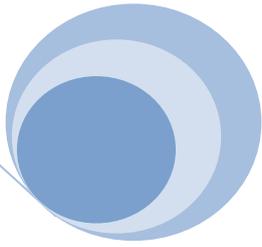


République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Abderrahmane Mira de Bejaia
Faculté de technologie
Département électronique



جامعة بجاية
Tasdawit n'Bgayet
Université de Béjaïa

Mémoire De Fin d'Etude

En vue de l'obtention d'un diplôme Master en électronique

Option : Automatique

Thème

Etude et Supervision de deux bacs d'huile brute
CEVITAL-BEJAIA

Réalisé par :

M^{elle} ARRAD Fadila

M^{elle} BOUHAMOU Fatima

Encadré par :

M^{me} N.BELLAHSENE

M^r S.KHIMA

Promotion

2014/2015

Remerciements

Avant tout, nous remercions ALLAH tout puissant de nous avoir donnés la volonté, la santé et la patience durant toutes ces longues années pour réaliser ce modeste travail.

*On tient à exprimer notre profonde gratitude à notre promotrice **M^{me} N.BELLAHSENE**, pour avoir accepté de diriger ce travail son aide et surtout pour tous ses conseils précieux.*

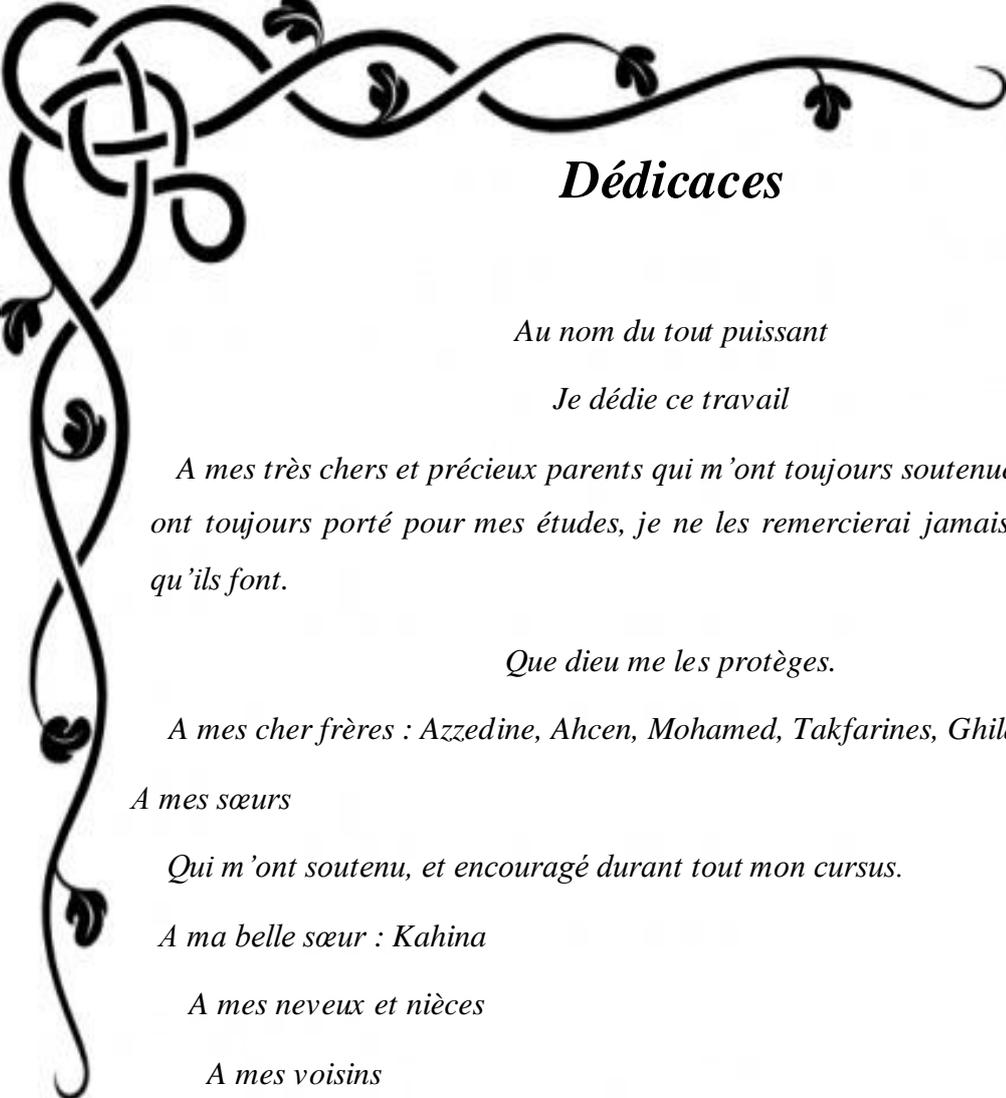
*Nos sincères remerciements, les plus vifs s'adressent à notre co-promoteur **Mr S.KHIMA** pour sa disponibilité entière, son aide et pour avoir mis à notre disposition les moyens logistique et pour son implication dans notre projet.*

*Ainsi que tous le personnel de la direction technique de **CEVITAL** (unité raffinerie d'huile), particulièrement **Mr RACHID** pour son aide.*

Nous tenons aussi à remercier le président et les membres du jury d'avoir accepté de juger notre travail.

Nos remerciement vont également à tous ceux qui nous ont aidé et conseillé de près ou de loin au long de nos études jusqu'à la finalisation de ce projet.





Dédicaces

Au nom du tout puissant

Je dédie ce travail

A mes très chers et précieux parents qui m'ont toujours soutenue, et à l'intérêt qu'ils ont toujours porté pour mes études, je ne les remercierai jamais assez, pour tout ce qu'ils font.

Que dieu me les protèges.

A mes cher frères : Azzedine, Ahcen, Mohamed, Takfarines, Ghilas, Rabah.

A mes sœurs

Qui m'ont soutenu, et encouragé durant tout mon cursus.

A ma belle sœur : Kahina

A mes neveux et nièces

A mes voisins

A ma cher binôme : Bouhamou Fatima et ses parents

A toutes mes copines de chambre D204

Et tous mes amis

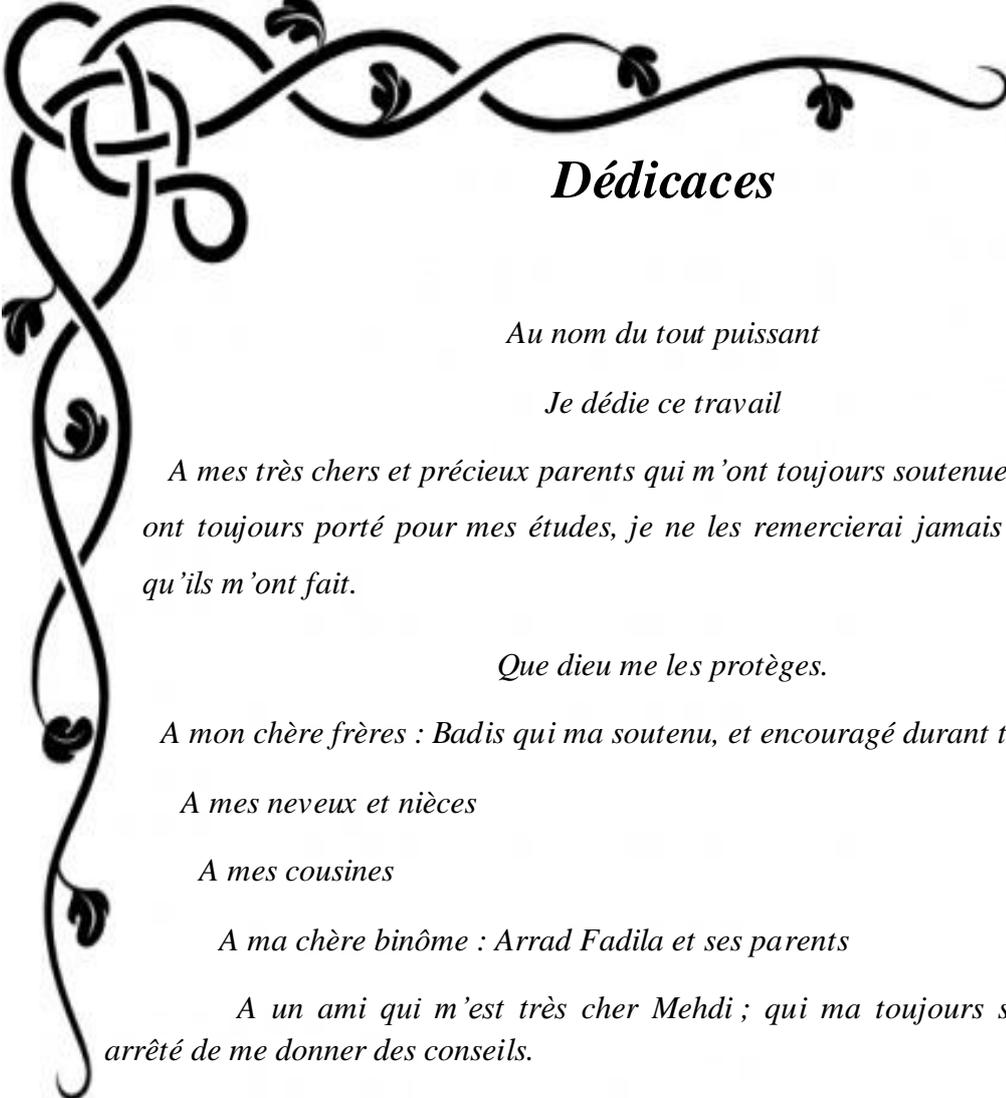
Je tiens à remercier tous les étudiants de la promotion 2014/2015 AUTOMATIQUE.

En fin à toute personne qui m'est cher au cœur.

Que dieu nous rassemble tous au paradis

FADILA





Dédicaces

Au nom du tout puissant

Je dédie ce travail

A mes très chers et précieux parents qui m'ont toujours soutenue, et à l'intérêt qu'ils ont toujours porté pour mes études, je ne les remercierai jamais assez, pour tout ce qu'ils m'ont fait.

Que dieu me les protèges.

A mon chère frères : Badis qui ma soutenu, et encouragé durant tout mon cursus.

A mes neveux et nièces

A mes cousines

A ma chère binôme : Arrad Fadila et ses parents

A un ami qui m'est très cher Mehdi ; qui ma toujours soutenue et n'a pas arrêté de me donner des conseils.

Et toutes mes familles et mes amis

Je tiens à remercier tous les étudiants de la promotion 2014/2015 AUTOMATIQUE.

En fin à toute personne qui m'est cher au cœur.

Que dieu nous rassemble tous au paradis

FATIMA



SOMMAIRE

Liste des figures.....	1
Liste des abréviations et symboles	1
Introduction générale	1

CHAPITRE I : Présentation et Etude du processus

I Introduction	3
I.1 Objectifs du raffinage.....	3
I.2 Les différentes étapes de production.....	3
I.3 Les sous-unités de la raffinerie d'huile	4
I.4 Structure de la raffinerie d'huile	5
I.5 Etapes de raffinage d'huile.....	6
I.5.1 Les procédés industriels de la raffinerie.....	7
I.5.1.1 Définition.....	7
I.5.1.2 Les différents procédés	7
I.6 La partie électrique.....	8
I.6.1 Introduction	8
I.6.2 L'appareillage.....	8
I.7 Présentation du Projet	11
I.8 Présentation de l'état des deux bacs d'huiles brutes	11
I.8.1 Instrumentation des deux bacs G et H.....	12
I.9 Analyse fonctionnelle des deux bacs G et H.....	17
I.9.1 processus de remplissage d'huiles des bacs	17
I.9.2 processus d'extraction d'huiles des bacs	19
I.10 Conclusion.....	20

CHAPITRE II : Automates API et supervision WinCC

II Introduction.....	21
II.1 Système automatisé.....	21
II.1.1 Objectifs de l'automatisation	21
II.1.2 Structure d'un système automatisé	21
II.2 Généralité sur les automates programmables API.....	22
II.2.1 Définition	22
II.2.2 Architectures des automates.....	22
II.2.2.1 Structure externe.....	22
II.2.2.2 Structure interne	24
II.2.3 Cycle de l'automate programmable	25
II.2.4 Critère de choix d'un automate	25
II.2.5 Nature des informations traitées par l'automate	26
II.3 Présentation de l'automate utilisé	26
II.3.1 Module de fonction	26
II.3.2 Module de communication.....	27
II.4 Caractéristique de la CPU	27
II.4.1 Interface MPI	27

II.4.2	Signification des états de l'automate	27
II.4.3	Commutateur de mode de l'automate	28
II.5	Programmation sous STEP7	29
II.5.1	Aperçu du progiciel STEP7	29
II.5.2	Création d'un projet sous STEP7	29
II.5.3	Langages de programmations STEP7	30
II.5.3.1	Programmation à schémas logique(LOG)	30
II.5.3.2	Programmation à schéma contacte(CONT)	31
II.5.3.3	Programmation à liste d'instruction(LIST)	31
II.5.4	Structures du programme	31
II.5.4.1	Programmation linéaire	31
II.5.4.2	Programmation structurée	31
II.5.5	Type de blocs dans le programme utilisateur sous STEP7	32
II.5.5.1	Les blocs d'organisation.....	32
II.5.5.2	Blocs d'organisation OB	32
II.5.5.3	Les blocs fonctionnels (FB)	33
II.5.5.4	Les bloc FC	33
II.5.5.5	Les blocs de données DB	33
II.5.5.6	Blocs de données d'instance.....	34
II.5.5.7	Blocs de données globaux	34
II.5.6	Les mémentos	34
II.6	Simulation du programme sous STEP7 avec S7-PLCSIM.....	34
II.6.1	Présentation de S7-PLCSIM	34
II.6.2	Caractéristique de S7-PLCSIM.....	35
II.7	Généralité sur la supervision	35
II.7.1	Définition	35
II.7.2	Avantage de la supervision	36
II.8	Présentation du logiciel de supervision WinCC flexible	36
II.8.1	Introduction	36
II.8.2	Description du logiciel WinCC flexible.....	37
II.8.3	Eléments du WinCC flexible	37
II.8.4	WinCC flexible Runtime	38
II.8.5	Système graphique	39
II.8.5.1	Editeur graphique	39
II.8.5.2	Bibliothèque.....	39
II.8.6	Les liaison	39
II.8.7	Création des vues	41
II.8.8	Les variables	41
II.8.9	Alarmes systèmes	42
II.9	Conclusion	43

CHAPITRE III : Résultats et interprétation

III.	Introduction	44
III.1	Résultats pratique	44

III.1.1 Structure du GRAFCET	44
III.1.2 Configuration matérielle	46
III.1.3 Structure du programme élaboré	47
III.2 Simulation du programme élaboré	59
III.2.1 Ouverture de l'AP S7-PLCSIM	59
III.2.2 Chargement de programme	60
III.2.3 Configuration de l'API de simulation.....	60
III.2.3.1 Création des fenêtres pour l'exemple de programme	60
III.3 Simulation du projet	61
III.4 Conclusion.....	64
Conclusion Générale	65

LISTE DES FIGURES

Figure I.1 Organigramme des lignes A et B.....	5
Figure I.2 Organigramme de la ligne C.....	5
Figure I.3 Diagramme de raffinage	6
Figure I.4 Armoire électrique	8
Figure I.5 Variateur de vitesse	9
Figure I.6 Disjoncteur.....	9
Figure I.7 Contacteur.....	10
Figure I.8 Fusible	10
Figure I.9 Manomètre	10
Figure I.10 Structure de production d'huile	11
Figure I.11 Sonde de température	12
Figure I.12 Transmetteur de niveau	12
Figure I.13 Schéma électrique du Transmetteur de Niveau	13
Figure I.14 Capteur de niveau	13
Figure I.15 Schéma électrique capteur de niveau.....	14
Figure I.16 Vanne TOR.....	14
Figure I.17 Schéma électrique de la vanne TOR.....	14
Figure I.18 Agitateur	15
Figure I.19 Schéma électrique agitateur	15
Figure I.20 Moteur Pompe	15
Figure I.21 Schéma électrique d'un Moteur Pompe.....	17
Figure I.22 Diagramme de remplissage	18
Figure I.23 Diagramme d'extraction d'huile.....	19
Figure II.1 Structure d'un système automatisé	21
Figure II.2 Automate Compact.....	23
Figure II.3 Automate Modulaire(Siémens).....	23
Figure II.4 Structure interne d'un automate	24
Figure II.5 Représentation de la CPU S7-300.....	27
Figure II.6 Organigramme de création du projet	30
Figure II.7 Interface logicielle de Win CC.....	38
Figure II.8 Fenêtre des outils	39

Figure II.9 L'éditeur de liaison	40
Figure II.10 Création d'une vue	41
Figure II.11 L'éditeur variable	42
Figure II.12 Fenêtres des erreurs et avertissements	43
Figure III.1 Structure du GRACET	44
Figure III.2 GRAFCET	45
Figure III.3 Configuration matérielle.....	47
Figure III.4 Les programmes principaux dans l'OB1	47
Figure III.5 Programme du transmetteur de niveau LT	49
Figure III.6 Programme du transmetteur de température TIC	50
Figure III.7 Programme du capteur de niveau	51
Figure III.8 Programme choix du bac et pompe	52
Figure III.9 Programme contrôle vanne	53
Figure III.10 Programme appel vanne	54
Figure III.11 Programme contrôle agitateur et pompe.....	55
Figure III.12 Programme appel agitateur	56
Figure III.13 Programme appel pompe	57
Figure III.14 Table des blocs de donnée DB1	58
Figure III.15 Table des blocs de donnée DB2	59
Figure III.16 Ouverture de l'API de simulation.....	59
Figure III.17 Fenêtre de programme simulé	61
Figure III.18 Simulation du projet	63

Liste des abréviations et symboles

API : Automate Programmable Industriel

CONT : Schéma à contacte

CP : Processeur de Communication

CPU : Computer Access Memory

DB : Bloc de Donnée

E/S : Entrées/Sortie

EEPROM:

FB : Bloc Fonctionnel

FC : Fonction

FM : Module de Fonction

GRAFCET : GRAF de Commande Etapes-Transition

HMI : Interface Homme Machine

KW : Kilo Watts

LED : Diode Electro Luminisante

LIST (IL) : Liste d'Instruction

LOG : Logigramme

mA : Milli Ampère

MPI: Multi Point Interface

N/A : Numérique/Analogique

NB, NM, NH : Niveau (Bas , Moyen, Haut)

OB : Bloc d'Organisation

PB : Pompe Bateau

PG, PGs: Pomp bac G

PH, PHs: Pomp bac H

PO : Partie Opérative

PROFIBUS : Protocole de communication

RAM: Random Access Memory

ROM: Read Only Memory

SM : Module de Signaux

STEP7: Logiciel de programmation et de simulation

T/J : Tonne/Jour

TGBT : Tableau Général de Basse Tension

TOR : Tout Ou Rien

tr/min : Tour/minute

VB : Vanne Bateau

VG : Vanne bac G

VG1, VG2, VGs1, VGs2: Vanne bac G (VG1 :vanne d'aspiration, VG2:vanne de refoulement)

VH: Vanne bac H

VH1, VH2, VHs1, VHs2: Vanne bac H

WinCC Flexible : Logiciel de supervision

Y/D: Etoile/Triangle

L'industrie moderne, que l'on peut qualifier d'industrie de qualité et de quantité, ne cesse d'exiger un matériel de contrôle de plus en plus performant, afin de réaliser les deux objectifs simultanément.

En effet, l'évolution rapide des nouvelles technologies a permis de contourner la plupart des difficultés rencontrées dans le monde industriel, et a fourni plusieurs possibilités pour satisfaire les exigences et les critères demandés.

L'unité de production d'huile de CEVITAL est un exemple d'automatisation des systèmes de production en Algérie. Dans tout le processus de sa fabrication, du stockage d'huile brute au conditionnement ; la majorité des différentes étapes de production d'huile est assurée par un matériel industriel automatisé où l'intervention humaine est réduite à la surveillance de différents paramètres des machines qui assurent le bon fonctionnement de la chaîne de production.

La seule tâche qui n'est pas assurée par un Automate programmable dans le processus de production d'huile finie, est la gestion des bacs de stockages des huiles brutes où se posent de nombreux problèmes dans les fonctions principales de cette tâche. Citons le remplissage des bacs journaliers et le soutirage des huiles brutes.

L'automate programmable industriel est l'organe principal de la boucle de commande placé dans un procédé industriel, en vue de le contrôler. Il a pour tâche principale, la récolte des informations relatives à l'état du système, à partir des différents capteurs via des interfaces d'entrées, et les traite pour prendre une décision ; et ainsi commander les actionneurs via ses interfaces suivant une logique de fonctionnement mise en évidence, par un programme inscrit dans la mémoire.

Née du besoin d'un outil de visualisation des processus industriels, actuellement, la supervision est un maillon de l'information totale intégrée de l'entreprise. Chaque personne, quelle que soit son niveau, peut bénéficier d'un accès direct en temps réel à toutes les données nécessaires à son travail.

La supervision industrielle, consiste à surveiller l'état de fonctionnement d'un procédé pour l'amener et le maintenir à son point de fonctionnement optimal.

L'objectif de notre travail, est de mettre à jours l'installation de stockage des huiles au niveau de la raffinerie d'huile, par l'amélioration et l'automatisation, ainsi que la mise en œuvre d'un système de supervision pour cette installation.

Pour cela nous devons effectuer l'inventaire de l'existant, étudier les caractéristiques de l'instrumentation utilisée, maîtriser le fonctionnement de l'automate et concevoir un programme de gestion automatique et de supervision.

Reparti en trois chapitres notre travail illustre bien les principales étapes de l'analyse d'un processus industriel. Il est présenté comme suit :

- ❖ **Chapitre I : Présentation et étude du processus** ; ce chapitre résume le principe de raffinage d'huile et décrit la problématique de notre thème.
- ❖ **Chapitre II : Automates API et supervision WinCC**; une étude générale sur les automates en générale ainsi que l'API choisit pour effectuer l'automatisation qui est consacrée à l'élaboration d'un système de commande en suite des généralités sur la supervision spécialement le logiciel WinCC flexible.
- ❖ **Chapitre III : Résultats et interprétation**; nous présenterons l'architecture, la configuration et la programmation Nous terminerons ce travail par la simulation.

Introduction

Nous allons étudier dans ce chapitre, l'état des bacs et leurs fonctionnement, afin d'établir une analyse fonctionnelle qui nous permettra de faire un diagnostic pour cette installation, et d'établir un programme de fonctionnement approprié, pour la bonne gestion de l'entreprise.

Cette installation fonctionne actuellement en mode manuel, donc on fait l'étude théorique du fonctionnement des deux bacs G et H pour faciliter les tâches de l'installation ; avant d'envoyer l'huile brute à la raffinerie qui représente une étape importante dans la production d'huile.

La raffinerie d'huile comporte trois chaînes de raffinage A, B et C en soft colonne. Les deux lignes de production A et B ont une capacité de plus de 400 T/J chacune (elles sont identiques) et ligne C de 1000 T/J se qui fait un total de 1800T/J.

I.1 Objectifs du raffinage

Le raffinage a pour objectif l'amélioration des caractéristiques organoleptiques:

- ❖ Réduire le niveau de la couleur.
- ❖ Réduire ou éliminer l'odeur.
- ❖ Eliminer les produits indésirables pour avoir un produit d'un bon niveau de pureté et de stabilité.
- ❖ Modifier la structure des triglycérides nécessaires.
- ❖ Préserver le contenu en vitamines.
- ❖ Minimiser les pertes en triglycérides.
- ❖ Protéger les triglycérides de la dégradation au cours du procès.

I.2 Les différentes étapes de production

La capacité de production de la raffinerie est de 1800T /J ; cette raffinerie est conçue pour traiter toutes les qualités d'huiles comestibles tel que : le colza, le tournesol, l'olive, le soja...etc. [2]

Chaque type d'huile a des spécifications propres et requiert par conséquent un procédé de traitement et des paramètres opératoires spécifiques, mais en général, les huiles brutes issues de la pression et de l'extraction des graines oléagineuses, ont une acidité de plus de 0.5% et contiennent : des mucilages, des impuretés, de l'humidité, des pigments colorés et des matières odoriférantes. Les éléments indésirables sont éliminés par les opérations suivantes :

- ❖ **La démulagination ou dégomme** : éliminations des mucilages par action de l'acide phosphorique et centrifugation.

- ❖ **La neutralisation** : action de la fonde caustique sur les acides gras libres pour réduire l'acidité à 0.05 % max et élimination des pâtes par centrifugations en séparateur auto- débordeur.
- ❖ **Lavage** : par eau chaude à 95° et séchage sous vide de 50 tr minimum.
- ❖ **La décoloration** : action de la terre décolorante sur les pigments du carotène et de chlorophylle.
- ❖ **La désodorisation** : distillation des acides gras résiduels et élimination des pigments colorés résiduels sous vide de 2 millibars et une température de 250° c.

