENTATION DU RÉGULATEUR ET RÉGLAGE DU PROCÈDE » Projet de fin d'étude, 2010/2011.

[2] « Cours Automatique » École polytechnique fédérale de Lausanne, 2008/2009.

[3] MOUADH Sakli « Régulation Industrielle de Processus, Système de Régulation de Niveau d'eau, Interface à base de microprocesseur PIC 16F877, Commande & Régulation avec LabVIEW » Projet automatisme de fin d'étude, 2006/2007.

[4] IKHELEF Mohamed, LUIBA Yanis « automatisation et régulation d'une pompe par débitmètre »Raport de projet fin d'étude, 2013/2014.

[5] MAGAROTTO Eric « Cour de régulation » Université de Caen, 2004/2005.

[6] GONZAGA Alain « Les automates programmables industriels » Novembre, 2004.

[7] Documentation sur cd Simatic « Wincc Control Center » ref : C79000-G7077-C516-01.

[8] KADRI Ahmed Yacine « Cours Régulation Automatique » Université OUARGLA, 2013/2014.

[9] Andre Simon « Automates programmable industriels niveau 1 » Edition L'ELAN-LIEGE 1991.

[10] JP VISKOVIC « Support Omron » ref : STA37-V3, 25/10/2007

[11] L.BERGOUGNOUX « Automates Programmables Industriels » Polytech Marseille Département de Mécanique Énergétique, 2004/2005.

[12] SIMATIC « Mise en route STEP7 » Édition 03 /2006.

[13] J-M Bleux. J-L Fanchon « Automatisme industriel », 2005.

## BIBLIOGRAPHIE

---

[14] ANDJOUH Lyes, TOUATI Riad « Automatisation et supervision de la fosse de relevage de la raffinerie d'huile au niveau du complexe agroalimentaire cevital» Mémoire de fin d'étude, 2012/2013.

[15] «SIMATIC WinCC flexible», Brochure, Mars 2010.

[16] SIMATIC IHM WinCC flexible «Getting started debutants», 2010.

### **Site web**

[17] [www.cevital.dz](http://www.cevital.dz)

## RÉSUMÉ

---

Les automates programmables industriels représentent l'élément important de la chaîne automatisée, car ils assurent une meilleure flexibilité et facilitent la maintenance.

Ce présent travail est réalisé au sein de l'entreprise CEVITAL, avec l'équipe maintenance raffinerie d'huile.

Le but de notre travail consiste à automatiser une partie de l'usine en mettant au point une régulation de niveau avec un système de fonctionnement commandé par un automate programmable industriel API Siemens S7-300 et nous allons intégrer une nouvelle technique de surveillance qui est la supervision.

La réalisation de ce projet nous a permis de mettre en évidence l'aspect pratique de nos connaissances théoriques acquises, ainsi enrichir nos connaissances et notre savoir-faire dans le domaine de la conception et l'automatisation des systèmes industriels.

### **Abstract**

The industrial programmable automats represent the important component of the automated chain, because they ensure a better flexibility and facilitate maintenance.

The goal of our work consists in automating part of the factory by developing a regulation of level with a system of operation ordered by an industrial programmable automat API Siemens S7-300 and we will integrate a novel method of monitoring which is the supervision.

This present work is completed within company CEVITAL, with the team maintenance refinery of oil.

The realization of this project enabled us to highlight the practical aspect of our acquired theoretical knowledge, thus to enrich our knowledge and our know-how in the field by the design and the automation of the industrial systems.