L'huile ainsi obtenue est exempté (délivré) d'impuretés, d'humidité et de produits oxydes, à sa sortie de la désodorisation, elle est refroidie à 25° avant d'aller en stockage, puis en conditionnement.

D'autres opérations spécifique à chaque type d'huile sont ajoutées à cette chaine de raffinage, tels que la cristallisation pour l'huile de tournesol et démulcination pour l'huile de soja. Des utilités tels que la vapeur, l'eau brute, l'eau os rosée et l'air comprimée sont mis en œuvre par l'ensemble des procédés de raffinage.

I.3 Les sous-unités de la raffinerie d'huile

La raffinerie d'huile de CEVITAL est composée des sous-unités suivantes: [1]

- ❖ **Sous-unité approvisionnement** : son rôle est de stocker les huiles brutes et l'alimentation de la chaine de raffinage de cette dernière.
- ❖ **Sous-unité de raffinage** : l'emplacement des procédés industriels (les séparateurs, les mélangeurs, les réacteurs....).
- ❖ **Sous-unité de conditionnement** : le lieu de fabrication d'emballage, ainsi que la mise en bouteille du produit fini.
- ❖ **Sous-unité de composition de la patte** : pour le traitement des déchets de raffinage afin de récupérer des sous-produits et d'évaluer les pertes en huiles.
- ❖ **Sous-unité d'épuration des eaux** : pour le traitement des eaux avant de les déverser dans la nature.
- ❖ **Laboratoire de la raffinerie** : pour le contrôle de qualité des huiles.
- ❖ **Laboratoire de conditionnement** : il est destiné au contrôle physicochimique et micro biologique de la matière à son arrivée au port, des produits finis, ainsi que des produits laitiers destinés à la margarinerie.

I.4 Structure de la raffinerie d'huile

La raffinerie d'huile est subdivisée en trois lignes principales [1]

❖ **Ligne de production de 400 T/J chacune :**

Elles sont décomposées en trois sections principales enchainées comme suit

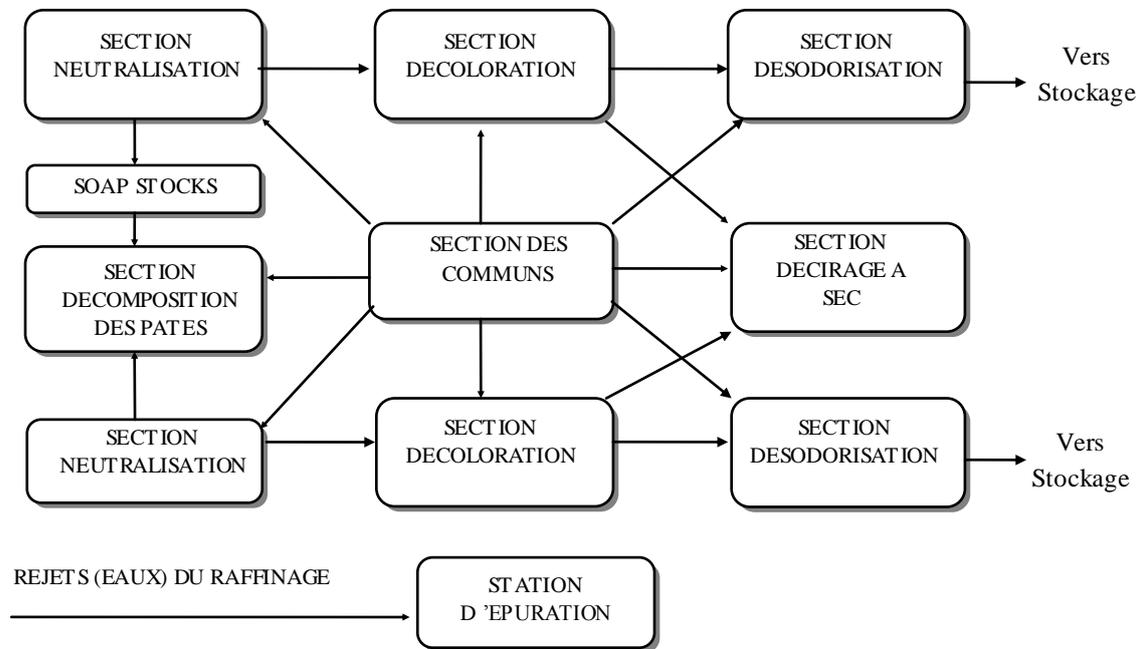


Figure I.1 Organigramme des lignes A et B [1]

❖ **Ligne de production de 1000 T / J**

Nouvellement créer pour satisfaire la demande du marché national, ses trois sections principales sont enchainées comme suit

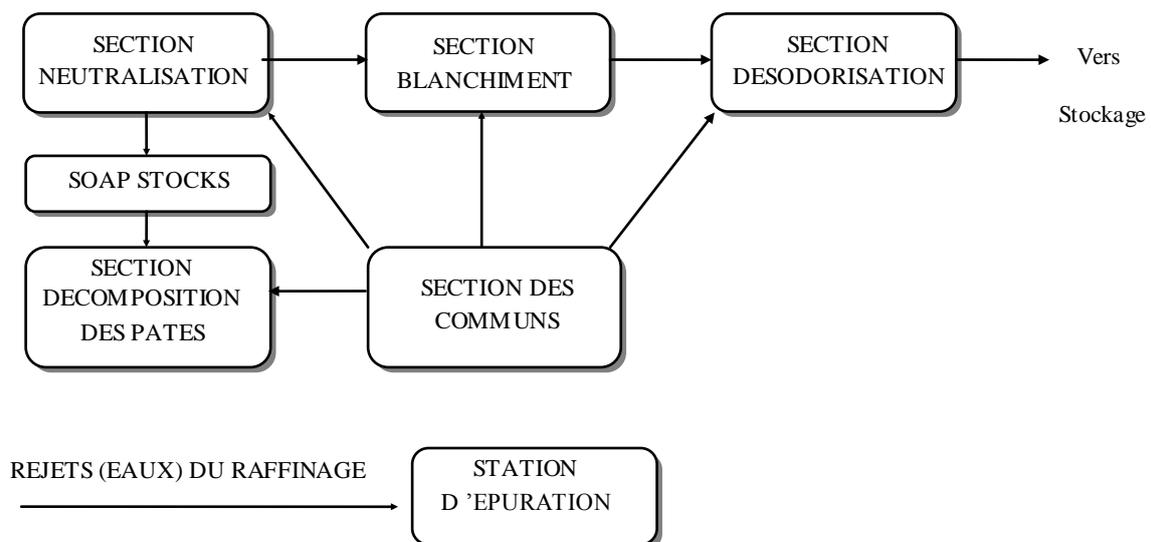


Figure I.2 Organigramme de la ligne C [1]

I.5 Etapes de raffinage d'huile

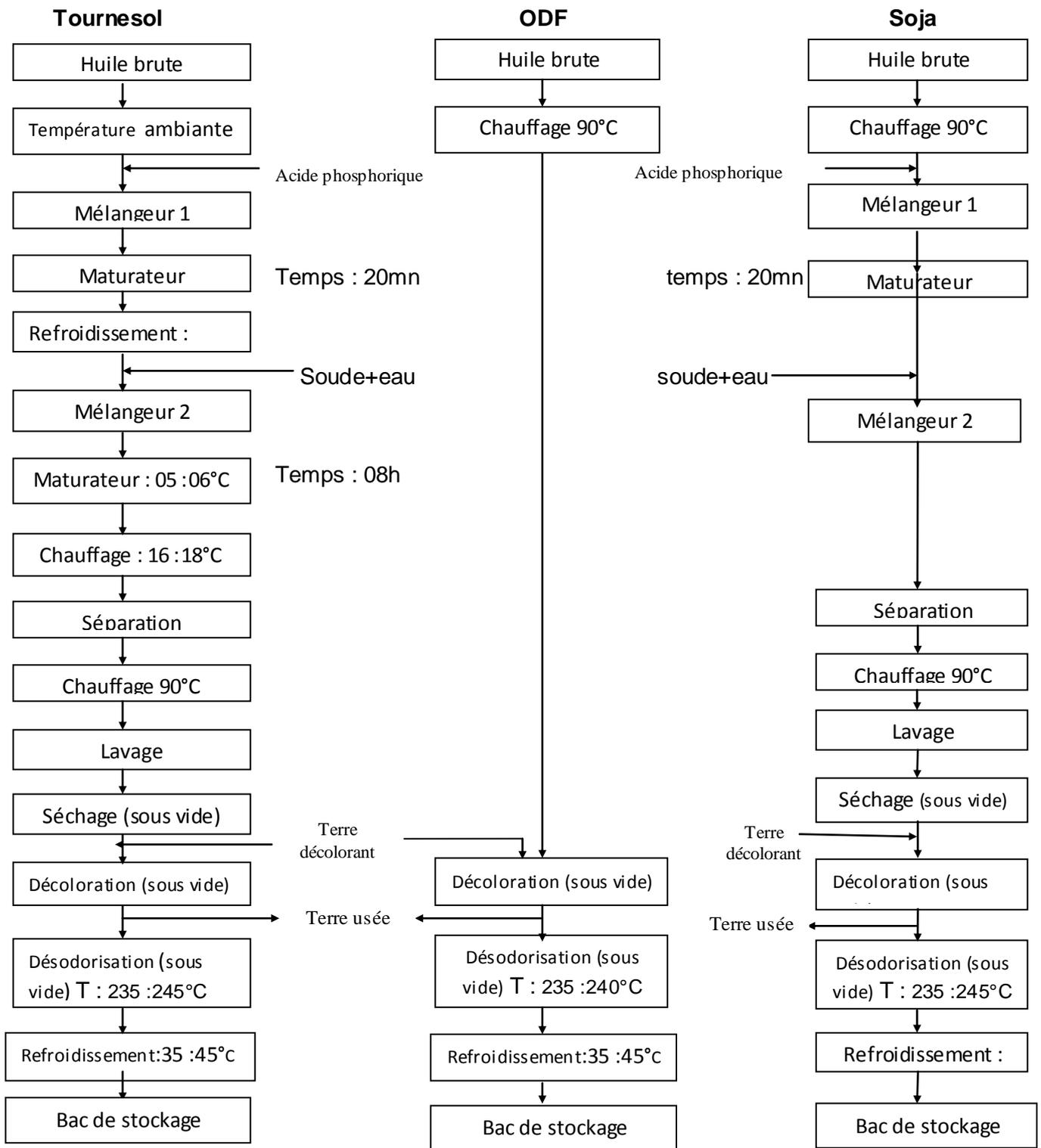


Figure I.3 Diagramme de raffinage [1]

I.5.1 Les procédés industriels de la raffinerie

I.5.1.1 Définition

La raffinerie d'huiles dispose des procédés (équipements) récents qui composent les trois chaînes de raffinage

- ❖ Les deux lignes A et B
- ❖ La ligne C

I.5.1.2 Les différents procédés

❖ Les mélangeurs

Ce sont des appareils qui permettent de mélanger l'huile avec l'acide phosphorique lors de la démulcination, et l'huile avec la soude caustique lors de la neutralisation.

❖ Les réacteurs chimiques

C'est là où s'effectue le brassage pendant 20 à 30 min.

❖ Les cristalliseurs

Dans lesquels passe de l'eau glucosée qui maintient la température à 8°C.

❖ Les séparateurs

Ce sont des équipements qui permettent la séparation des phases.

❖ Les échangeurs

Ils permettent l'échauffement d'huile.

❖ Un sécheur

Permet d'annuler l'humidité d'huile après lavage.

❖ Les filtres

Ils permettent de débarrasser l'huile de la terre décolorante lors de la décoloration.

❖ La colonne de distillation

Elle contient des colonnes à différentes températures qui permettent la désodorisation d'huile.

I.6 La partie électrique

I.6.1 Introduction

Au sein du complexe CEVITAL, l'énergie électrique est distribuée entre ses différentes unités sous forme d'un réseau mixte. La valeur de la puissance utilisée par la partie raffinerie d'huile est de 30KW.

Il existe deux transformateurs qui assurent l'abaissement de la tension jusqu'à 400V et utilise le neutre pour avoir une tension de 200V utilisé pour la partie commande (armoires électriques).

I.6.2 L'appareillage

❖ Alimentation

A partir de la TGBT (Tableau Général Basse Tension), on alimente toutes les armoires électriques des différentes machines.

❖ Armoire électrique

Elle contient tous les équipements électriques nécessaires au fonctionnement et la protection de la raffinerie tels que, les variateurs de vitesse, disjoncteurs, contacteurs, relais... etc.

- Bouton choix de fonctionnement c'est un bouton commutateur noir à deux positions pour le choix de fonctionnement soit manuel ou automatique.
- Démarrage des moteurs ce sont des boutons commutateurs noirs à deux positions marche /arrêt des (vis, transporteurs,... etc.).
- Arrêt d'urgence il existe deux modèles, le premier est un bouton poussoir rouge à Champignon, le deuxième est un fil rouge, les deux sont à accrochage mécanique avec déclenchement par rotation qui arrête tout les fonctionnements de la raffinerie.



Figure I.4 Armoire électrique

❖ Les variateurs de vitesse

Un variateur de vitesse est un équipement permettant de faire varier (commander) la vitesse d'un moteur, une nécessité pour de nombreux procédés industriels.



Figure I.5 Variateur de vitesse

❖ Le disjoncteur

Un disjoncteur est un appareil mécanique de connexion capable d'établir, de supporter et d'interrompre un courant dans un circuit électrique. Un disjoncteur protège l'installation contre : [1]

- Les surcharges (action du déclencheur thermique).
- Contre les courts-circuits (action de déclencheur magnétique).
- Pour remettre l'installation sous tension, il suffit de remettre l'interrupteur en position "on".



Figure I.6 Disjoncteur.

❖ Le sectionneur

C'est un appareil mécanique de connexion ; capable d'ouvrir et fermer un circuit lorsque le courant est nul afin d'isoler la partie de l'installation en aval du sectionneur.

- Il n'a pas de pouvoir de coupure ou de fermeture.
- Ce n'est pas un appareil de protection.

❖ Le contacteur

Un contacteur est un appareil mécanique de connexion, commandé autrement qu'à la main, capable d'établir, de supporter et d'interrompre des courants dans des conditions normales et aussi en cas de surcharges.

L'intérêt du contacteur est de pouvoir être commandé à distance, actionné manuellement ou automatiquement.



Figure I.7 Contacteur

❖ Les fusibles

Le principe de la protection par fusible repose sur la fusion contrôlée d'un élément fusible, fusion qui intervient après un temps donné pour un courant donné.



Figure I.8 Fusible

❖ Manomètre

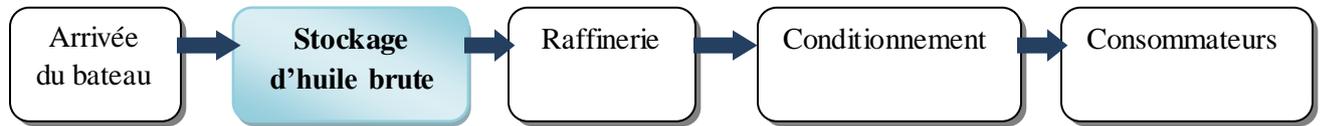
Il sert à Contrôler la pression régnant dans un circuit par lecture sur le cadran d'un appareil de mesure. Ce dernier est monté à demeure sur le circuit (montage en « piquage » sur ce circuit.



Figure I.9 Manomètre

I.7 Présentation du projet

Pour améliorer le taux de production, CEVITAL a mis en place deux nouveaux bacs pour le stockage d'huile brute outre que ceux existant déjà; avant d'envoyer l'huile vers la raffinerie; puis conditionnement et enfin le consommateur Comme suit :



I.10 Structure de production d'huile

Après l'arrivé du bateau, l'huile est stockée dans deux grands bacs en attendant l'étape suivante.

I.8 Présentation de l'état des deux bacs d'huiles brutes

Le stockage d'huile se fait sur deux bacs ; de même caractéristique physique (identique) :

Localisation	Matière	Hauteur Mm	Diamètre mm	Volume T	T° c en service
G	TORNESOL, SOJA, ODF, COLZA	28000 mm	11000 mm	11000 t	35°C
H	TOURNESOL , SOJA, ODF, COLZA	28000 mm	11000 mm	11000 t	35°C

Tableau présentation des deux bacs G et H

❖ Chaque bac est constitué de

- Sonde de température
- Un transmetteur de niveau.
- trois capteurs de niveau
- Cinq vannes
- Trois agitateurs
- deux moteurs pompes.
- Une vanne et un moteur pompe en commun avec les deux bacs.

I.8.1 instrumentation des deux bacs G et H

❖ Sonde de température

La sonde Pt 100 est un capteur de température qui est utilisé dans le domaine industriel (agroalimentaire, chimie, raffinerie...). Ce capteur est constitué d'une résistance en Platine. La valeur initiale du Pt100 est de 100 ohms correspondant à une température de 0°C.[14]



Figure I.11 Sonde de température

➤ Référence

Dans notre projet on utilise une sonde de type : **Omnigrad S TR66 Thermomètre Pt100 ; Entre 0°C à 150°C.**

➤ Caractéristiques

- ✓ Tube prolongateur pour protéger le transmetteur de tête de sonde contre la surchauffe ;
- ✓ Temps de réponse rapide avec extrémité rétreinte/conique ;
- ✓ Gamme de mesure: -200 à 600 °C ;
- ✓ Gamme de pression statique jusqu'à 100 bars ;
- ✓ Le transmetteur transforme la résistivité en intensité (signal normalisé 4-20mA).

❖ Transmetteur de niveau

C'est un transmetteur de niveau radar avec afficheur compact destiné à la mesure continue et sans contact dans les liquides, pâtes et boues. La mesure est insensible aux produits changeants, variations de température, ciels gazeux ou vapeurs. [11]



Figure I.12 Transmetteur de niveau

➤ Référence

On utilise le radar de niveau **Micropilot FMR240**.

➤ Avantages

- ✓ Mesure sans contact : pratiquement indépendante des propriétés du produit.
- ✓ Configuration sur site simple via l'afficheur alphanumérique avec menus déroulants
- ✓ Technique 2 fils, économique : une réelle alternative à la pression différentielle, aux flotteurs et aux dis-placers. La technique 2 fils réduit les couts de câblage et permet une intégration simple dans des systèmes existants
- ✓ En option : traversée étanche aux gaz pour améliorer la sécurité du procès.

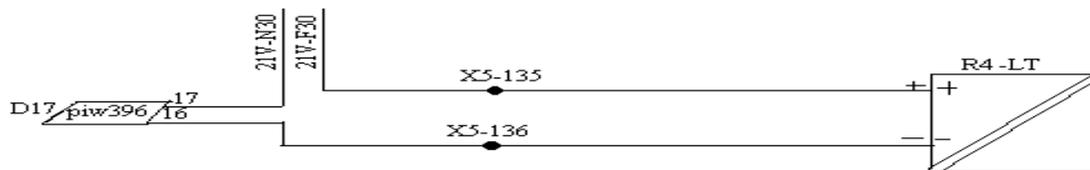


Figure I.13 Schéma électrique du Transmetteur de Niveau [1]

❖ Capteur de niveau

La mesure de niveau est une mesure continue, c'est-à-dire que le capteur et son conditionneur délivrent un signal proportionnel au niveau de liquide dans le réservoir. À chaque instant, l'opérateur connaît exactement le volume du liquide (ou le volume encore disponible dans le réservoir). [11]



Figure I.14 Capteur de niveau

➤ Référence

On utilise un détecteur de niveau de type : **FMR 245**

➤ Caractéristique

- ✓ Courant 4-20Ma
- ✓ Il possède une tige de longueur L=400mm.
- ✓ Hautement résistant jusqu'à +200°C et facile à nettoyer

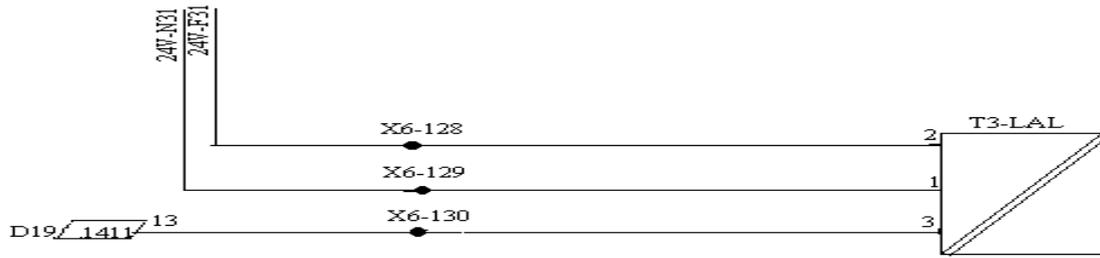


Figure I.15 Schéma électrique capteur de niveau [1]

❖ Vanne

Pour ce type de vanne, la section de passage du fluide est égale à 0% ou 100% de la section de passage à plein ouverture, selon la forme de siège de la vanne et la disposition du clapet.



Figure I.16 Vanne TOR

➤ Caractéristique

- ✓ Vanne TOR

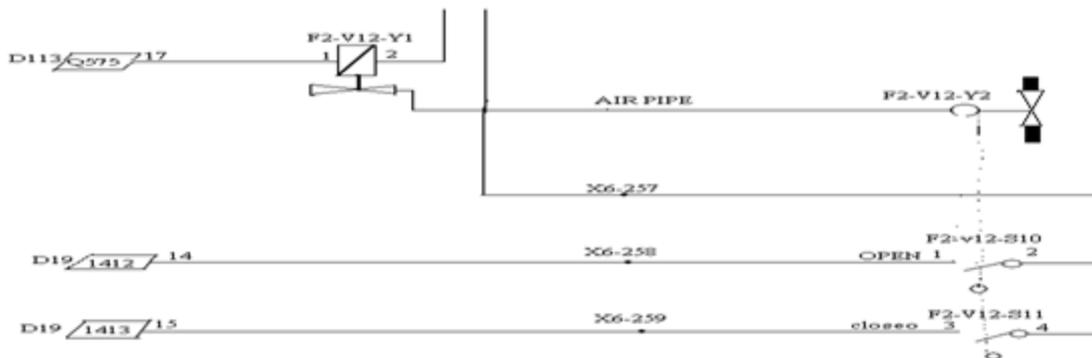


Figure I.17 Schéma électrique de la vanne TOR [1]

❖ AGITATEUR

Les agitateurs ont pour fonction de mélanger l'huile à l'intérieure des bacs, l'opération de soutirage de n'importe quel bac ne doit pas être effectuée si l'agitateur de bac concerné est à l'arrêt ou en panne. Les agitateurs ne doivent pas fonctionner si le niveau d'huile à l'intérieure de bac n'a pas atteint le niveau bas.



Figure I.18 Agitateur

➤ Référence

Moteur réducteur **SEW-USOCOME**
11kw/22A/COS 0.81/1475 (tr/min)

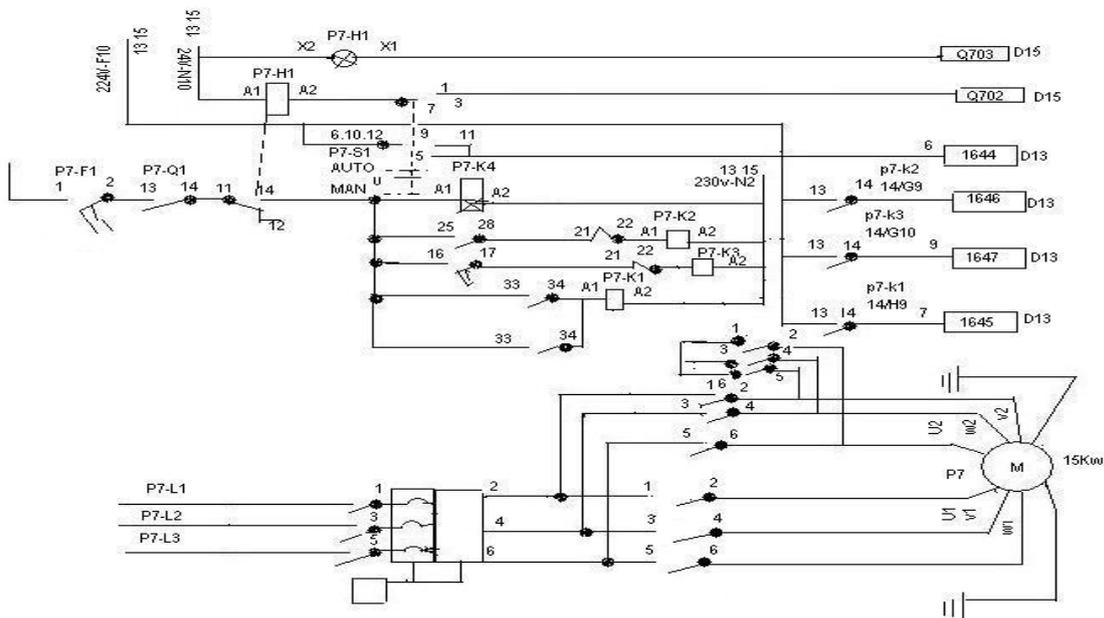


Figure I.19 Schéma électrique de l'agitateur [1]

❖ Moteur pompe

Ce sont les pompes les plus utilisées dans le domaine industriel à cause de la large gamme d'utilisation qu'elles peuvent couvrir, de leurs simplicités et de leurs faibles couts [11]



Figure I.20 Moteur Pompe

➤ Référence

Moteurs **ABB/M2BAT**

➤ **Caractéristique**

- ✓ puissance 132Kw-3000t/m
- ✓ fréquence 50hz/
- ✓ tension 400v
- ✓ courant
- ✓ facteur de puissance COS 0.88

➤ **Principe de fonctionnement**

Une pompe centrifuge est constituée de

- ✓ une roue à aubes tournant autour de son axe
- ✓ un distributeur dans l'axe de la roue
- ✓ un collecteur de section croissante, en forme de spirale appelée volute.

Le liquide arrive dans l'axe de l'appareil par le distributeur, et la force centrifuge le projette vers l'extérieur de la turbine. Il acquiert une grande énergie cinétique qui se transforme en énergie de pression dans le collecteur où la section est croissante.

➤ **Choix du moteur et son démarrage:**

Le moteur à pompe est un moteur asynchrone. Le moteur asynchrone triphasé est largement utilisé dans l'industrie, sa simplicité de construction en fait un matériel très fiable et qui demande peu d'entretien. Il est constitué d'une partie fixe, le stator qui comporte le bobinage, et d'une partie rotative, le rotor qui est bobiné en cage d'écureuil. Les circuits magnétiques du rotor et du stator sont constitués d'un empilage de fines tôles métalliques pour éviter la circulation de courants de Foucault. [13]

➤ **Démarrage étoile /triangle :**

Le démarrage étoile/triangle du moteur asynchrone triphasé est conçu pour limiter le courant de ligne durant la phase de démarrage du moteur asynchrone.

➤ **Conditions à remplir pour le démarrage**

- ✓ Le couplage triangle doit correspondre à la tension du réseau
- ✓ Le démarrage du moteur doit se faire en deux temps :
 - ✓ **Premier temps**: mise sous tension et couplage étoile des enroulements ;
Le moteur démarre à tension réduite.
 - ✓ **Deuxième temps**: Suppression du couplage étoile, et mise en couplage triangle Le moteur est alimenté sous pleine tension.

➤ **Avantage**

- ✓ Réduction du courant de démarrage d'un facteur $\sqrt{3}$.
- ✓ Cout réduit
- ✓ Simplicité de mise en œuvre.

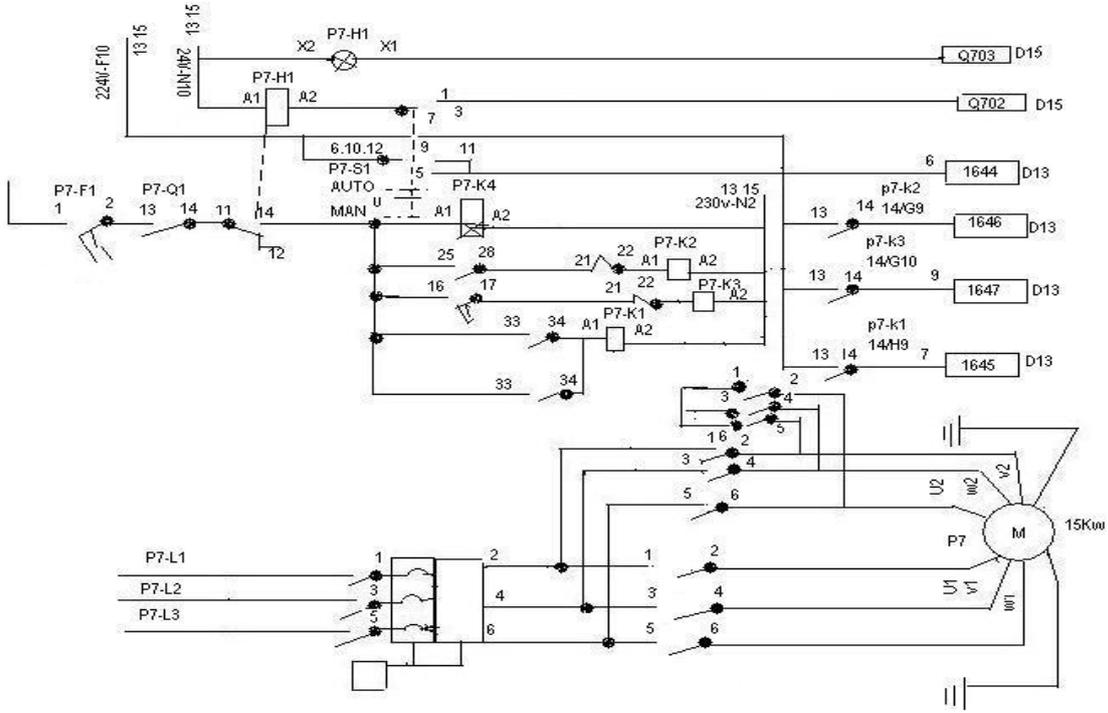


Figure I.21 Schéma électrique d'un Moteur Pompe [1]

I.9 Analyse fonctionnelle des deux bacs G et H

I.9.1 processus de remplissage des bacs

Le remplissage des deux bacs G et H se fait d'après la demande et le besoin du consommateur.

A l'arrivée du bateau avec l'huile commandé par l'opérateur que se soit du : TOURNESOL, SOJA, ODF, COLZA ; on effectue la connexion des tuyauteries du bateau à celle relié aux deux bacs G et H.

Deux huiles de différente nature ne peuvent pas être mélangé; donc après l'analyse de la quantité d'huile commandé et celle déjà consommé on peut connaître l'état des bacs (remplie, à moitié remplie ou vide) et choisir le bac ou les deux en même temps à utiliser d'après les conditions suivantes:

- ❖ Si les bacs G et H sont vide on remplit G jusqu'au remplissage puis H.
- ❖ Si le bac G est déjà rempli on choisit le bac H pour le remplissage.

- ❖ Si le bac G est déjà rempli à moitié et l'huile reçue est de même nature en continue de le remplir, si non on remplit le bac H.
- ❖ Si les deux bacs G et H sont déjà rempli ou à moitié rempli et l'huile reçue n'est pas de même qualité que celui existant ; l'huile est stocker dans d'autre bac.

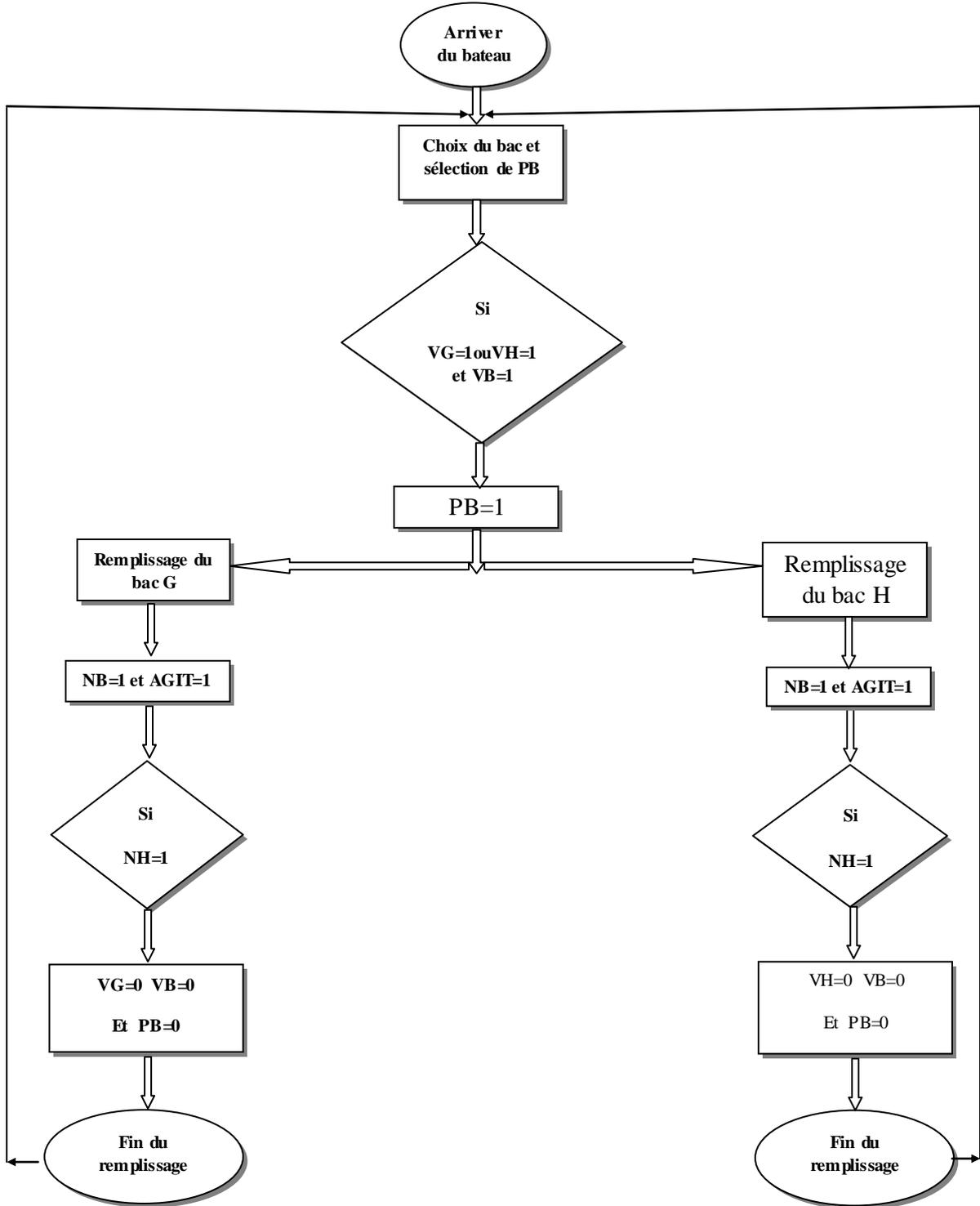


Figure I.22 Diagramme de remplissage

I.9.2 processus d'extraction des deux bacs G et H

On désigne par l'extraction l'utilisation de l'huile brute donc l'envoyée à la raffinerie. Le choix du bac à utiliser se fait par l'opérateur : d'après le type d'huile brute et le niveau du bac.

Après choix du bac l'operateur choisie la pompe à utilisé selon leurs condition :

- ❖ On cas de panne d'une pompe.
- ❖ On cas de panne d'une vanne.
- ❖ Ou même les utilises les deux à la foi.

Pui lance le processus de l'envoie de l'huile brute vers la raffinerie.

Le diagramme suivent explique les étapes pour l'envoi à la raffinerie de l'huile brute.

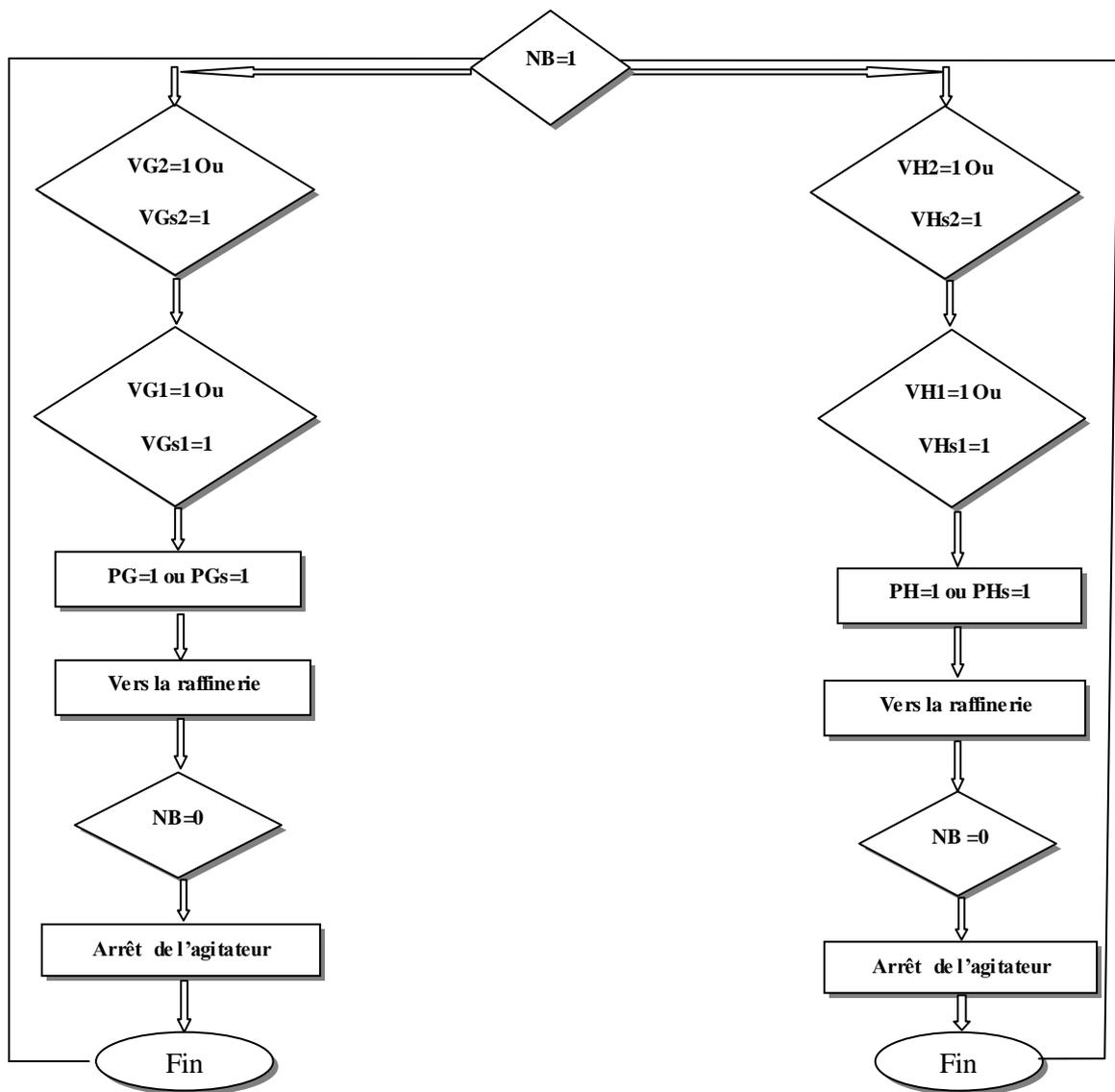


Figure I.23 Diagramme d'extraction d'huile

Conclusion

Dans ce chapitre on a pu étudier les procédés de fabrication d'huile tel que: la neutralisation, décoloration et désodorisation à partir d'huile brute c'est se qu'on appelle le processus de raffinage d'huile; et se familiariser avec les différents instruments utilisés.

On a pu constater que CEVITAL veille à appliquer strictement les normes pour le bien de l'entreprise et ses employés.

La compréhension du processus et les principes de fonctionnement des instruments utilisés dans le projet étudié est une phase importante dans l'étude et la gestion d'un système automatisé, car cela nous permettra de remédier à la problématique et de bien satisfaire les objectifs exigés, ce qui nous facilitera la tâche pour l'élaboration de la commande qui sera traitée dans le chapitre suivant.

Introduction

Aujourd'hui, au XXI^e siècle, l'automatisme est une unité qui nous entoure au quotidien, qu'ils soient simples ou complexes, les systèmes automatisés sont partout dans notre environnement. Avec l'automatisme, on pense souvent au monde de l'industrie. C'est là que l'on trouve principalement les API (Automate Programmable Industriel).

Par ailleurs la supervision industrielle consiste à surveiller l'état de fonctionnement d'un procédé pour l'amener à son point de fonctionnement optimal. Le but c'est de disposer en temps réel d'une visualisation de l'état d'évolution des paramètres du processus, ce qui permet à l'opérateur de prendre rapidement des décisions appropriées à ses objectifs telle que la cadence de production, qualité des produits et sécurité des biens et des personnes.

II.1 Système Automatisé

II.1.1 Objectifs de l'automatisation [3]

- ❖ Elimination des tâches répétitives
- ❖ Simplification du travail humain
- ❖ Augmentation de la sécurité
- ❖ Accroître la productivité
- ❖ Economiser la matière première et l'énergie
- ❖ S'adapter à des contextes particuliers

II.1.2 Structure d'un système automatisé

Tout système automatisé peut se décomposer selon le schéma suivant [4]

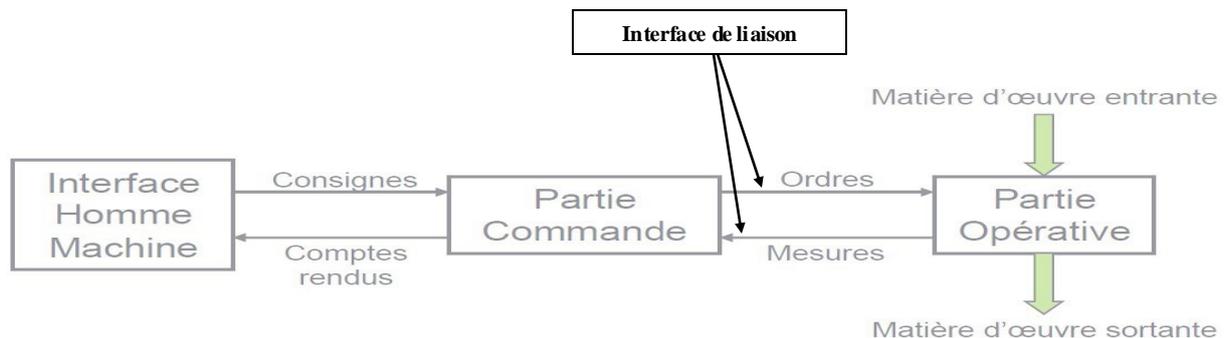


Figure II.1 Structure d'un système automatisé

❖ Partie Commande

C'est la partie qui gère le fonctionnement du Système Automatisé. Elle est en général composée d'un ordinateur qui contient dans ses mémoires un programme. Elle transmet les ordres aux actionneurs de la partie opérative à partir :

- Du programme qu'elle contient,
- Des informations reçues par les capteurs,
- Des consignes données par l'utilisateur ou par l'opérateur.

❖ Partie opérative

Elle consomme de l'énergie électrique, pneumatique (air) ou hydraulique (huile). Elle comporte en général un boîtier (appelé aussi bâti) contenant :

- les actionneurs (transforment l'énergie reçue en énergie utile : moteur, vérin, lampe).
- les capteurs (transforment la variation des grandeurs physiques liée au fonctionnement de l'automatisme en signaux électriques : capteur de position, de température, bouton poussoir).

❖ Poste de contrôle

Il est composé des pupitres de commande et de signalisation et il permet à l'opérateur de commander le système (marche, arrêt, départ de cycle ...).

Il permet également de visualiser les différents états du système à l'aide des voyants, du terminal de dialogue ou d'interface homme machine.

II.2 Généralités sur les automates programmables (API)

II.2.1 Définition

Un automate programmable industriel (API) est un dispositif électronique possédant l'architecture d'un ordinateur (très proche de l'ordinateur) adapté au milieu industriel destiné à la commande de processus par un traitement séquentiel. Il envoie des ordres vers les actionneurs (partie opérative) à partir de données d'entrée (partie commande), des consignes et d'un programme informatique. [2]

II.2.2 Architecture des Automates

II.2.2.1 Structure Externe

Les automates peuvent être de type compact ou modulaire [3]

❖ Type Compact

On distingue les modules de programmation (LOGO de Siemens, ZELIO de Schneider, MILENIUM de Crouzet...) sont des micros automates. Ils intègrent le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, ils pourront réaliser certaines

fonctions supplémentaires (comptage rapide, E/S analogique...etc.) et recevoir des extensions en nombre limité.

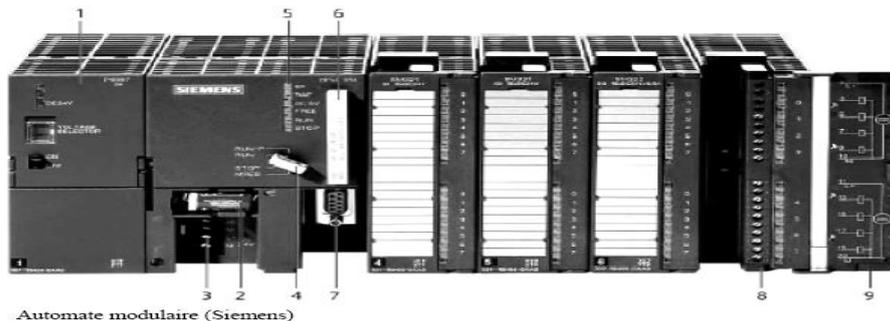
Ces automates, de fonctionnement simple sont généralement destinés à la commande de petits automatismes.



Figure II.2 Automate Compacte

❖ Type Modulaire

On trouve 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 >>. Ces automates sont intégrés dans les automatismes complexes ou puissance, capacité de traitement et flexibilité sont nécessaires.



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

Figure II.3 Automate Modulaire(Siémens)

❖ Le Châssis ou Rack

Les modules d'un automate, à structure modulaire, sont montés sur châssis spécifique (Rack).

Le châssis permet d'assurer :

- L'assemblage mécanique des modules.
- La distribution de la tension d'alimentation aux différents modules.
- L'acheminement des bus (Donnée, adresse commande) vers les modules.

II.2.2.2 Structure Interne

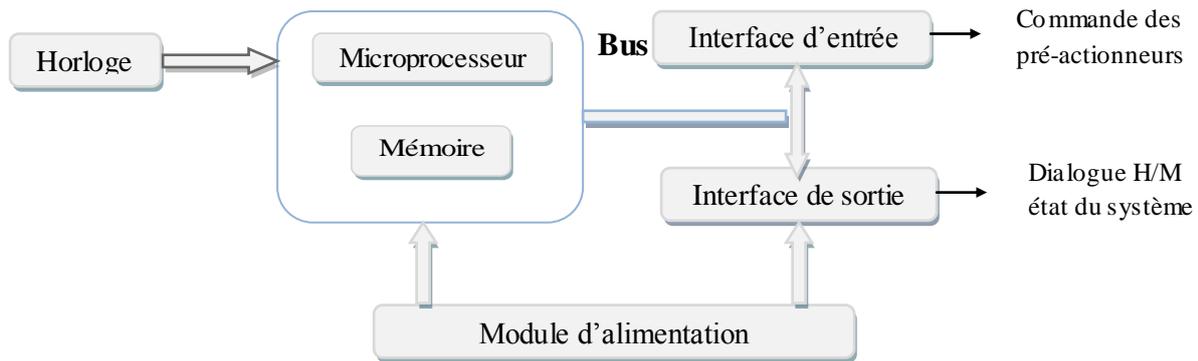


Figure II.4 Structure interne d'un automate

❖ **Unité Centrale(CPU)**

L'unité centrale assure le traitement de l'information et la gestion de l'ensemble des unités. Ce module comporte un microprocesseur, des circuits périphériques de gestion des entrées sortie, des mémoires RAM et EEPROM nécessaires pour stocker les programmes, les données, et les paramètres de configuration du système. Les instructions sont effectuées les unes après les autres, séquencées par une horloge. Elle est caractérisé par :

- **Vitesses de traitement** C'est la vitesse de l'UC pour exécuter 1 K-instructions logiques. (10 à 20 ms/Kmot).
- **Temps de réponse** scrutation des entrées, vitesse de traitement et affectation des sorties.

❖ **Mémoire**

Deux types de mémoire cohabitent :

- **La mémoire Langage** où est stocké le langage de programmation. Elle est en général figée, c'est à dire en lecture seulement. (ROM : mémoire morte).
- **La mémoire Travail** utilisable en lecture-écriture pendant le fonctionnement c'est la RAM (mémoire vive)

❖ **Alimentation (PS)**

Un PS est composé de blocs qui permettent de fournir à l'automate l'énergie nécessaire à son fonctionnement. A partir d'une alimentation en 220V alternative, ces blocs délivrent des sources de tension dont l'automate a besoin : 24V, 12V ou 5V en continu.

❖ Module entrée/sorties TOR (Tout Ou Rien)

La gestion de ce type de variables constituant le point de départ des API reste l'une de leurs activités majeures. Leurs nombres est en générale de 8, 16, 24, ou 32 entrées/sorties, qui peuvent fonctionner :

- En continue 24V, 48V.
- En alternative 24V, 48V, 100V/120V, 200V/240V.

❖ Module entrées/sorties analogique

Elles permettent l'acquisition de mesures (entrées analogique), et générer des signaux de commande (sortie analogique). Ces modules comportent un ou plusieurs convertisseurs analogiques/ numériques (A/N) pour les entrées, et numériques/analogiques (N/A) pour les sorties dont la résolution est de 8 à 16 bits.

Les standards les plus utilisés sont : $\pm 10V$, 0-10V, $\pm 20mA$, 0-20mA et 4-20mA. ces modules sont en général multiplexées en entrée pour utiliser un sel convertisseur A/N, alors que les sorties exigent un convertisseur N/A par voie pour pouvoir garder la commande durant le cycle de l'API.

II.2.3 Cycle de l'automate programmable

Le fonctionnement d'un API est basé sur l'exécution des tâches qui lui sont assignés de manière répétitive, pour cela on représente le cycle de l'automate par cinq phases qui s'exécutent de la manière suivante : [7]

- ❖ **Phase 1 gestion de système** : autocontrôle de l'automate
- ❖ **Phase 2 acquisition des entrées** : prise en compte des informations du module d'entrées et écriture de leurs valeurs dans la RAM (zone de données).
- ❖ **Phase 3 traitement des données** : lecture du programme (située dans la RAM programme) par l'unité de traitement, lecture des variables (RAM données), traitement et écriture des variables dans la RAM de données.
- ❖ **Phase 4 émission des ordres** : lecture des variables de sortie dans la RAM de données et leur transfert vers les modules de sortie.
- ❖ **Phase 5** : dialogue éventuel avec une console ou autre.

II.2.4 Critères de choix d'un automate

Afin de choisir un type d'automate on doit respecter certains critères [3]

- ❖ La capacité de traitement du processeur
- ❖ Le nombre d'entrées/sorties

- ❖ La nature des entrées/sorties (numériques, analogiques, booléennes)
- ❖ Fiabilité du produit
- ❖ Durée de garantie

II.2.5 Nature des informations traitées par l'automate

Les informations peuvent être de type [2]

❖ **Tout Ou Rien (TOR)**

L'information ne peut prendre que deux états (vrai/faux, 0 ou 1). C'est le type d'informations délivrées par un détecteur, un bouton poussoir...etc.

❖ **Analogique**

L'information est continue et peut prendre une valeur comprise dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (pression, température... etc.)

❖ **Numérique**

L'information 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.

II.3 Présentation de l'automate utilisé

L'automate S7-300 est un mini automate modulaire de milieu de gamme de la série S7, fabriqué par la firme SIEMENS. On peut le composer en fonction de nos besoins à partir d'un vaste éventail de module tel que

- ❖ Module alimentation PS.
- ❖ Module unité centrale CPU.
- ❖ Module de signaux d'entrées/sorties SM.
- ❖ Module de fonction FM.
- ❖ Module Processeurs de communication pour liaison réseau CP

Tous les modules du S7-300 sont montrés sous un boîtier .La constitution d'un S7-300 est représentée par la figure suivante [4]

II.3.1 Module de fonction (FM)

Il a pour rôle, l'exécution de tâche du traitement des signaux du processus à temps critique et nécessitant une importante capacité de mémoire comme le comptage, positionnement et régulation.

II.3.2 Module de communication (CP)

Ils permettent d'établir des liaisons Homme-Machine qui sont effectuées par les interfaces de communication suivantes

- Point à point
- Profibus
- Industriel Ethernet

II.4 Caractéristique de la CPU

La gamme S7-300 offre une grande variété de CPU tels que la CPU312, 314M, 315, 315-2DP,...etc. Chacun possède certaines caractéristiques différentes des autres et par conséquent, le choix de la CPU pour un problème d'automatisation donné est conditionné par les caractéristiques offertes par la CPU choisie.

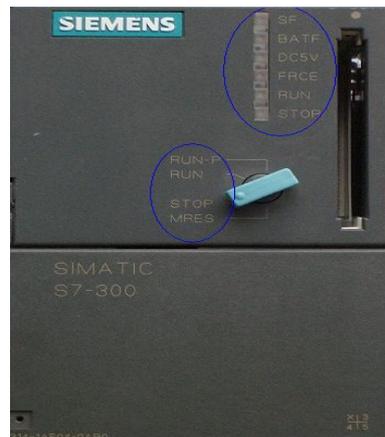


Figure II.5 Représentation de la CPU S7-300

II.4.1 Interface MPI

Une liaison MPI est nécessaire pour programmer un SIMATIC S7-300 depuis le PC ou la PG-MPI signifie **M**ulti **P**oint **I**nterface (interface multipoint). Elle est une interface de communication utilisée pour la programmation, le contrôle-commande avec HMI et l'échange de données entre CPU SIMATIC S7 jusqu'à 32 nœuds maximum. Chaque CPU du SIMATIC S7-300 est équipée d'une interface MPI intégrée.

II.4.2 Signification des états de l'automate

La CPU comporte des LED de signalisation suivante [10]

❖ **SF(rouge)**

signalisation groupée de défauts, elle s'allume si on a défaut matériels et en cas d'erreurs de programmation, de paramétrage, de calcul...etc.

❖ **BATF(rouge)**

Défaut de la pile, elle s'allume si elle est défectueuse, absente ou déchargée.

❖ **DC5V(vert)**

Alimentation 5Vcc pour la CPU et le bus S7-300, elle s'allume si les 5V sont présentes et elle clignote s'il ya surcharge de courant.

❖ **FRCE (jaune)**

elle s'allume en cas de forçage permanent.

❖ **RUN(vert)**

Etat de fonctionnement RUN, elle clignotte en cas de démarrage de la CPU.

❖ **STOP(jaune)**

Etat de fonctionnement STOP, elle s'allume si la CPU ne traite aucun programme utilisateur et elle clignote en cas ou la CPU demande un effacement général.

Les LED de signalisation de défaut SF-DP et BUSF ne se rencontrent que dans le cas de la CPU31 relative à la configuration maitre-esclave du S7-300.

II.4.3 Commutateur de mode de l'automate

❖ **RUN**

La CPU traite le programme de l'utilisateur. Dans cette position, la clé peut être retirée pour éviter q'une personne non habilitée change le mode de fonctionnement.

❖ **RUN-P**

La CPU traite le programme de l'utilisateur et la clé ne peut être retirée, il est possible de lire le programme de la CPU avec une PG (CPU vers PG) et de transférer des programmes dans la CPU (PG vers CPU).

❖ **STOP**

La CPU ne traite aucun programme utilisateur, la clé peut être retirée pour éviter le changement de mode inattendue mais on peut lire et écrire dan la CPU.

❖ MRES

Cette position instable du commutateur de mode de fonctionnement, en vue de l'effacement générale de la CPU. Le contenu de la mémoire de chargement rémanente intégrée reste inchangé après un effacement général.

II.5 Programmation sous STEP7**II.5.1 Aperçu du progiciel STEP7**

Le STEP7 est le progiciel de base pour la programmation et la configuration des systèmes d'automatisation SIMATIC et qui s'exécute sous l'environnement Windows à partir d'une console de programmation ou d'un PC.

Step7 assiste l'utilisateur dans toutes les phases de création des solutions d'automatisation, comme par exemple :

- ❖ Création et la gestion de projet.
- ❖ La configuration et le paramétrage de matériels et de la communication.
- ❖ La gestion des mnémoniques.
- ❖ La création de programme.
- ❖ Le chargement de programme.
- ❖ Le test de l'installation d'automatisation.
- ❖ Le diagnostic lors de perturbation de l'installation.

II.5.2 Création d'un projet sous STEP7

Pour créer un projet avec STEP7, on dispose d'une certaine liberté d'action, en effet on a deux solutions possibles

- ❖ **Alternative 1** commencer par la configuration matérielle.
- ❖ **Alternative 2** commencer par la création du programme.

Le schéma suivant illustre les deux solutions possibles lors de la conception d'une solution d'automatisation

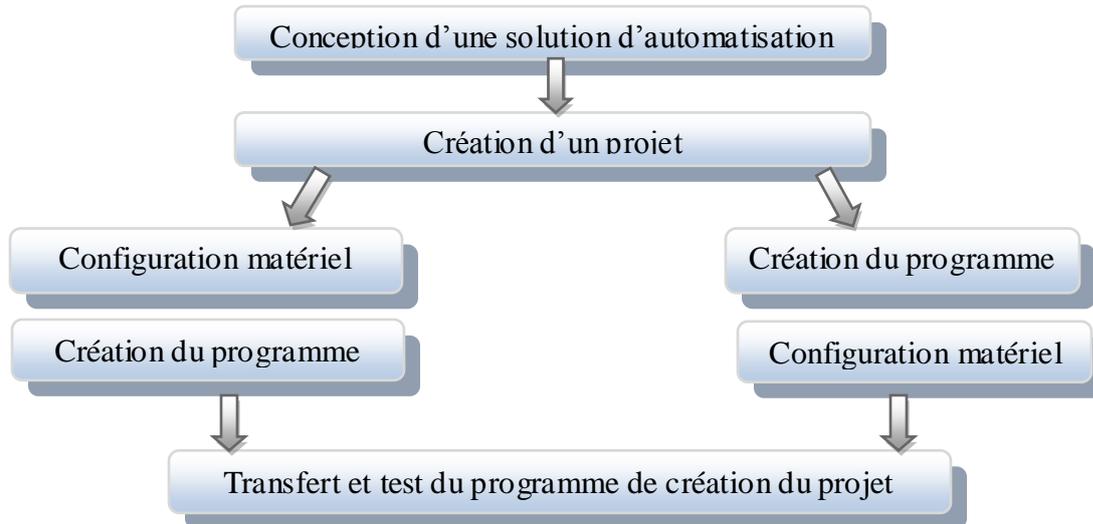


Figure II.6 Organigramme de création du projet [7]

La tâche principale dans la réalisation d'un projet, consiste à préparer ces données et à effectuer la programmation.

Dans un projet, les données sont enregistrées sous forme d'objets. Les objets sont organisés à l'intérieur d'un projet selon une structure arborescente (hiérarchie du projet).

II.5.3 Langages de programmations step7

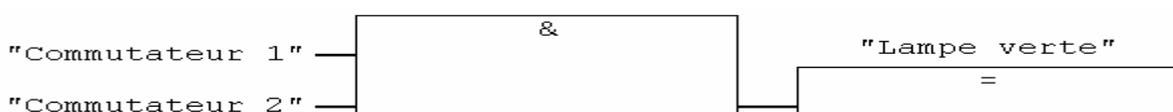
Le logiciel Siemens STEP7 permet une programmation multi langage, c'est-à-dire qu'il peut être programmé dans plusieurs langages différents, qui peuvent être même mélangés dans un même programme (mais pas dans une même sous-routine) : [7]

- ❖ Liste d'instructions ou Instructions List (IL)
- ❖ Langage à contacts ou Ladder diagramme (CONT)
- ❖ Logigramme ou Functional Block (LOG)

II.5.3.1 Programmation à schémas logique (LOG)

Le logigramme (LOG) est un langage de programmation graphique qui utilise les boîtes de l'algèbre de Boole pour représenter les opérations logiques. Les fonctions complexes, comme par exemple les fonctions mathématiques, peuvent être représentées directement combinées avec les boîtes logiques.

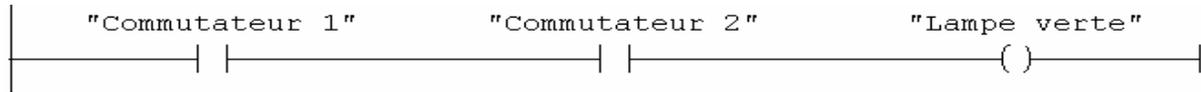
Exemple :



II.5.3.2 Programmation à schéma à contacte (CONT) :

C'est un langage de programmation graphique. La syntaxe des instructions fait penser aux schémas de circuits. CONT permet de suivre facilement le trajet du courant entre les barres d'alimentation en passant par les contacts les éléments complexes et les bobines.

Exemple :



II.5.3.3 Programmation à liste d'instruction (LIST) :

La liste d'instructions (LIST) est un langage de programmation textuel proche de la machine. Dans un programme LIST, les différentes instructions correspondent, dans une large mesure, aux étapes par lesquelles la CPU traite le programme. Pour faciliter la programmation, LIST a été complété par quelques structures de langage évolué (comme, par exemple, des paramètres de blocs et accès structurés aux données).

Exemple :

```

U      "Commutateur 1"
U      "Commutateur 2"
=      "Lampe verte"

```

II.5.4 Structures du programme :

Il existe deux types de programmation [11]

II.5.4.1 Programmation linéaire

Le programme utilisateur peut s'écrire en entier en une seule liste ou dans un bloc (Programmation linéaire) où les instructions s'exécutent les unes après les autres jusqu'à la fin.

Cela n'est toutefois recommandé que pour les programmes d'un simple s'exécutant sur des CPU d'une mémoire peu importante. Le développement d'un tel programme par cette méthode devient difficilement gérable lorsque ce dernier dépasse un certain volume.

II.5.4.2 Programmation Structurée

La programmation structurée consiste à subdiviser un programme plus ou moins complexe en plusieurs sous-programmes (SR) où chacun de ces sous-programmes est développé pour exécuter une tâche ou fonction spécifique. Un autre programme dit programme principal sera chargé de gérer ces sous-programmes (ces sous-programmes

peuvent être des tâches périodiques, tâches événementielles ou des fonctions prédéfinies) et d'en faire appel autant de fois qu'il est nécessaire.

II.5.5 Types de blocs dans le programme utilisateur sous step7

Le logiciel de programmation STEP 7 nous permet de structurer le programme utilisateur, c'est-à-dire de le subdiviser en différentes parties autonomes. Il en résulte les avantages suivants :

- ❖ Ecrire des programmes importants mais clairs.
- ❖ Standardiser certaines parties du programme.
- ❖ Simplifier l'organisation du programme.
- ❖ Modifier facilement le programme.
- ❖ Simplifier le test du programme, car vous pouvez l'exécuter section par section.
- ❖ Faciliter la mise en service.

Les principaux blocs utilisés pour la programmation sous step7 sont : [7]

II.5.5.1 Les blocs d'organisation (OB)

Les modules d'organisation (OB) sont, comme leur nom l'indique, utilisés pour l'organisation interne du programme et forment ainsi un moyen puissant et essentiel pour la programmation structurée. Ils servent par exemple au déroulement cyclique du programme principal, à l'exécution de programmes d'interruption par des fonctions d'alarmes ou de temps, ou par des fonctions diagnostic interne autant du point de vue hardware que software du système complet.

Les blocs d'organisation définissent l'ordre (événements de déclenchement) dans lequel les différentes parties du programme sont traitées. L'exécution d'un OB peut être interrompue par l'appel d'un autre OB. Cette interruption se fait selon la priorité : les OB de priorité plus élevée interrompent les OB de priorité plus faible.

II.5.5.2 Blocs d'organisation OB

On retiendra principalement l'OB1 qui est examiné à chaque cycle d'automate ; il sert à l'exécution du programme utilisateur. C'est donc à partir de ce bloc que l'on fera les appels aux différents blocs de programmes blocs fonctionnels FB ou aux fonctions FC ou à d'autres types de structure. L'OB100 et l'OB101 sont uniquement appelés au démarrage (respectivement à chaud et à froid) qui se fait après la mise sous tension de la CPU qui se

trouve en mode de fonctionnement marche ‘’ RUN ou RUN-P’’. On y appellera donc les blocs traitant les initialisations.

❖ Bloc d'organisation OB22

L'utilité des OB d'alarme temporisée est de programmer l'exécution retardée de certaines parties du programme utilisateur.

II.5.5.3 Les Blocs fonctionnels (FB)

Les blocs fonctionnels sont subordonnés aux blocs d'organisation. Ils renferment une partie du programme qui peut être appelée dans l'OB1 ou dans un autre bloc fonctionnel FB.

Avant de commencer la programmation du bloc fonctionnel, il est indispensable de remplir la table de déclaration des variables d'entrées/sorties dans chaque bloc fonctionnel, en utilisant des noms qui ne figurent pas dans la table des mnémoniques, ainsi que les paramètres formels et les données statiques.

II.5.5.4 Les blocs FC

Une fonction FC est un bloc de code sans mémoire. Les variables temporaires d'une fonction sont sauvegardées dans la pile des données locales qui sont perdues à l'achèvement de la fonction.

Il Permet de simplifier la tâche du programmeur, le Bloc est écrit une fois pour exécuter une fonction et peut être utilisé autant de fois que nécessaire. Pour cela nous allons définir dans la partie déclarative du bloc les :

- **IN** : paramètre entrant du bloc.
- **OUT** : paramètre sortant du bloc.
- **IN/OUT** : paramètre sortant ou entrant du bloc.
- **TEMP** : Mémoire Interne remise à zéro à chaque cycle.

II.5.5.5 Les blocs de données DB

Dans les blocs de données, sont mémorisées les données nécessaires au traitement du programme et les données affectées à chaque bloc fonctionnel. On distingue deux types de blocs de données :

II.5.5.6 Blocs de données d'instance

Un bloc de données d'instance est associé à chaque appel de bloc fonctionnel FB transmettant des paramètres. Ces blocs contiennent les paramètres effectifs et les données statiques du FB.

Les variables déclarées dans la FB déterminent la structure du bloc de données d'instance. L'instance est l'appel d'un bloc fonctionnel, Si par exemple, un bloc fonctionnel est appelé cinq fois dans le programme utilisateur S7, il existe cinq instances de ce bloc.

II.5.5.7 Blocs de données globaux :

Contrairement aux blocs de code, les blocs de données ne contiennent pas d'instruction STEP7 ; ils servent à l'enregistrement des données du programme utilisateur pouvant être utilisées par tous les autres blocs : ils contiennent des données variables que le programme utilisateur utilise.

II.5.6 les mémentos

Les mémentos sont des éléments électroniques bistables servant à mémoriser l'état logique 0 et 1, ils sont utilisés pour les opérations internes de l'automate pour lesquels l'émission d'un signal n'est pas nécessaire.

Chaque automate programmable dispose d'une grande quantité de mémentos, on programme ces derniers comme des sorties en cas de panne de la tension de service, le contenu sauvegardé des mémentos sera alors perdu.

II.6 Simulation du programme sous step7 avec S7-PLCSIM

II.6.1 Présentation de S7-PLCSIM

S7-PLCSIM est une application qui permet d'exécuter et de tester le programme utilisateur élaboré dans un automate programmable (AP), la simulation s'exécute dans un ordinateur ou dans une console de programmation elle est complètement réalisée au sein du logiciel STEP 7, il n'est pas nécessaire qu'une liaison soit établie avec un matériel S7 quelconque (CPU ou module de signaux).

S7-PLCSIM fournit une interface simple avec le programme utilisateur STEP 7 pour visualiser et forcer différents objets comme des entrées et des sorties. Tout en exécutant le programme dans l'AP de simulation, il procure également la possibilité de mettre en œuvre les diverses applications du logiciel STEP 7 comme, par exemple, la table des variables (VAT) afin d'y visualiser et d'y forcer des variables.

II.6.2 Caractéristiques de S7-PLCSIM

Le S7-PLCSIM nous permet d'exécuter les tâches suivantes: [7]

- ❖ L'activation ou la désactivation de la simulation dans le gestionnaire de projets SIMATIC ce fait en cliquant sur le bouton de simulation . Au démarrage de S7-PLCSIM, on peut ouvrir une simulation existante ou sélectionner un nœud dans un projet STEP 7 pour l'accès à la CPU. Lorsque S7-PLCSIM s'exécute, toute nouvelle liaison depuis STEP 7 est automatiquement dirigée vers la CPU de simulation.

- ❖ S7-PLCSIM permet d'exécuter des programmes destinés aux CPU S7-300 ou S7-400, y compris la CPU 317-T sur l'AP de simulation.

- ❖ La création des fenêtres dans lesquelles il y 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 l'adressage symbolique.

- ❖ Il fournit la possibilité d'exécuter des temporisations automatiquement ou les définir et les réinitialiser manuellement. La réinitialisation peut porter sur des temporisations individuelles ou sur toutes les temporisations à la fois.

- ❖ Il offre la possibilité de changer la position du commutateur de mode de fonctionnement de la CPU (RUN, RUN-P et STOP). De plus, la simulation de modules met à la disposition de l'utilisateur une fonction de pause qui permet d'interrompre momentanément la CPU, sans affecter l'état du Programme.

- ❖ Il permet aussi l'enregistrement d'une série d'évènements (manipulation des zones de mémoire d'entrée et de sortie, mémentos, temporisations et compteurs) et lire votre enregistrement afin d'automatiser les tests de programme

II.7 Généralités sur la supervision

II.7.1 Définition [6]

La supervision est une forme évoluée de dialogue Homme-Machine, dont la possibilité va bien au-delà de celle des fonctions de conduite et surveillance réalisée avec les interfaces de dialogues décrite précédemment.

Les fonctions de la supervision sont nombreuses, on peut citer quelques unes :

- ❖ Elle répond à des besoins nécessitant en général une puissance de traitement importante.
- ❖ Assure la communication entre les équipements d'automatismes et les outils consistant à surveiller l'état de fonctionnement d'un procédé dont les possibilités vont bien au

delà de celle de fonctions de conduite et surveillance réalisée avec les interfaces informatiques d'ordonnancement et de gestion de production.

- ❖ Coordonne le fonctionnement d'un ensemble de machines enchainées constituant une ligne de production, en assurant l'exécution d'ordres communs (marche, arrêt,...) et de tâches telles que la synchronisation.
- ❖ Assiste l'opérateur dans les opérations de diagnostic et de maintenance.

II.7.2 Avantages de la supervision

Un système de supervision donne de l'aide à l'opérateur dans la conduite du processus, son but est de présenter à l'opérateur des résultats expliqués et interprétés et son avantage principal est : [9]

- ❖ La surveillance du processus à distance.
- ❖ La détection des défauts.
- ❖ Le diagnostic et le traitement des alarmes.
- ❖ Le traitement des données.

II.8 Présentation du logiciel de supervision WINCC flexible

II.8.1 Introduction

Lorsque la complexité du 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 dernière s'obtient au moyen de l'interface HMI qui signifie 'Humain Machine Interface'.

WinCC est un système HMI performant utilisé sous Microsoft Windows XP, il constitue l'interface entre l'homme (opérateur) et la machine (installation/processus).

Le contrôle proprement dit du processus est assuré par les automates programmables. Il s'établit par conséquent une communication entre Win CC et l'opérateur d'une part et entre Win CC et les automates programmables d'autre part.

Un système HMI se charge des tâches suivantes

❖ Représentation du processus

Le processus est représenté sur le pupitre opérateur. Lorsqu'un état du processus évolue par exemple, L'affichage du pupitre opérateur est mis à jour.

❖ Commande du processus

L'opérateur peut commander le processus via l'interface utilisateur graphique, il peut par exemple définir une valeur de consigne pour un automate ou démarrer un moteur.

❖ Vue des alarmes

Lorsque surviennent des états critiques dans le processus, une alarme est immédiatement déclenchée (par exemple lorsqu'une valeur limite est franchie).

❖ Archivage de valeurs processus et d'alarmes

Les alarmes et valeurs processus peuvent être archivées par le système HMI, on peut ainsi documenter la marche du processus et accéder ultérieurement aux données de la production écoulée.

❖ Documentation des valeurs processus et d'alarmes

Les alarmes et valeurs processus peuvent être éditées par le système HMI sous forme de journal.

❖ Gestion des paramètres de processus et de machine

Les paramètres du processus et des machines peuvent être enregistrés au sein du système HMI dans des recettes. Ces paramètres sont transférables en une seule opération sur l'automate pour démarrer la production d'une variante du produit par exemple.

II.8.2 Description du logiciel WinCC flexible

WinCC flexible, est un logiciel compatible avec l'environnement step 7, et propose pour la configuration de divers pupitre opérateur, une famille de systèmes d'ingénierie, évolutifs adaptés aux tâches de configuration.

Crée l'interface graphique et les variables, c'est pouvoir lire les valeurs des processeurs via l'automate, les afficher pour que l'opérateur puisse les interpréter et ajuster, éventuellement, le processus toujours via l'automate. [8]

II.8.3 Eléments de WinCC flexible

L'environnement de travail de WinCC Flexible se compose de plusieurs éléments. Certains de ces éléments sont liés à des éditeurs particuliers et sont uniquement visibles lorsque cet éditeur est activé. Il met à disposition un éditeur spécifique pour chaque tâche de configuration.

On peut configurer par exemple l'interface utilisateurs graphique d'un pupitre opérateur avec l'éditeur 'vue'. Pour la configuration des alarmes, on utilise par exemple l'éditeur 'Alarmes TOR'.

Les différents outils et barres de l'éditeur de vues sont représentés comme cela Contient toutes les commandes nécessaires à l'utilisation du WinCC Flexible. Les raccourcis disponibles sont indiqués aux regards de la commande menu.

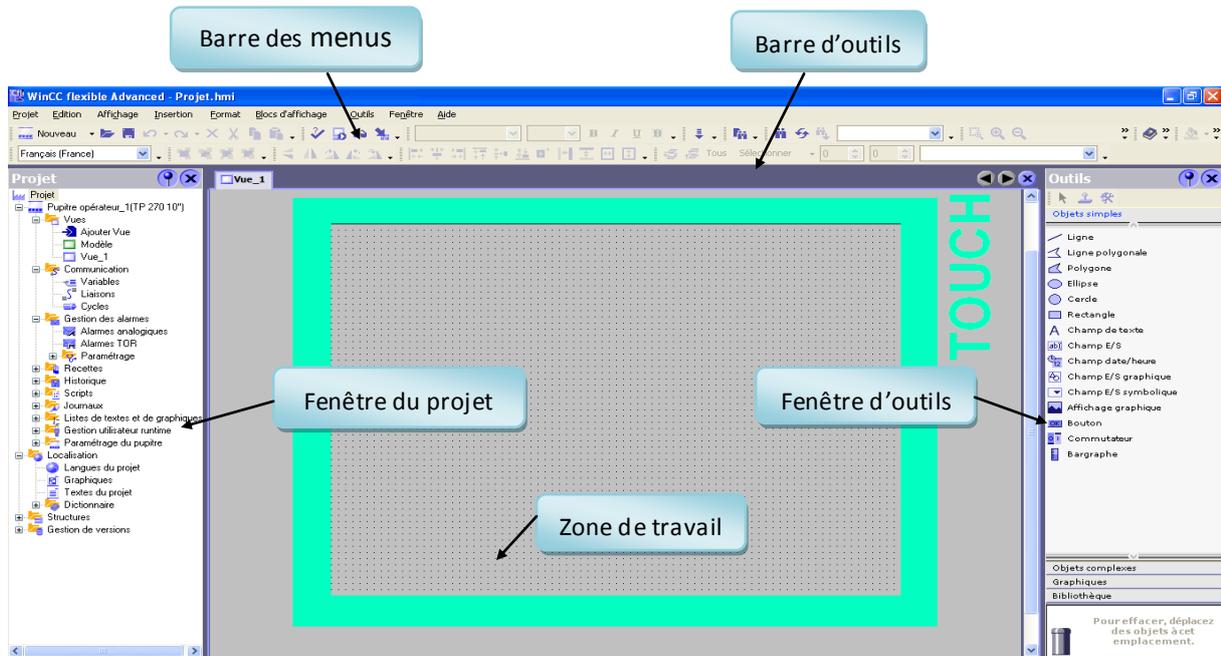


Figure II.7 Interface logicielle WinCC

❖ Barre d'outils

La barre d'outils permet d'afficher tout ce dont le programmeur a besoin.

❖ Zone de travail

La zone de travail sert à configurer des vues, de façon qu'il soit les plus compréhensibles par l'utilisateur, et très faciles à manipuler et consulter les résultats.

❖ Fenêtre d'outils

La fenêtre des outils propose un choix d'objets simples ou complexes qu'on insère dans les vues, par exemple les objets graphiques et les éléments de commande.

❖ Fenêtre du projet

La fenêtre sert à gérer les paramètres du projet.

II.8.4 WinCC flexible Runtime

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

- ❖ communication avec les automates
- ❖ Affichage des vues à l'écran
- ❖ Commande du processus comme par exemple ouverture et fermeture des vannes
- ❖ Archivage des données de runtime actuelles, des valeurs processus et événements de signalisation (alarmes) [13]

II.8.5 Système graphique

II.8.5.1 L'éditeur graphique

Les éditeurs graphiques tels que l'éditeur de vues affichent les objets correspondants aussi bien dans la fenêtre du projet que dans la fenêtre des objets. Dans le cadre d'éditeurs graphiques, on ouvre chaque objet dans la zone de travail.

II.8.5.2 Bibliothèques

Les bibliothèques de composants facilitent considérablement la création des vues. Il suffit d'intégrer lors de la configuration les objets de la bibliothèque dans les vues par glisser-déplacé.

Il existe quatre bibliothèques : (objets simples, objets complexes, graphiques et bibliothèques), elles contiennent de nombreux objets prédéfinis, tel que vannes, moteurs, tuyauteries, instruments d'affichage et autres.

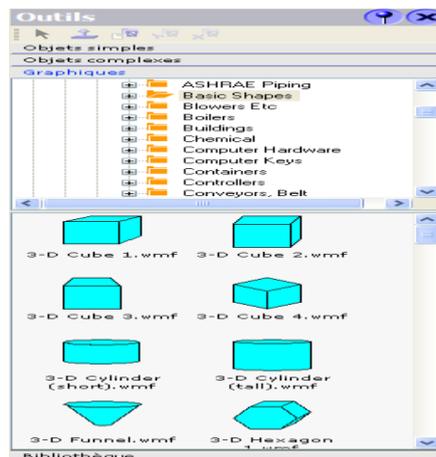


Figure II.8 Fenêtre des outils

II.8.6 Les liaisons

L'échange de données entre deux partenaires de communication est considéré comme une communication. Les partenaires de communication (entre les CPU et les modules de

communication, entre les pupitres opérateur et les processeurs de communication dans le PC) peuvent être reliés via une liaison directe ou via un réseau.

Les données échangées entre les partenaires de communication peuvent remplir des objectifs différents :[8]

- ❖ Commande d'un processus
- ❖ Acquisition de données provenant du processus
- ❖ Signalisation d'état dans un processus
- ❖ Archivage de données de processus

La communication entre le pupitre opérateur et l'automate s'effectue dans Win CC flexible via des variables et une zone de communication.

Dans l'éditeur '**Liaisons**' on peut paramétrer dans l'onglet 'paramètres' les propriétés d'une liaison entre le pupitre opérateur et le partenaire de communication.

Les paramètres de communication sont schématiquement représentés dans l'onglet 'paramètres'. Selon l'interface utilisée, différents paramètres peuvent être sélectionnés pour le 'pupitre opérateur' le 'réseau' et 'l'automate'.

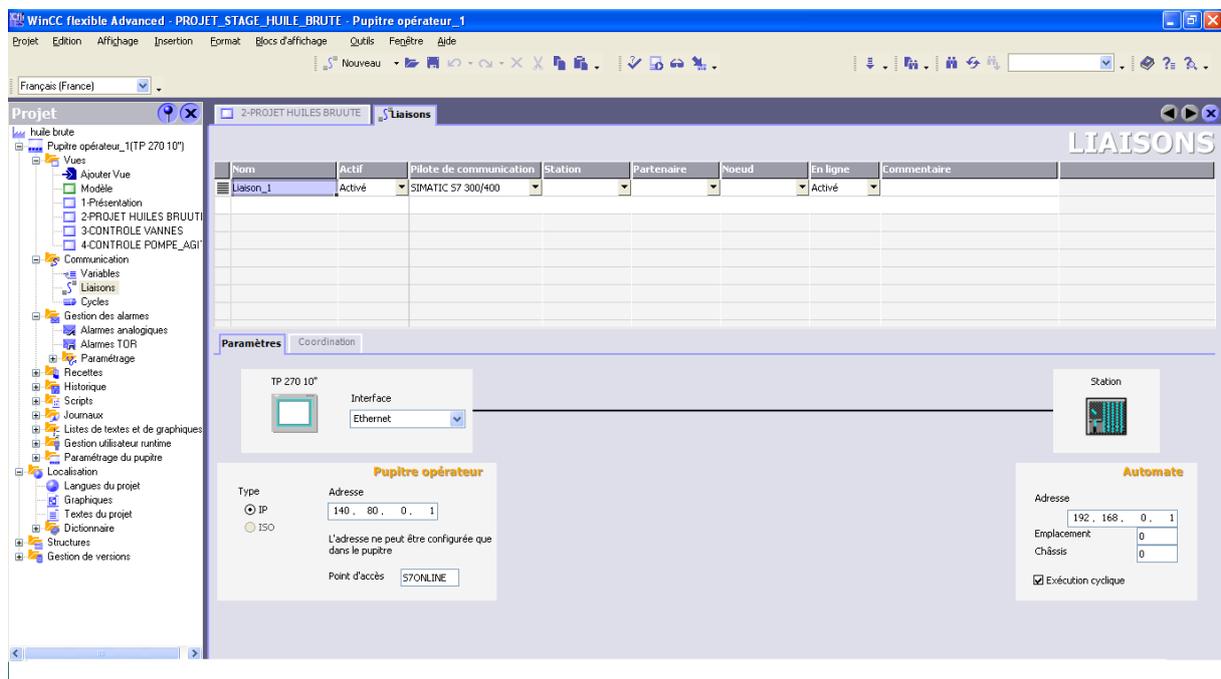


Figure II.9 L'éditeur de liaison

II.8.7 Création des vues

Dans Win CC flexible, on peut créer des vues pour le contrôle-commande de machines et d'installations. Pour créer des vues, on dispose d'objets prédéfinis permettant de représenter notre installation, d'afficher des procédures et de définir des valeurs de processus. Une vue peut être composée d'éléments statiques et d'éléments dynamiques (qui changent au Runtime). [13]

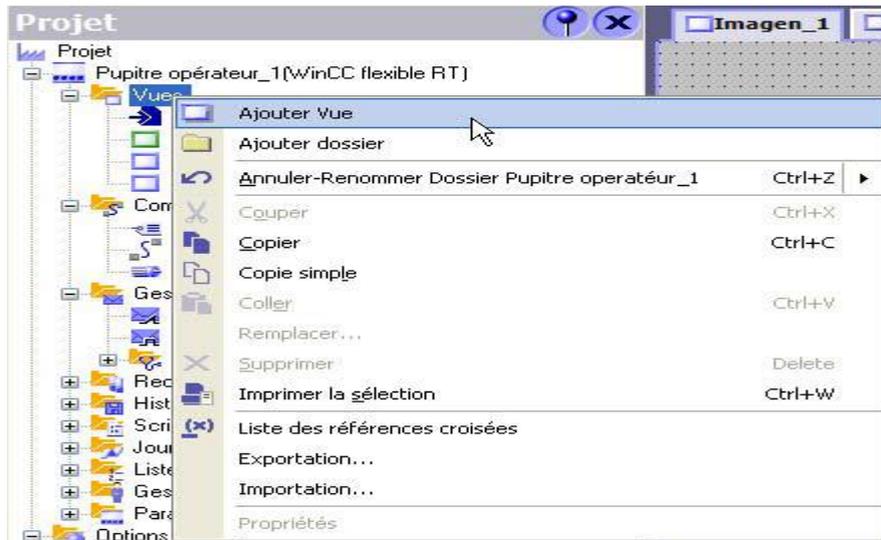


Figure II.10 Création d'une vue [13]

II.8.8 Les variables

Les variables externes permettent de communiquer, C'est-à-dire d'échanger des données entre les composants d'un processus automatisé, entre un pupitre opérateur et un automate par exemple. Une variable externe est l'image d'une cellule mémoire définie de l'automate. L'accès en lecture et en écriture à cette cellule mémoire est possible aussi bien à partir du pupitre opérateur que de l'automate.

Les variables externes étant une image d'une cellule mémoire de l'automate, les types de données utilisables dépendent de l'automate auquel le pupitre opérateur est connecté. [13]

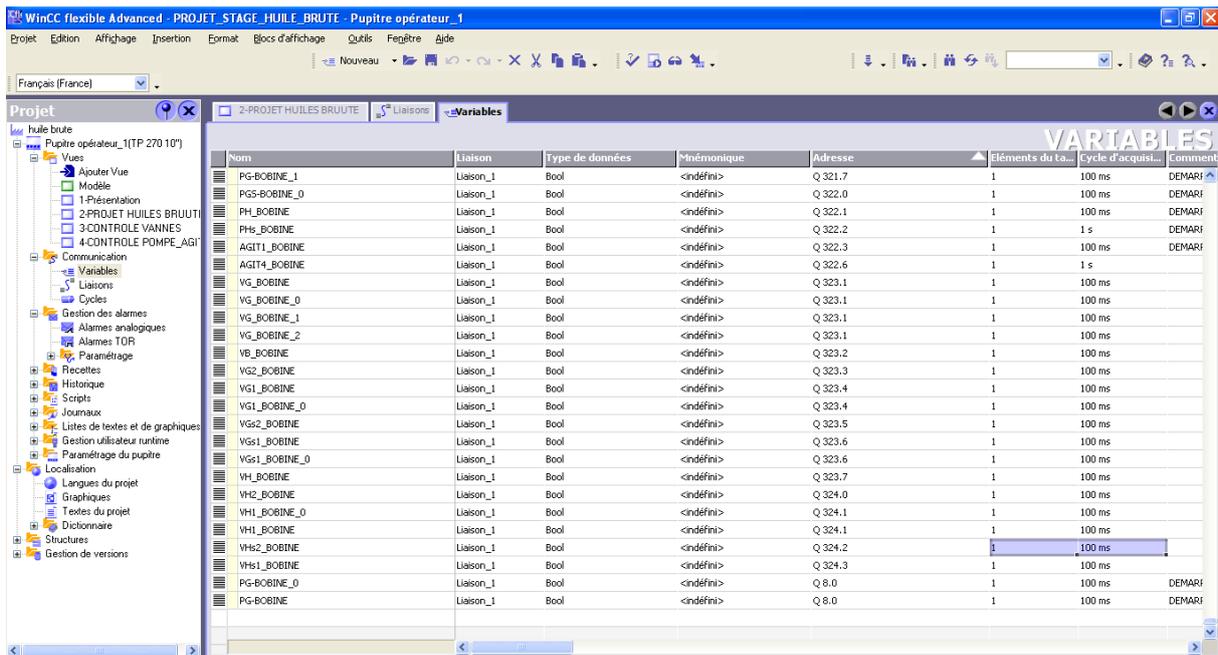


Figure II.11 L'éditeur variable

II.8.9 Alarmes systèmes

Des alarmes système sont prédéfinies dans le pupitre opérateur et dans l'automate afin d'afficher certains états système de ces appareils.

Les alarmes utilisateur et alarmes système sont déclenchées par le pupitre opérateur ou par l'automate et peuvent s'afficher sur le pupitre opérateur.

❖ Taches du système d'alarmes

- **Affichage sur le pupitre opérateur** pour signaler les événements ou les états survenant dans l'installation ou dans le processus
- **Listage** les événements d'alarme sont imprimés
- **Archivage** les événements d'alarmes sont enregistrés pour être traités et évalués.

❖ Etats des alarmes

Les alarmes de bit et les alarmes analogiques peuvent prendre les états suivants :

- Lorsque la condition de déclenchement d'une alarme est vraie, l'alarme est à l'état 'Apparaissant'. Lorsque l'opérateur a acquitté l'alarme, elle est à l'état 'Apparaissant/Acquittée'.

- Lorsque la condition de déclenchement d'une alarme n'est plus vraie, l'alarme est à l'état 'Apparaissant/Disparaissant'. Lorsque L'opérateur a acquitté l'alarme disparue, elle est à l'état 'Apparaissant/Disparaissant/Acquittée'.

Chaque apparition de ces états peut être affichée et archivée sur le pupitre opérateur ainsi que listée sur une imprimante.

Heure	Catégorie	Description
01:58:00.70	TIA	Mise à jour de \Step7\Scie RHP 260...
01:58:00.70	TIA	Mise à jour de \Step7\Scie RHP 260...
01:58:00.70	TIA	Mise à jour de \Step7\Scie RHP 260...
01:58:00.70	TIA	Mise à jour de \Step7\Scie RHP 260...
01:58:00.71	TIA	Synchronisation STEP 7 terminée

Figure II.12 Fenêtre des erreurs et avertissements

Conclusion

L'étude des différentes composantes des automates de la gamme S7-300 et leurs caractéristiques dans ce chapitre nous a permis d'évaluer leurs importance et de constater leur rôle primordial dans la gestion des processus industriels ainsi que le rôle du logiciel **S7-PLCSIM** pour le test du programme élaboré.

Nous avons présenté le logiciel de supervision Win CC flexible et quelque notion de base tel que : la partie configuration, les différentes options qu'il possède ainsi que son fonctionnement.

Introduction

Avec l'évolution de la technologie ; les exigences attendues de l'automatisation sont très importantes. Elle doit assurer l'augmentation de la productivité, l'amélioration de la qualité et la diminution des coûts de production. En plus de cela elle doit garantir également l'amélioration des conditions de travail, la sécurité et la sûreté de fonctionnement et la suppression des tâches pénibles ou répétitives.

Pour répondre à toutes ces exigences dans l'installation présentée précédemment, la mise en place d'un automate programmable et une supervision industrielle est nécessaire afin d'assurer une bonne gestion du stockage d'huile.

III.1 Résultats pratique

III.1.1 Structure du Grafcet

Le Grafcet (Graphe Fonctionnel de Commande Etape Transition), également appelé Diagramme Fonctionnel en Séquence ou Séquentiel Function chart, permet l'établissement de description de la fonction et du comportement des systèmes de commandes en établissant une représentation graphique indépendante de la réalisation technologique.

Plus pragmatiquement, le Grafcet est destiné à représenter des automatismes logiques séquentiels, c'est-à-dire des événementiels dans lesquelles les informations sont de type booléennes (tout ou rien) ou peuvent s'y ramener (numériques). Le GRAFCET est utilisé généralement pour spécifier et concevoir le comportement souhaité de la partie commande d'un système mais il peut également être utilisé pour spécifier le comportement attendu de la partie opérative ou bien de tout le système de commande.

Destiné à être un moyen de communication entre l'automaticien et son client, le Grafcet est un outil utilisé pour la rédaction d'un cahier de charge d'un automatisme. Cependant un des points forts du Grafcet est la facilité de passer du modèle au Grafcet à l'implémentation technologique de celui-ci dans un automate programmable industriel. [5]

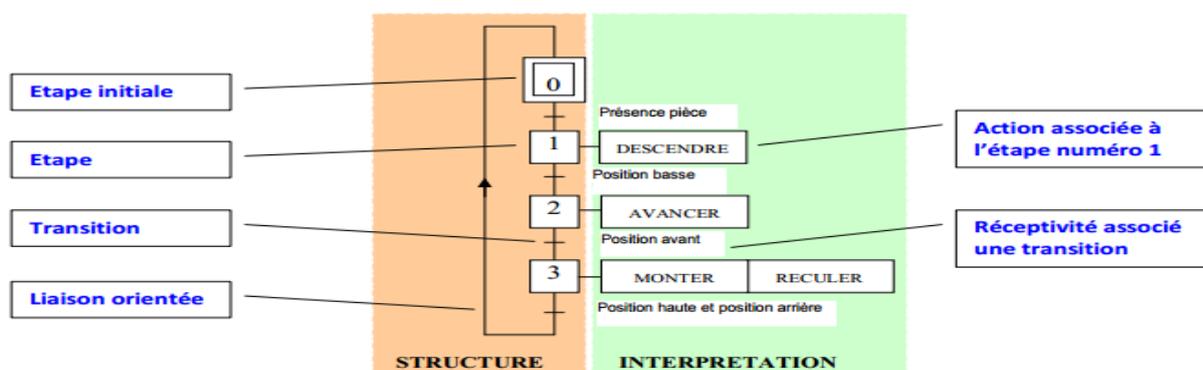


Figure III.1 Structure du GRAFCET [12]

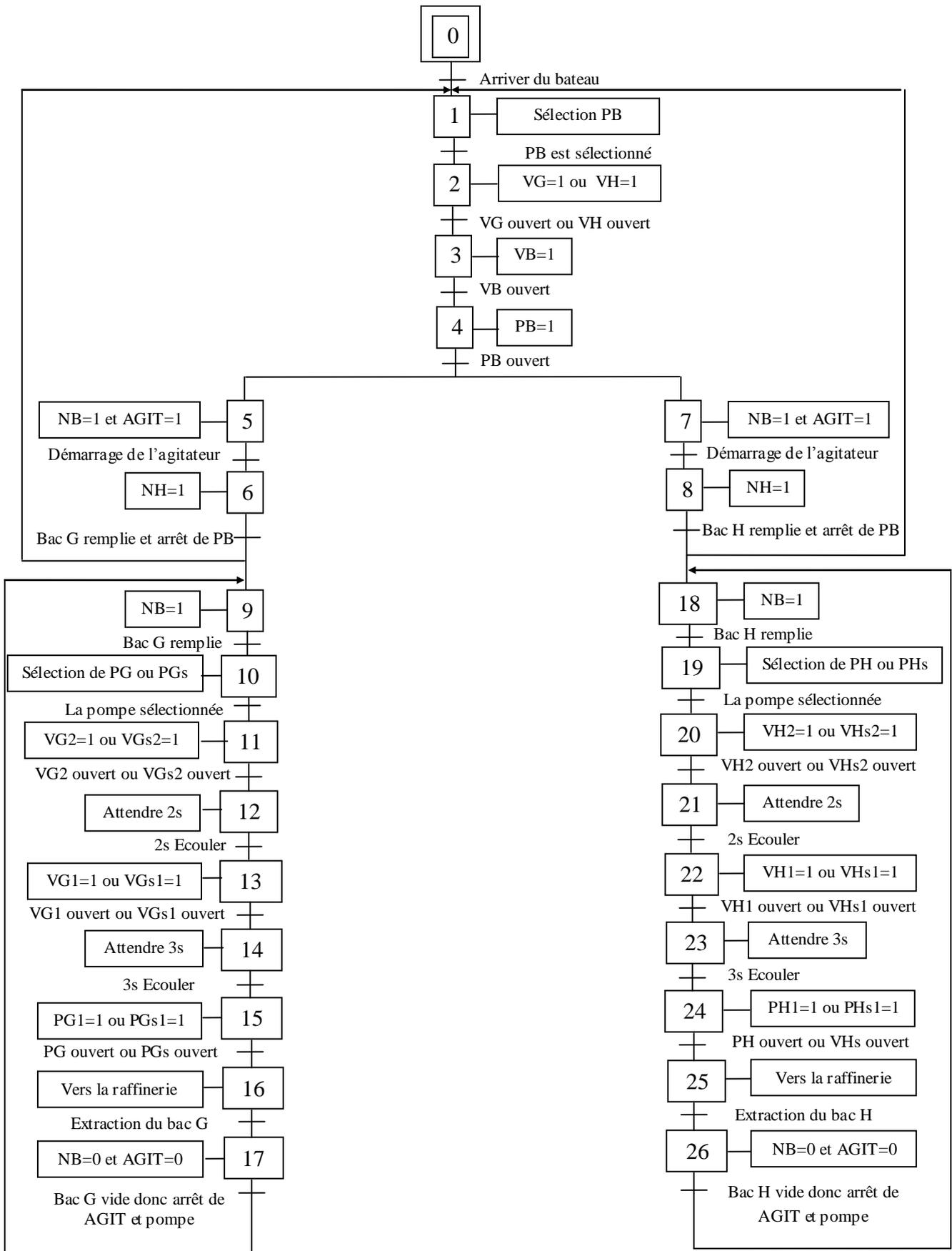


Figure III.2 GRAFCET

❖ Processus de gestion des bacs

- ✓ A l'arrivée du bateau la sélection du bac se fait par l'opérateur d'après le type et la quantité d'huile présente dans les bacs.
- ✓ Après le choix du bac ; on sélectionne la pompe PB si on choisit le bac G la vanne VG s'ouvre (affiche en vert) après 2s puis la vanne VB s'ouvre (affiche en vert) après 3s ; la pompe PB démarre et le remplissage du bac se fait. jusqu'au niveau du capteur NH qui affiche en vert ce qui représente le niveau haut du bac (rempli en entier) et la pompe PB est désélectionnée et les vannes VB et VG se ferment automatiquement.
- ✓ Au besoin , l'opérateur fait l'extarction d'huile brute des bacs.
- ✓ L'extraction d'huile se fait par la sélection de la pompe PG (par exemple) ; la vanne VG2 (vanne d'aspiration) s'ouvre après 2s (affiche en vert) puis la vanne VG1 (vanne de refoulement) s'ouvre après 3s (affiche en vert) donc la pompe PG démarre (affiche en vers). L'huile est envoyée à la raffinerie. L'arrêt de la pompe se fait en la désélectionnent et les vannes VG2 et VG1 s'arrête automatiquement.
- ✓ L'utilisation de la pompe d'extraction PG(par exemple) se fait par le choix de l'opérateur et utilise la pompe PGs en cas de panne de la première pompe ou d'une des vannes.
- ✓ Le même principe s'applique pour le bac H.

III.1.2 Configuration matérielle

La configuration matérielle est la disposition de profilés support ou châssis, de modules, d'appareils de la périphérie décentralisée et de cartouches interface dans une fenêtre de station. Les profilés support ou châssis sont représentés par une table de configuration, dans laquelle on peut enficher un nombre défini de modules, tout comme dans les profilés support ou châssis "réels".

A la mise en route de l'automate programmable, la CPU compare la configuration prévue créée avec STEP7 à la configuration sur site de l'installation. Aussi, les erreurs éventuelles sont-elles immédiatement détectées et signalées.

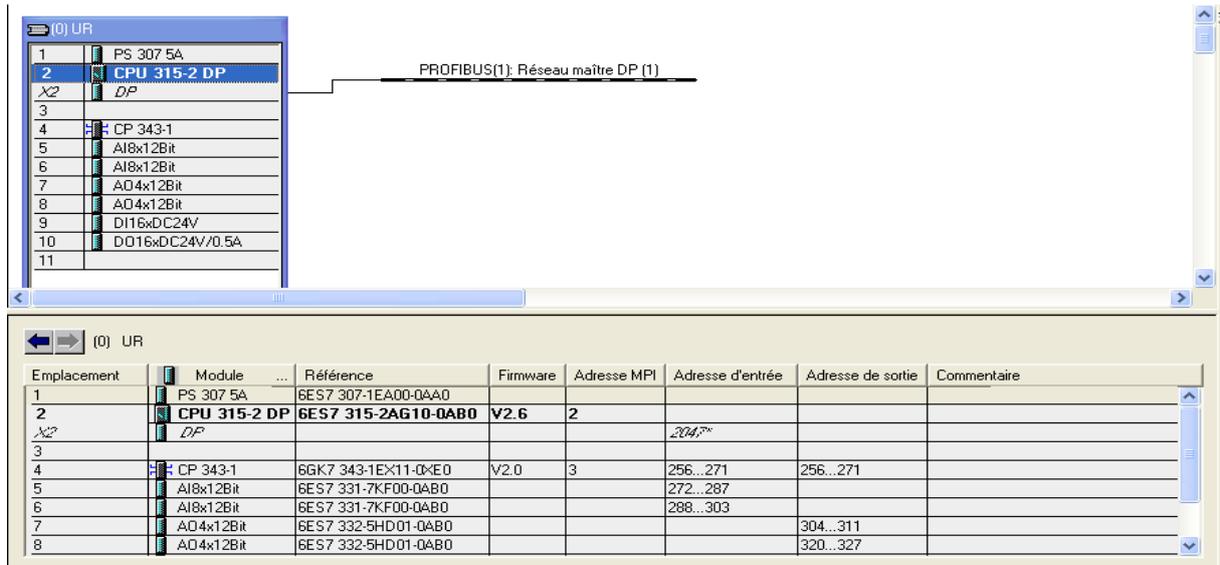


Figure III.3 Configuration matérielle

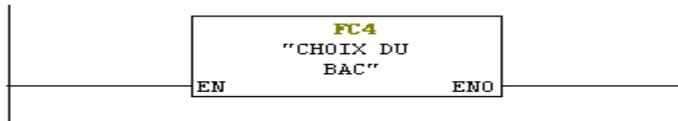
III.1.3 Structure du programme élaboré

Le programme utilisateur qui gère le processus du stockage des huiles brutes est subdivisé en plusieurs fonctions subordonnées au bloc principal **OB1**. Elles sont programmées selon la tâche à exécuter, on cite :

- ✓ **OB1 « programme principal »** : Bloc d'organisation contenant toutes les routines du programme. La CPU appelle ce bloc et exécute les instructions qu'il contient à chaque cycle.

OB1 : Bloc d'organisation pour le traitement cyclique du programme

Réseau 1 : Appel de la fonction FC4 concernant le choix du bac



Réseau 2 : Appel de la fonction FC1 concernant le transmetteur de température



Réseau 3 : Appel de la fonction FC2 concernant le transmetteur de niveau



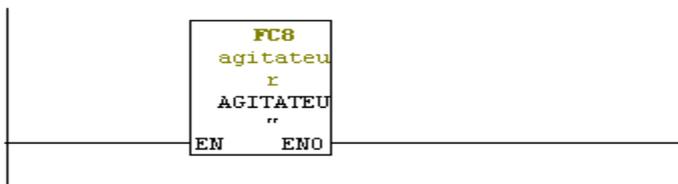
Réseau 4 : Appel de la fonction FC3 concernant les capteurs de niveaux



Réseau 5 : Appel de la fonction FC6 concernant les vannes



Réseau 6 : Appel de la fonction FC8 concernant les agitateurs



Réseau 7 : Appel de la fonction FC9 concernant les pompes

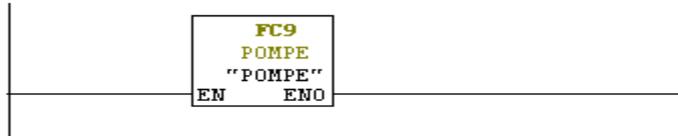


Figure III.4 Les programmes principaux dans l'OB1

- ✓ **FC1** : Ce bloc contient la programmation contenant le détecteur de température des bacs.

FC1 : SONDE DE TEMPERATURE

Réseau 1 : Capteur de température BAC G

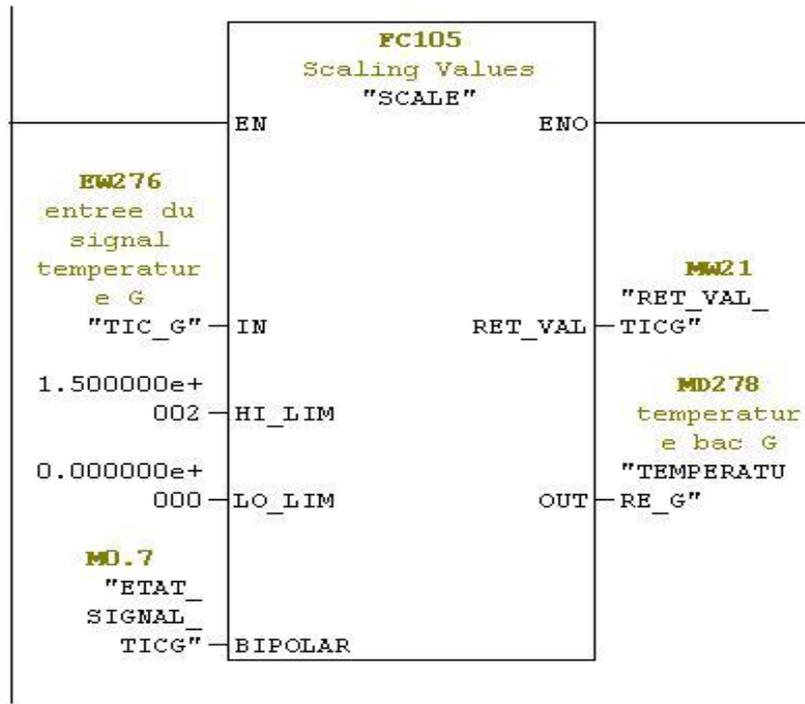


Figure III.5 Programme du transmetteur de niveau LT

✓ **FC2** : Ce bloc contient la programmation contenant le transmetteur de niveaux des bacs.

FC2 : TRANSMETTEUR DE NIVEAU DES BACS G et H 11000T

Réseau 1 : TRANSMETTEUR DE NIVEAU LTG POUR 11000T

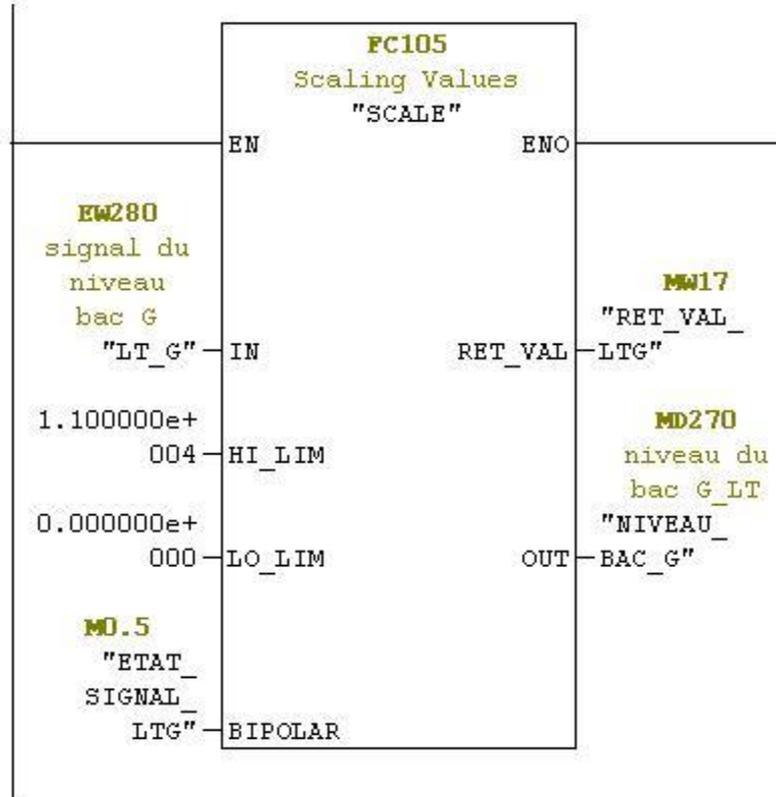


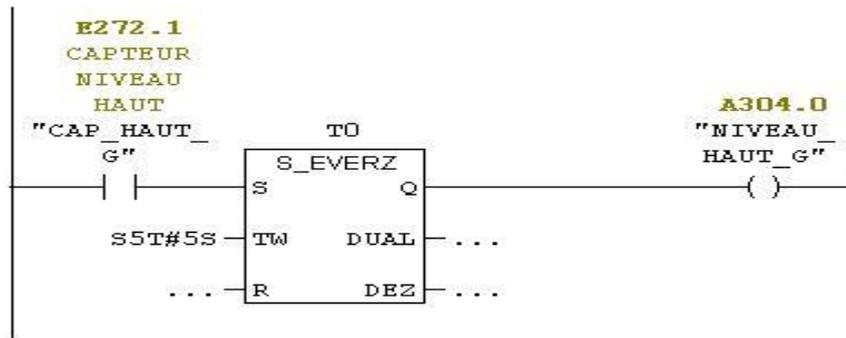
Figure III.6 Programme de transmetteur de température TIC

✓ **FC3** : Ce bloc est destiné pour la programmation des détecteurs de niveaux (bas, moyen, haut).

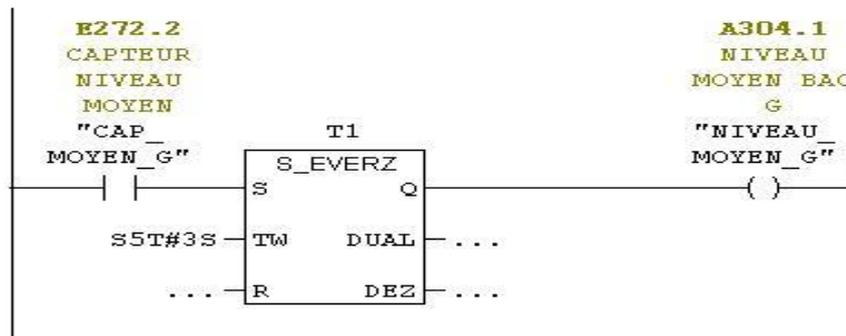
- ✓ NIVEAU_HAUT_G représente le NH
- ✓ NIVEAU_BAS_G représente le NB
- ✓ NIVEAU_MOYEN_G représente le NM

FC3 : CAPTEUR DE NIVEAUX DES BACS G ET H

Réseau 1 : NIVEAUX HAUT BAC G



Réseau 2 : NIVEAU MOYEN BAC G



Réseau 3 : NIVEAU BAS bac G

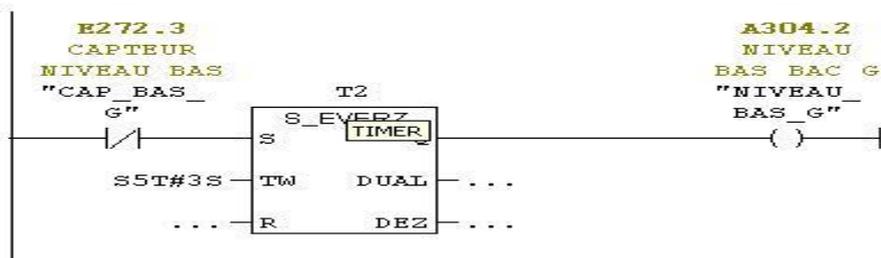


Figure III.7 Programme du capteur de niveau

- ✓ FC4 : Ce bloc contient le programme destiné aux choix du bac à utiliser par l'opérateur et le choix des pompes à utiliser.

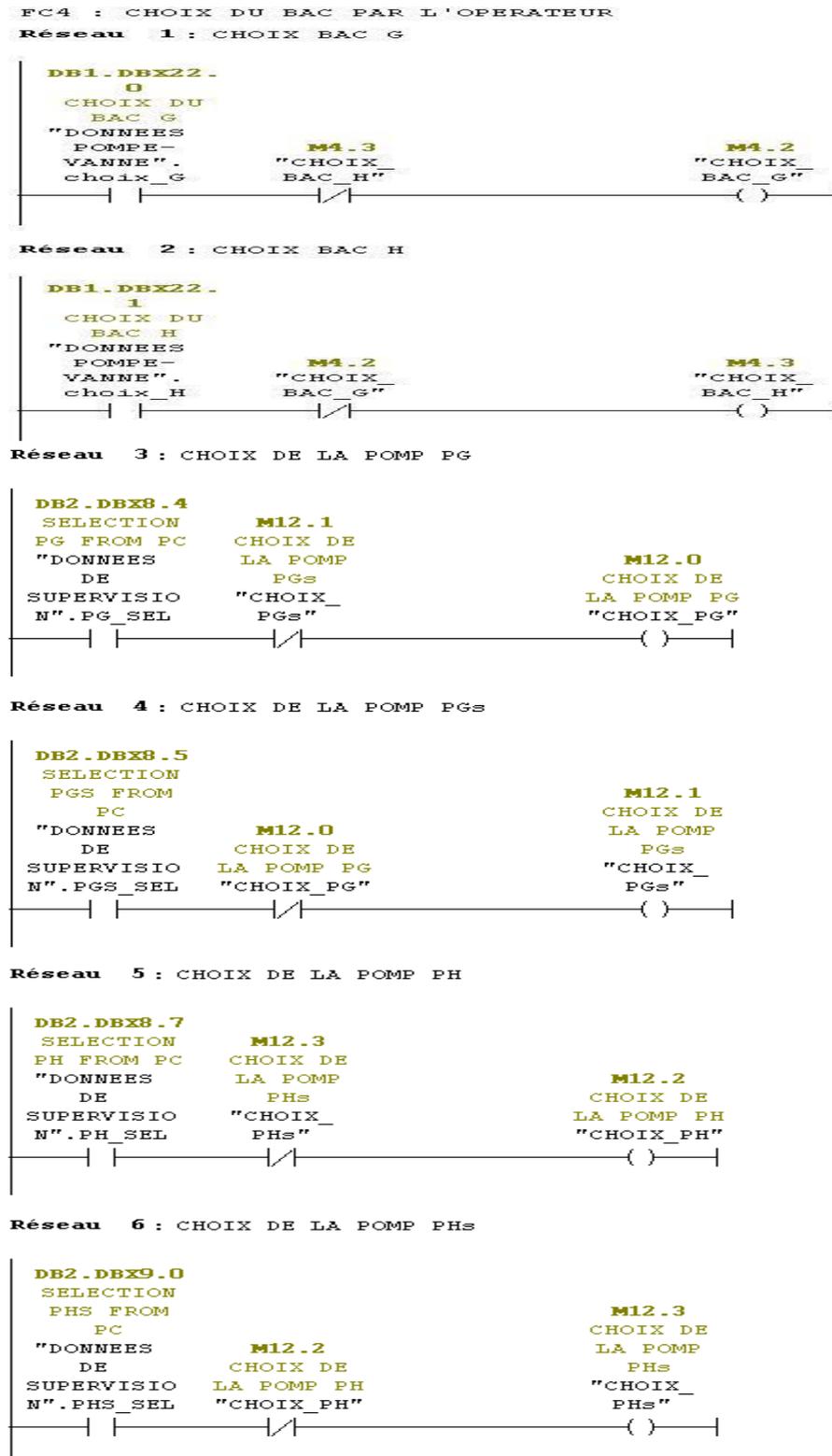


Figure III.8 Programme choix du bac et pompe

- ✓ **FC5** : Pour programmer une vanne, il suffit d'appeler ce bloc qui accepte les conditions d'ouverture et de fermeture comme argument. Le bloc FC5 est appelé par FC6.

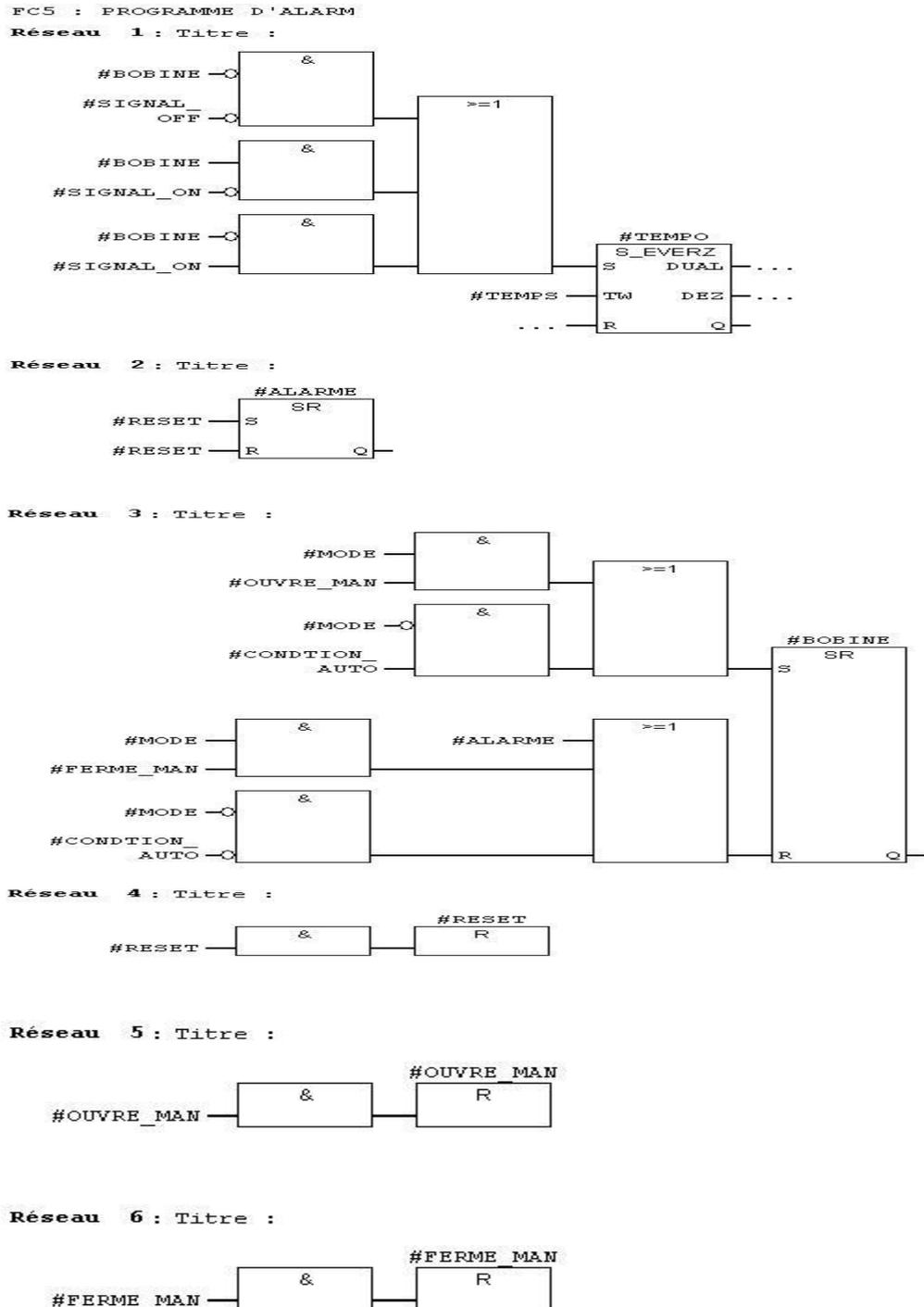


Figure III.9 Programme contrôle vanne

- ✓ **FC6** : Ce bloc contient toute l'instruction de commande d'ouverture et de fermeture de toutes les vannes utilisées.

FC6 : PROGRAMMATION DES VANNE

Réseau 1 : VANNE BAC G (VG)

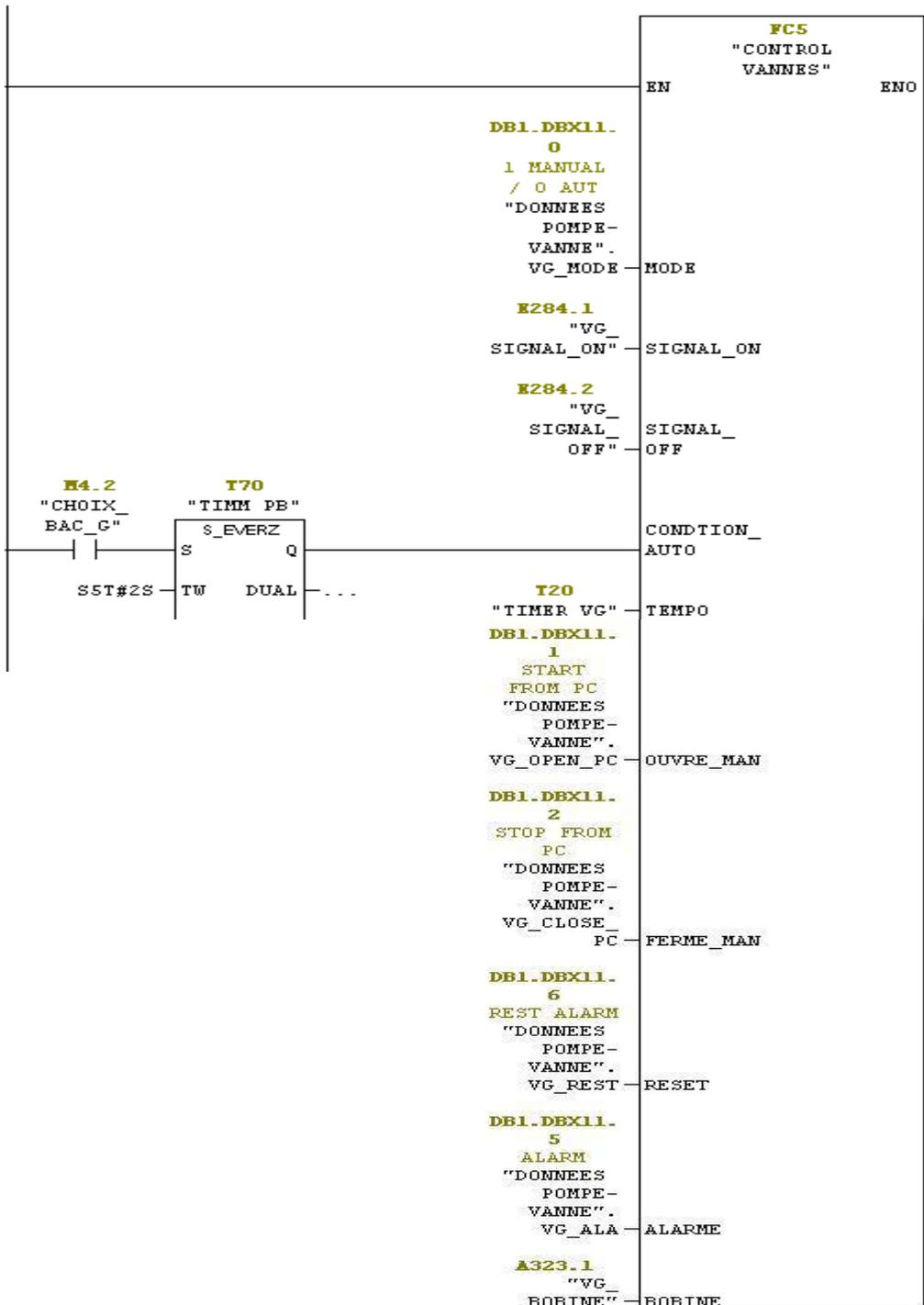


Figure III.10 Programme appel vanne

- ✓ **FC77** : Pour programmer un agitateur ou une pompe, il suffit d'appeler ce bloc qui accepte les conditions d'arrêt et démarrage comme arguments. le bloc FC77 est appelé par le FC 9 et FC8.

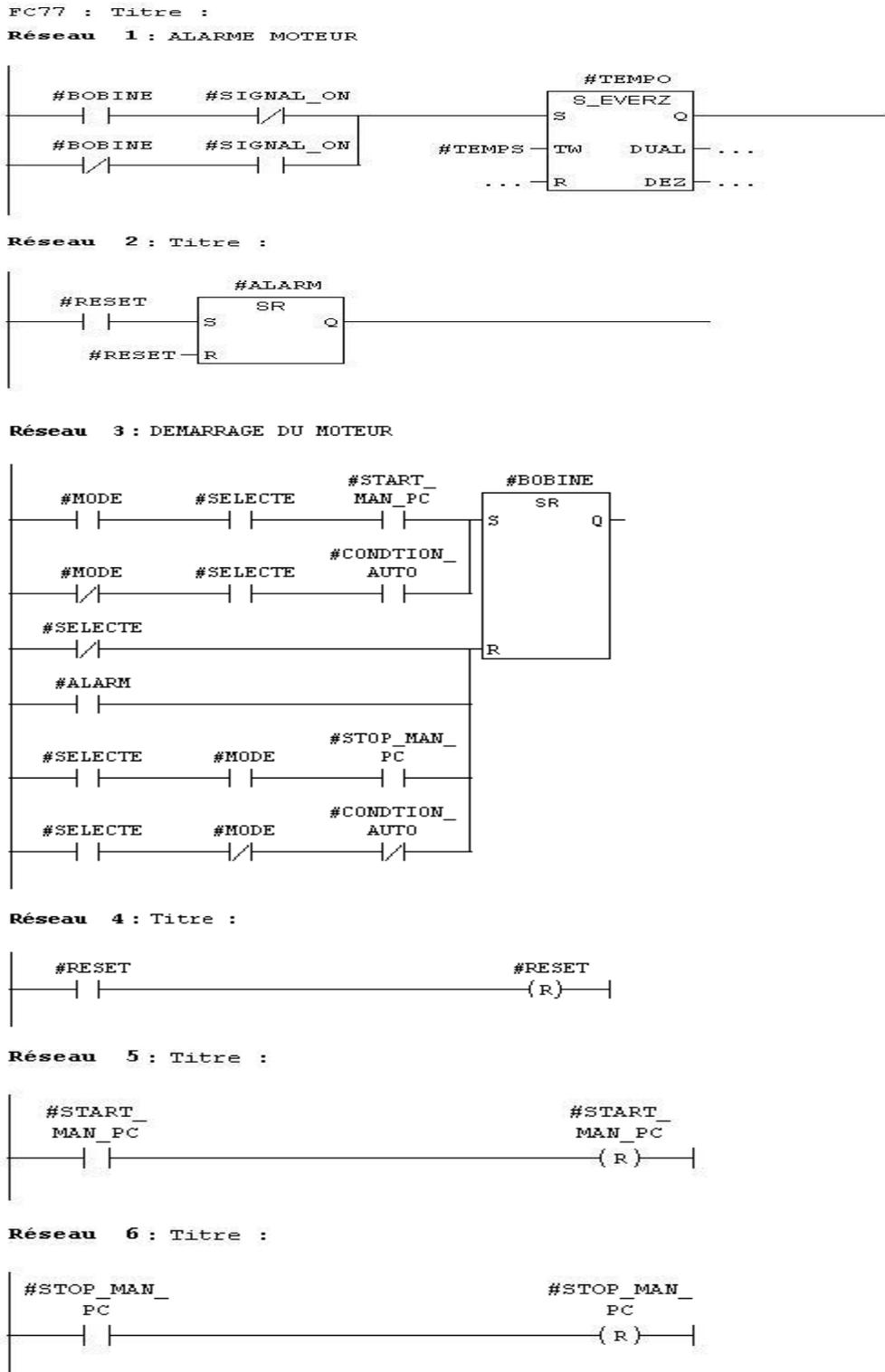


Figure III.11 Programme contrôle agitateur et pompe

- ✓ **FC8** : Bloc contient toute l'instruction de commande de démarrage et d'arrêt tous les agitateurs utilisés.



Figure III.12 Programme appel agitateur

✓ **FC9** : Ce bloc contient toute l'instruction de commande de démarrage et d'arrêt de toutes les pompes utilisées.

FC9 : appel pompes des bac G et H
 Réseau 1: APPEL POMPE PB

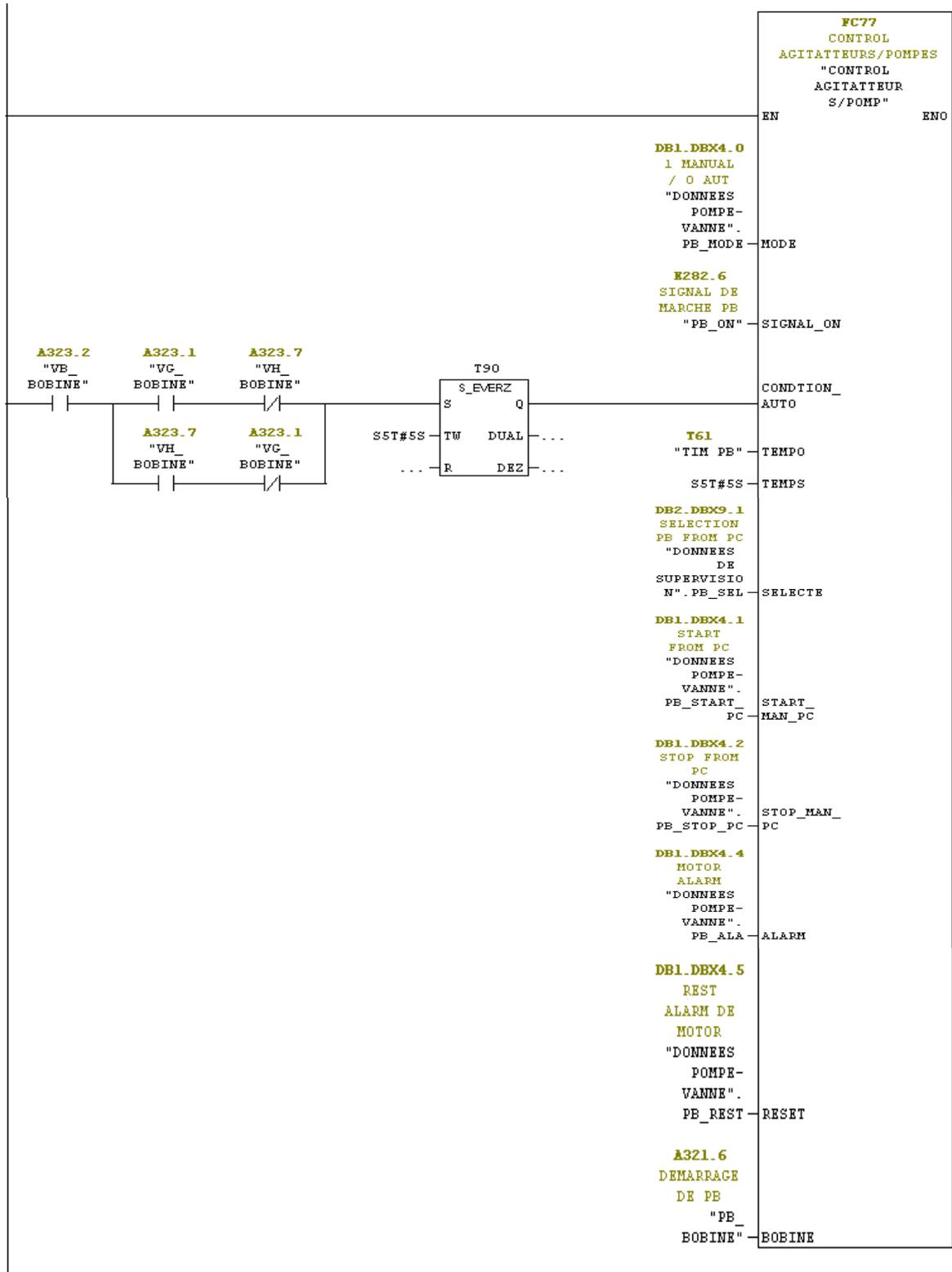


Figure III.13 Programme appel pompe

✓ **DB1 et DB2** : Dans ces deux blocs de données on mémorise les données nécessaires au traitement du programme et les données affectées a chaque bloc fonctionnel.

Adresse	Nom	Type	Valeur initiale	Commentaire
0.0		STRUCT		
+0.0	PG_MODE	BOOL	FALSE	1 MANUAL / 0 AUT
+0.1	PG_START_PC	BOOL	FALSE	START FROM PC
+0.2	PG_STOP_PC	BOOL	FALSE	STOP FROM PC
+0.3	PG_SIGNAL_PC	BOOL	FALSE	SIGNAL DE MARCHE DE P6
+0.4	PG_ALA	BOOL	FALSE	MOTOR ALARM
+0.5	PG_REST	BOOL	FALSE	REST ALARM DE MOTOR
+0.6	PG_RESERVE1	BOOL	FALSE	
+0.7	PG_RESERVE2	BOOL	FALSE	
+1.0	PGS_MODE	BOOL	FALSE	1 MANUAL / 0 AUT
+1.1	PGS_START_PC	BOOL	FALSE	START FROM PC
+1.2	PGS_STOP_PC	BOOL	FALSE	STOP FROM PC
+1.3	PGS_SIGNAL_PC	BOOL	FALSE	SIGNAL DE MARCHE DE P6S
+1.4	PGS_ALA	BOOL	FALSE	MOTOR ALARM
+1.5	PGS_REST	BOOL	FALSE	REST ALARM DE MOTOR
+1.6	PGS_RESERVE1	BOOL	FALSE	
+1.7	PGS_RESERVE2	BOOL	FALSE	
+2.0	PH_MODE	BOOL	FALSE	1 MANUAL / 0 AUT
+2.1	PH_START_PC	BOOL	FALSE	START FROM PC
+2.2	PH_STOP_PC	BOOL	FALSE	STOP FROM PC
+2.3	PH_SIGNAL_PC	BOOL	FALSE	SIGNAL DE MARCHE DE P7
+2.4	PH_ALA	BOOL	FALSE	MOTOR ALARM
+2.5	PH_REST	BOOL	FALSE	REST ALARM DE MOTOR
+2.6	PH_RESERVE1	BOOL	FALSE	
+2.7	PH_RESERVE2	BOOL	FALSE	
+3.0	PHs_MODE	BOOL	FALSE	1 MANUAL / 0 AUT
+19.0	VH1_MODE	BOOL	FALSE	
+19.1	VH1_OPEN_PC	BOOL	FALSE	
+19.2	VH1_CLOSE_PC	BOOL	FALSE	
+19.3	VH1_SIGNAL_ON	BOOL	FALSE	
+19.4	VH1_SIGNAL_OFF	BOOL	FALSE	
+19.5	VH1_ALA	BOOL	FALSE	
+19.6	VH1_REST	BOOL	FALSE	
+19.7	VH1_RESERVE1	BOOL	FALSE	
+20.0	VHs2_MODE	BOOL	FALSE	
+20.1	VHs2_OPEN_PC	BOOL	FALSE	
+20.2	VHs2_CLOSE_PC	BOOL	FALSE	
+20.3	VHs2_SIGNAL_ON	BOOL	FALSE	
+20.4	VHs2_SIGNAL_OFF	BOOL	FALSE	
+20.5	VHs2_ALA	BOOL	FALSE	
+20.6	VHs2_REST	BOOL	FALSE	
+20.7	VHs2_RESERVE1	BOOL	FALSE	
+21.0	VHs1_MODE	BOOL	FALSE	
+21.1	VHs1_OPEN_PC	BOOL	FALSE	
+21.2	VHs1_CLOSE_PC	BOOL	FALSE	
+21.3	VHs1_SIGNAL_ON	BOOL	FALSE	
+21.4	VHs1_SIGNAL_OFF	BOOL	FALSE	
+21.5	VHs1_ALA	BOOL	FALSE	
+21.6	VHs1_REST	BOOL	FALSE	
+21.7	VHs1_RESERVE1	BOOL	FALSE	
+22.0	choix_G	BOOL	FALSE	CHOIX DU BAC G
+22.1	choix_H	BOOL	FALSE	CHOIX DU BAC H
=24.0		END_STRUCT		

Figure III.14 Table des blocs de donnée DB1

+8.3	RESERVE18	BOOL	FALSE	RESERVE
+8.4	PG_SEL	BOOL	FALSE	SELECTION PG FROM PC
+8.5	PGS_SEL	BOOL	FALSE	SELECTION PGS FROM PC
+8.6	RESERVE	BOOL	FALSE	
+8.7	PH_SEL	BOOL	FALSE	SELECTION PH FROM PC
+9.0	PHS_SEL	BOOL	FALSE	SELECTION PHS FROM PC
+9.1	PB_SEL	BOOL	FALSE	SELECTION PB FROM PC
+9.2	AGIT1_SEL	BOOL	FALSE	SELECTION AGIT1 FROM PC
+9.3	AGIT2_SEL	BOOL	FALSE	SELECTION AGIT2 FROM PC
+9.4	AGIT3_SEL	BOOL	FALSE	SELECTION AGIT3 FROM PC
+9.5	AGIT4_SEL	BOOL	FALSE	SELECTION AGIT4 FROM PC
+9.6	AGIT5_SEL	BOOL	FALSE	SELECTION AGIT5 FROM PC
+9.7	AGIT6_SEL	BOOL	FALSE	SELECTION AGIT6 FROM PC
=10.0		END_STRUCT		

Figure III.15 Table des blocs de donnée DB2

- ✓ **FC105** : Pour programmer les capteurs de température et le transmetteur de niveaux, il suffit d'appeler ce bloc. Le FC105 est appelé par F1 et FC2.
- ✓ **FC99** : Ce bloc contient un programme qui nous permet de faire la simulation.

III.2 Simulation du programme élaboré

III.2.1 Ouverture de l'AP S7-PLCSIM

Pour ouvrir le simulateur on procède comme suit :

- ❖ Démarrer le gestionnaire de projet SIMATIC en cliquant sur son icône.
- ❖ Lancer l'AP S7-PLCSIM en cliquant sur le bouton d'activation/désactivation de la simulation qui se trouve sur la barre d'outils du gestionnaire de projets SIMATIC, comme la montre la figure ou en sélectionnant la commande outils simulation des modules.
- ❖ La fenêtre de l'application S7-PLCSIM s'ouvre avec une fenêtre CPU par défaut

Bouton d'activation/désactivation de la simulation

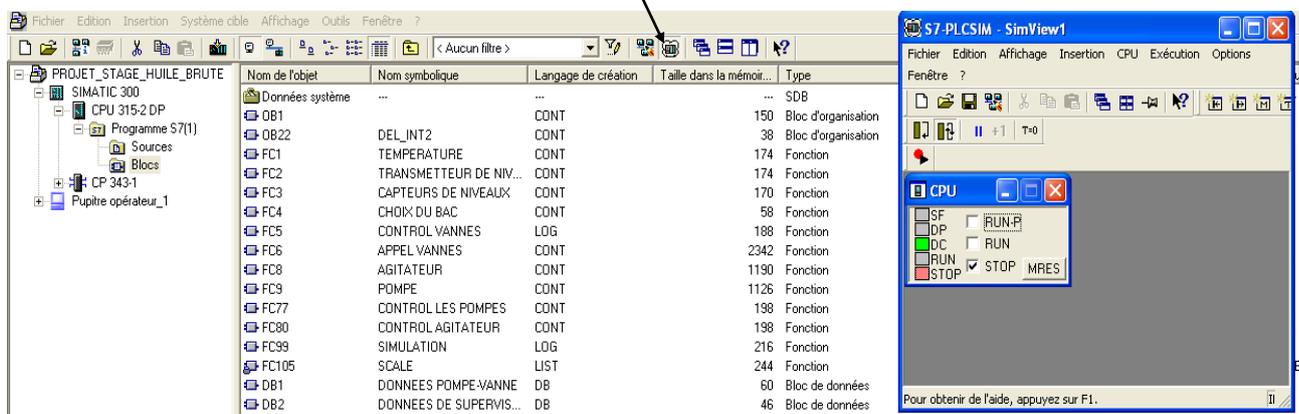


Figure III.16 Ouverture de l'API de Simulation

III.2.2 Chargement de programme

On procède de la manière suivante pour charger le programme édité :

- ❖ Pour ouvrir le projet « PROJET_STAGE_HUILES_BRUTES », on utilise la commande **Fichier ouvrir projet** du gestionnaire de projets SIMATIC.
- ❖ Parcourir la boîte de recherche jusqu'au classeur des blocs. La structure hiérarchique du projet est représentée à la figure .

Pour charger le classeur des blocs dans la CPU de simulation, choisir la commande **Système cible charger** ou cliquer sur le bouton de chargement.

III.2.3 Configuration de l'API de simulation

III.2.3.1 Création des fenêtres pour l'exemple de programme

Le programme utilise plusieurs entrées, sorties, mémentos et temporisation ; durant l'exécution du programme, on peut utiliser des fenêtres pour mettre les entrées à 1 ou à 0 et visualiser les valeurs des temporisations et changement des sorties. Les fenêtres utilisées dans le programme sont représentées dans la figure. Pour créer les diverses fenêtres, on procède de la manière suivante :

- ❖ Créer une fenêtre permettant de modifier l'état des entrées intervenant dans le programme :
 - Choisir la commande **Insertion variable d'entrée** ou directement à partir de la barre d'outils.
- ❖ Créer une fenêtre permettant de modifier l'état des sorties intervenant dans le programme :
 - Choisir la commande **Insertion variable de sortie** ou directement à partir de la barre d'outils.
- ❖ Créer une fenêtre pour les mémentos intervenant dans le programme :
 - Choisir la commande **Insertion Mémento** ou directement à partir de la barre d'outil.

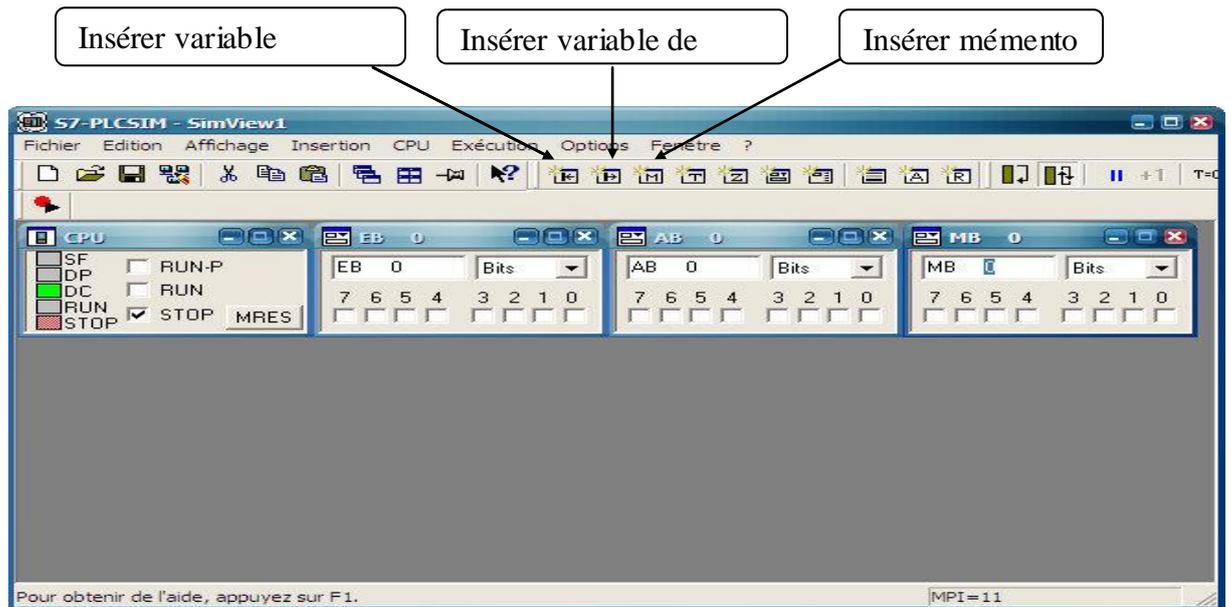


Figure III.17 Fenêtre pour le programme simulé

Une fois l'exemple de programme est chargé dans la CPU, on peut exécuter le programme il faut s'assurer préalablement que le cycle continu est sélectionné comme mode de l'exécution.

Pour cela choisir la commande **exécution mode cycle continu** (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) ou cliquer dans le bouton correspondant dans la barre d'outil

Pour exécuter le programme utilisateur chargé, on procède de la manière suivante :

- ❖ On coche la case RUN-P de la CPU pour la mettre en marche et démarrer ainsi le cycle de l'exécution du programme.
- ❖ On force l'état des entrées, memento,...etc. et cela en cochant le bit correspondant dans les fenêtres des variables créés préalablement puis on constate l'évolution des états des sorties a travers la fenêtre de sortie précédemment créée.

III.3 Simulation du projet

Le logiciel WinCC Flexible permet de simuler et tester le projet en l'absence de l'automate. Le simulateur RUNTIME permet de vérifier le bon fonctionnement des images configurées, des objets d'images ; et détecter des erreurs logiques à l'aide de la commande « démarrer le système Runtime du simulateur ».

La figure suivante représente la simulation sur WinCC flexible. La mise en marche du projet se fait comme suit :

- ❖ A l'arrivée du bateau la sélection du bac se fait par l'opérateur d'après le type et la quantité d'huile présente dans les bacs.
- ❖ Après le choix du bac ; on sélectionne la pompe PB si on choisi le bac G la vanne VG s'ouvre (affiche en vert) après 2s puis la vanne VB s'ouvre (affiche en vert) après 3s ; la pompe PB démarre et le remplissage du bac se fait. jusqu'au niveau du capteur NH qui affiche en vert ce qui représente le niveau haut du bac (rempli en entier) et la pompe PB est désélectionnée et les vannes VB et VG se ferment automatiquement.
- ❖ Au besoin , l'opérateur fait l'extarction d'huile brute des bacs.
- ❖ L'extraction d'huile se fait par la sélection de la pompe PG (par exemple) ; la vanne VG2 (vanne d'aspiration) s'ouvre après 2s (affiche en vert) puis la vanne VG1 (vanne de refoulement) s'ouvre après 3s (affiche en vert) donc la pompe PG démarre (affiche en vers). L'huile est envoyée à la raffierie. L'arrêt de la pompe se fait en la désélectionnent et les vannes VG2 et VG1 s'arrête automatiquement.
- ❖ L'utilisation de la pompe d'extraction PG(par exemple) se fait par le choix de l'opérateur et utilise la pompe PGs en cas de panne de la premieée pompe ou d'une des vannes.
- ❖ Le même principe s'applique pour le bac H.

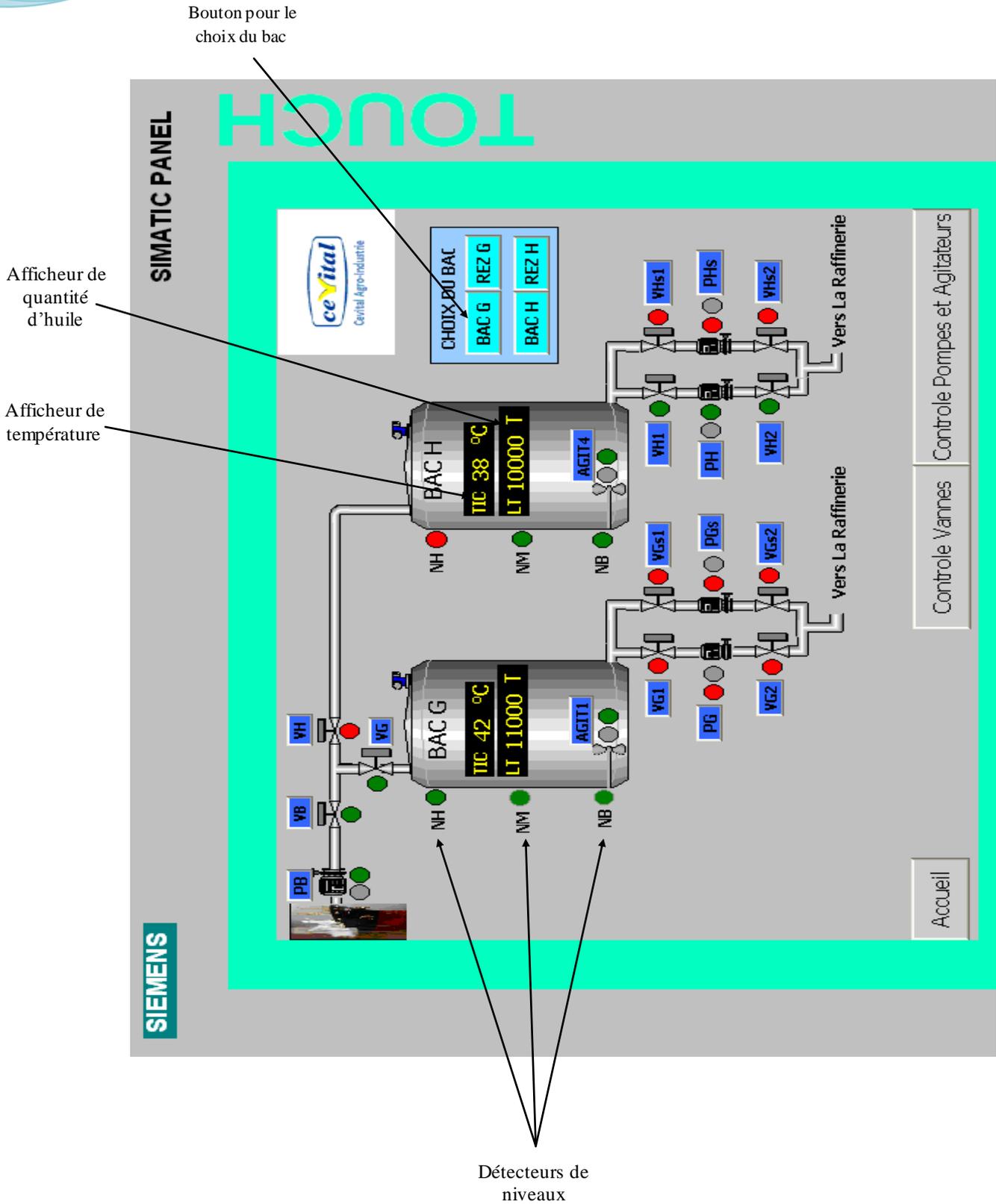


Figure III.18 Simulation du projet

Conclusion

Ce chapitre représente les programmes élaborés sous le logiciel STEP7 pour la bonne gestion du projet.

Le programme de supervision élaboré, et simulé nous a permis de tester le projet et de vérifier les programmes faits en STEP7 avant qu'ils soient chargés dans l'automate.

Les résultats que nous avons trouvés, après simulation sont très intéressants et leur mise en application sur le site de CEVITAL est imminente.

Le travail que nous avons mené au sein de la raffinerie d'huile de CEVITAL nous a permis d'une part d'acquérir des connaissances techniques et pratiques qui viennent compléter les enseignements théoriques acquis et d'autre part d'avoir la possibilité de nous familiariser avec le milieu industriel et ses exigences.

Au cours de cette étude, nous avons pu constater le rôle important des API et leurs conséquences sur la sécurité et le travail. Le deuxième constat, est que la mise au point d'un système de contrôle et de supervision doit répondre à toutes les exigences dictées par le cahier de charge. Il est donc nécessaire de rassembler toutes les informations concernant la composition de l'installation et le déroulement des processus.

Pour atteindre l'objectif de notre projet, nous avons commencé par prendre connaissance de l'installation qui est les bacs de stockage des huiles brutes existants puis identifier les éléments les constituants.

Nous avons commencé par une étude structurelle et fonctionnelle de l'installation pour l'acquisition de données du problème. L'étude d'état de la gestion des bacs d'huile brute.

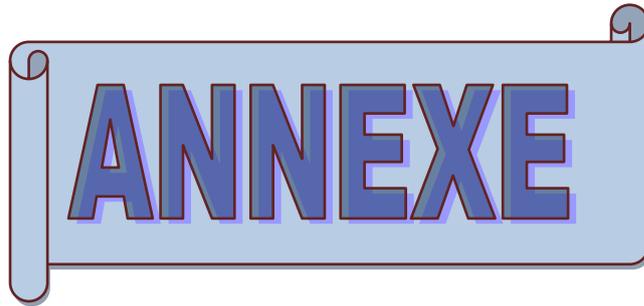
Ensuite vient l'élaboration d'un système de commande où nous avons développé un programme STEP7 qui sera chargé dans l'automate programmable qui commandera l'installation.

Nous avons terminé notre projet par l'introduction d'un système de supervision pour garantir l'interface Homme-Machine et assurer le contrôle et la surveillance de l'installation.

Enfin, nous espérons que notre travail permettra aux automaticiens et ingénieurs de la raffinerie d'huile de CEVITAL de trouver les réponses aux problèmes posés, et aux étudiants des futures promotions de mener des études d'optimisation entre autres.

Référence Bibliographique

- [1] Documentation interne de l'entreprise CEVITAL
- [2] Auder Simon, « Automate programmable industrielle », édition l'ELAN, Liège 1991
- [3] Philippe le BRUN, « Automate programmable », Lycée louis ARNAUD, Strasbourg, édition décembre 1999
- [4] Yves G.Palau, « L'automate programmable », Paris 1983
- [5] Paul Brard, Gerard Colombari, « le GRAFCET, outil de description des automatismes séquentiels », édition technique de l'ingénieur, Paris 1999
- [6] Pierre Bonnet, « Introduction à la supervision », Université Lille1, Mémoire Master SMaRT, Novembre 2010
- [7] Siemens, « Simatic STEP7 V5.1 Getting started », édition 08/2000
- [8] Siemens, « WinCC Simatic HMI Getting started », édition Mars 2000
- [9] IRMECHE Belaid, DJOUDER Abdenour, « Elaboration d'un Système de commande et de supervision pour la gestion des bacs d'huile finie à l'unité de conditionnement CEVITAL », Université M'HAMED BOUGUERRA BOUMERDAS, Année 2007
- [10] Mebarki Rachid, « Automatisation et supervision d'une section d'huile finie CEVITAL », Université A-MIRA BEJAIA, Année 13/14
- [11] <http://www.endress.com/en/Tailor-made-field-instrumentation/level-measurement/Radar-measurement-Micropilot-FMR240>
- [12] <https://www.yumpu.com/fr/document/view/16719540/cours-le-grafcet-lycee-ferry-versaillesfr>
- [14] <http://www.endress.com/en/Tailor-made-field-instrumentation/Temperature-measurement-thermometers-transmitters/Resistance-thermometer-Pt100-for-hazardous-areas-TR66>
- [13] <http://www.positron-libre.com/cours/electrotechnique/demarrage-moteur/schema-demarrage-etoile-triangle.php>



La table des mnémoniques

Résumé

Ce mémoire représente une méthodologie générale pour l'automatisation d'un système industriel. Il a été question d'une étude détaillée d'une section de la raffinerie qui est le stockage des huiles brutes ; il a permis de modéliser son fonctionnement suite au programme élaboré sur le logiciel STEP7 qui une fois transféré dans l'automate S7-300 va gérer le fonctionnement automatique du procédé.

Vous trouverez également une description détaillée sur les automates programmables industriels et plus précisément le S7-300 de la firme SIEMENS.

Une grande partie de ce travail est consacrée à la description du logiciel STEP7 en mettant en avant les étapes à suivre pour la création d'un projet d'automatisation, la configuration matériel, l'élaboration du programme et sa simulation.

Une supervision du projet a été créée pour assurer le contrôle à distance de l'installation avec WinCC Flexible.

Abstract

This memory represents a general methodology for automating an industrial system. It was question of detailed study of a section refinery which is storing crude; which made it possible to model its operation consequently a program was developed on the software STEP7 which once transferred in the S7-300. It will manage the automatic operation of the process.

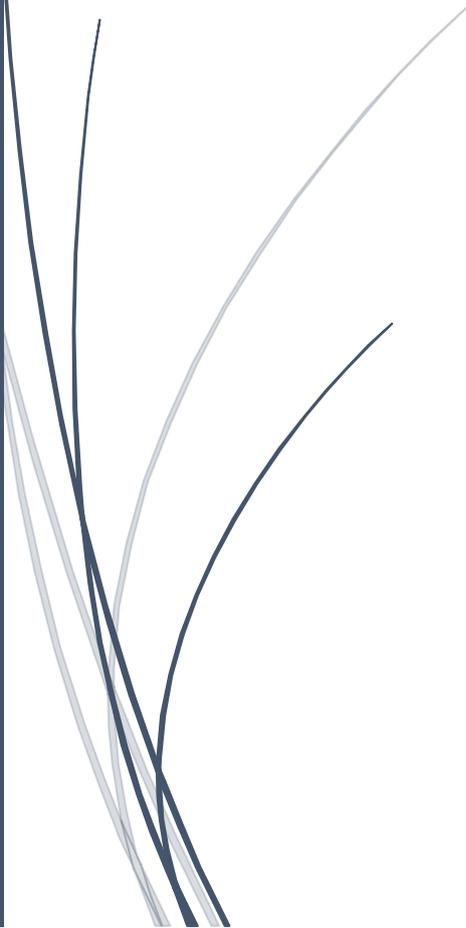
There is also a detailed description on industrial programmable logic controllers and precisely S7-300 of the SIEMENS firm.

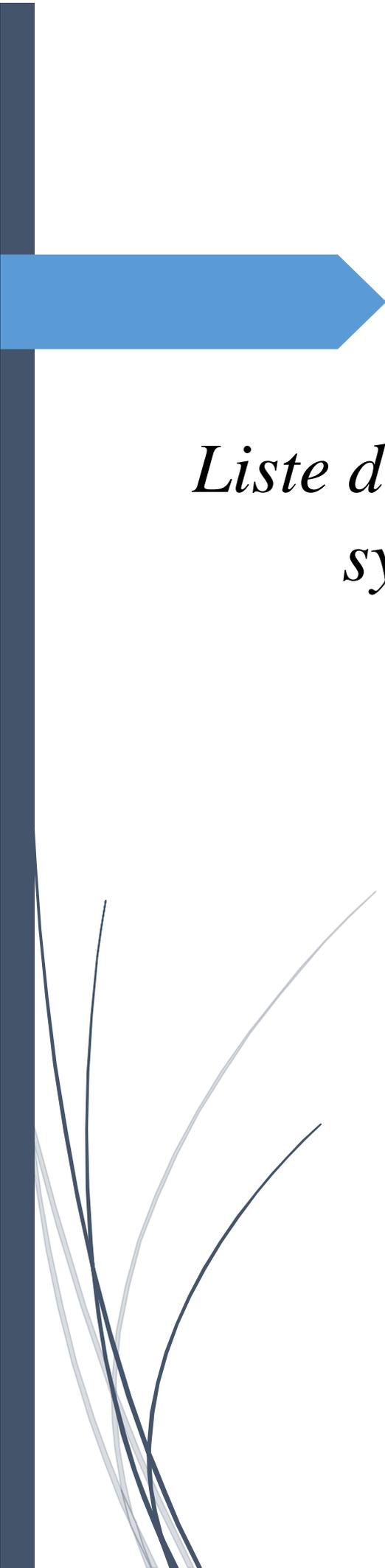
Much of this work is devoted to the description of the STEP7 software by putting forward the steps for creating an automation project, the hardware configuration, program development and simulation.

The Project supervision was created to ensure the remote control of the system with WinCC flexible.

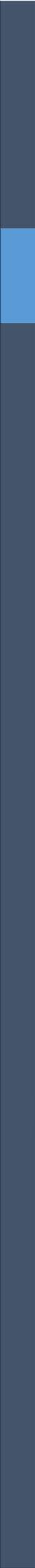


Introduction Générale



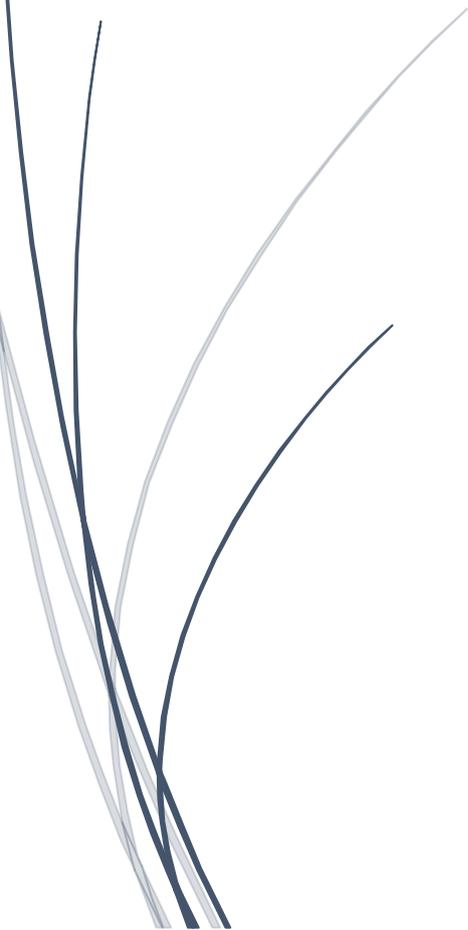


*Liste des abréviations et
symboles*



Chapitre I

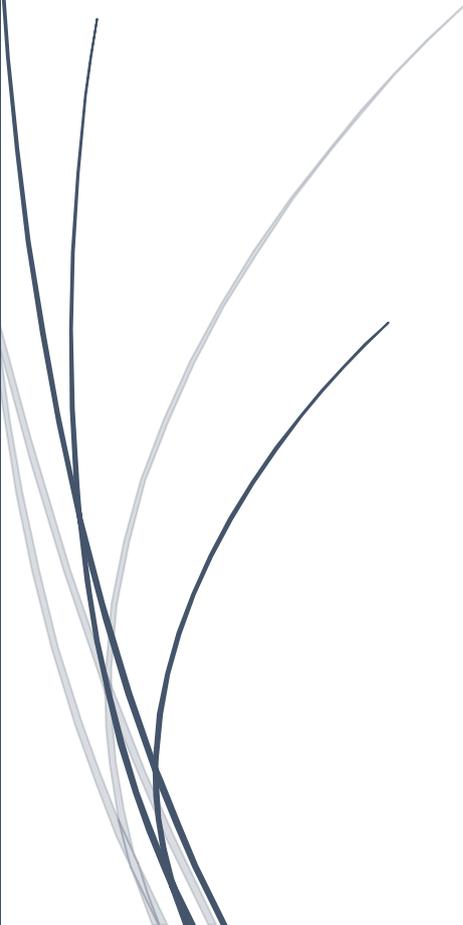
*Présentation et étude du
processus*





Chapitre II

Automates API et supervision WinCC





Chapitre III

Résultats et interprétation



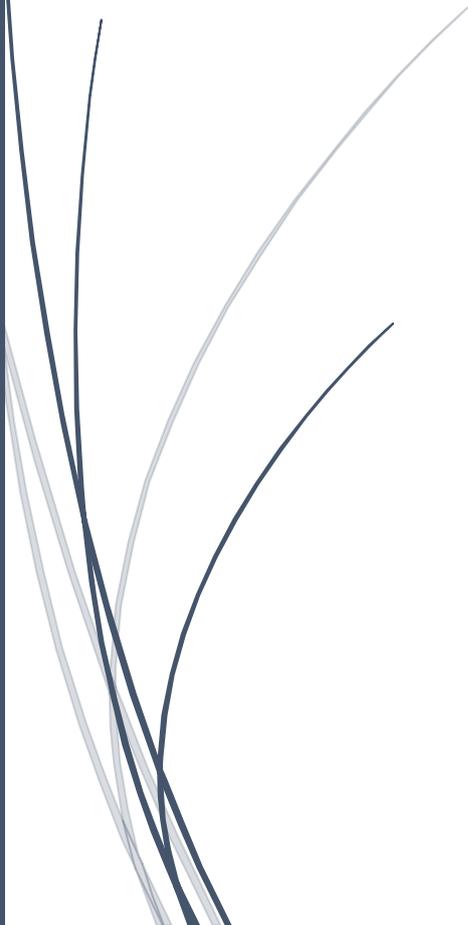


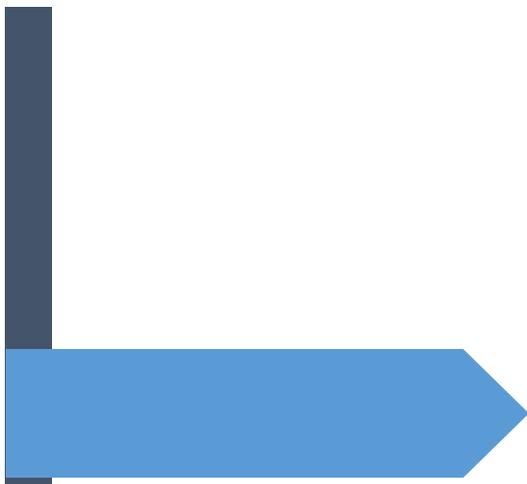
Conclusion Générale



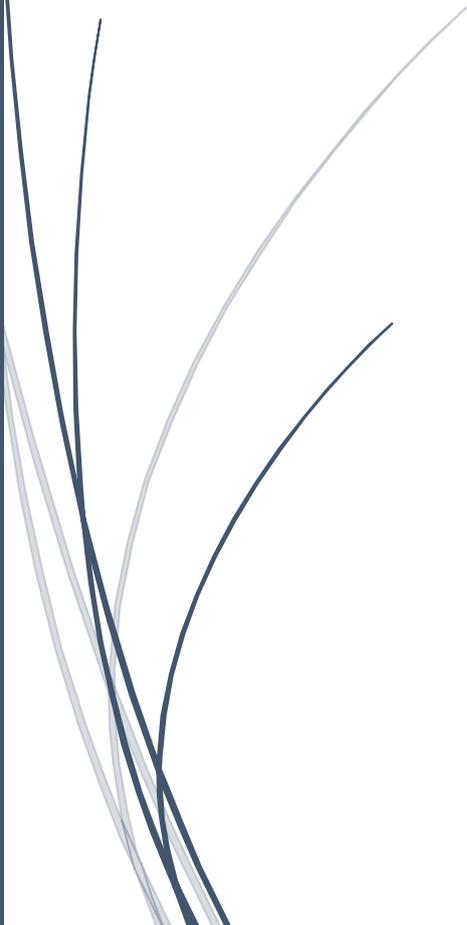


Liste des Figures





Bibliographie



Etat	Mnémonique	Opéran	Type de d	Commentaire
	NIVEAU_HAUT_G	A 304.0	BOOL	NIVEAU HAUT BAC G
	NIVEAU_MOYEN_G	A 304.1	BOOL	NIVEAU MOYEN BAC G
	NIVEAU_BAS_G	A 304.2	BOOL	NIVEAU BAS BAC G
	NIVEAU_HAUT_H	A 304.3	BOOL	NIVEAU HAUT BAC H
	NIVEAU_MOYEN_H	A 304.4	BOOL	NIVEAU MOYEN BAC H
	NIVEAU_BAS_H	A 304.5	BOOL	NIVEAU BAS BAC H
	PB_BOBINE	A 321.6	BOOL	DEMARRAGE DE PB
	PG-BOBINE	A 321.7	BOOL	DEMARRAGE DE PG
	PGS-BOBINE	A 322.0	BOOL	DEMARRAGE DE PGS
	PH_BOBINE	A 322.1	BOOL	DEMARRAGE DE PH
	PHs_BOBINE	A 322.2	BOOL	DEMARRAGE DE PHs
	AGIT1_BOBINE	A 322.3	BOOL	DEMARRAGE DE AGIT1
	AGIT2_BOBINE	A 322.4	BOOL	DEMARRAGE DE AGIT2
	AGIT3_BOBINE	A 322.5	BOOL	DEMARRAGE DE AGIT3
	AGIT4_BOBINE	A 322.6	BOOL	DEMARRAGE DE AGIT4
	AGIT5_BOBINE	A 322.7	BOOL	DEMARRAGE DE AGIT5
	AGIT6_BOBINE	A 323.0	BOOL	DEMARRAGE DE AGIT6
	VG_BOBINE	A 323.1	BOOL	DEMARRAGE DE VG
	VB_BOBINE	A 323.2	BOOL	DEMARRAGE DE VB
	VG2_BOBINE	A 323.3	BOOL	DEMARRAGE DE VG2
	VG1_BOBINE	A 323.4	BOOL	DEMARRAGE DE VG1
	VGs2_BOBINE	A 323.5	BOOL	DEMARRAGE DE VGs2
	VGs1_BOBINE	A 323.6	BOOL	DEMARRAGE DE VGs1
	VH_BOBINE	A 323.7	BOOL	DEMARRAGE DE VH
	VH2_BOBINE	A 324.0	BOOL	DEMARRAGE DE VH2
	VH1_BOBINE	A 324.1	BOOL	DEMARRAGE DE VH1
	VHs2_BOBINE	A 324.2	BOOL	DEMARRAGE DE VHs2
	VHs1_BOBINE	A 324.3	BOOL	DEMARRAGE DE VHs1
	DONNEES POMPE-VANNE	DB 1 DB 1		BLOC DE DONNEES
	DONNEES DE SUPERVISION	DB 2 DB 2		BLOC DE DONNEES DE SUPERVISION
	CAP_HAUT_G	E 272.1	BOOL	CAPTEUR NIVEAU HAUT
	CAP_MOYEN_G	E 272.2	BOOL	CAPTEUR NIVEAU MOYEN
	CAP_BAS_G	E 272.3	BOOL	CAPTEUR NIVEAU BAS
	CAP_HAUT_H	E 272.4	BOOL	CAPTEUR NIVEAU HAUT
	CAP_MOYEN_H	E 272.5	BOOL	CAPTEUR DE NIVEAU MOYEN

Etat	Mnémonique	Opéran	Type de d	Commentaire
	DONNEES POMPE-VANNE	DB 1 DB 1		BLOC DE DONNEES
	DONNEES DE SUPERVISION	DB 2 DB 2		BLOC DE DONNEES DE SUPERVISION
	CAP_HAUT_G	E 272.1	BOOL	CAPTEUR NIVEAU HAUT
	CAP_MOYEN_G	E 272.2	BOOL	CAPTEUR NIVEAU MOYEN
	CAP_BAS_G	E 272.3	BOOL	CAPTEUR NIVEAU BAS
	CAP_HAUT_H	E 272.4	BOOL	CAPTEUR NIVEAU HAUT
	CAP_MOYEN_H	E 272.5	BOOL	CAPTEUR DE NIVEAU MOYEN
	CAP_BAS_H	E 272.6	BOOL	CAPTEUR NIVEAU BAS
	PB_ON	E 282.6	BOOL	SIGNAL DE MARCHE PB
	PG-ON	E 282.7	BOOL	SIGNAL DE MARCHE DE P6
	PGs_ON	E 283.0	BOOL	SIGNAL DE MARCHE DE PGS
	PH_ON	E 283.1	BOOL	SIGNAL DE MARCH PH
	PHs_ON	E 283.2	BOOL	SIGNAL MARCHE PHs
	AGIT1_ON	E 283.3	BOOL	SIGNAL_ON AGIT1
	AGIT2_ON	E 283.4	BOOL	SIGNAL_ON AGIT1
	AGIT3_ON	E 283.5	BOOL	SIGNAL_ON AGIT1
	AGIT4_ON	E 283.6	BOOL	SIGNAL_ON AGIT1
	AGIT5_ON	E 283.7	BOOL	SIGNAL_ON AGIT1
	AGIT6_ON	E 284.0	BOOL	SIGNAL_ON AGIT1
	VG_SIGNAL_ON	E 284.1	BOOL	SIGNAL_ON AGIT1
	VG_SIGNAL_OFF	E 284.2	BOOL	SIGNAL_ON AGIT1
	VB_SIGNAL_ON	E 284.3	BOOL	SIGNAL_ON AGIT1
	VB_SIGNAL_OFF	E 284.4	BOOL	SIGNAL_ON AGIT1
	VG2_SIGNAL_ON	E 284.5	BOOL	SIGNAL_ON AGIT1
	VG2_SIGNAL_OFF	E 284.6	BOOL	SIGNAL_ON AGIT1
	VG1_SIGNAL_ON	E 284.7	BOOL	SIGNAL_ON AGIT1
	VG1_SIGNAL_OFF	E 285.0	BOOL	SIGNAL_ON AGIT1
	VGs2_SIGNAL_ON	E 285.1	BOOL	SIGNAL_ON AGIT1
	VGs2_SIGNAL_OFF	E 285.2	BOOL	SIGNAL_ON AGIT1
	VGs1_SIGNAL_ON	E 285.3	BOOL	SIGNAL_ON AGIT1
	VGs1_SIGNAL_OFF	E 285.4	BOOL	SIGNAL_ON AGIT1
	VH_SIGNAL_ON	E 285.6	BOOL	SIGNAL_ON AGIT1
	VH_SIGNAL_OFF	E 285.7	BOOL	SIGNAL_ON AGIT1
	VH2_SIGNAL_ON	E 286.0	BOOL	SIGNAL_ON AGIT1
	VH2_SIGNAL_OFF	E 286.1	BOOL	SIGNAL_ON AGIT1

Etat	Mnémonique	Opéran	Type de d	Commentaire
64	VH1_SIGNAL_ON	E 286.2	BOOL	SIGNAL_ON AGIT1
65	VH1_SIGNAL_OFF	E 286.3	BOOL	SIGNAL_ON AGIT1
66	VHs2_SIGNAL_ON	E 286.4	BOOL	SIGNAL_ON AGIT1
67	VHs2_SIGNAL_OFF	E 286.5	BOOL	SIGNAL_ON AGIT1
68	VHs1_SIGNAL_ON	E 286.6	BOOL	SIGNAL_ON AGIT1
69	VHs1_SIGNAL_OFF	E 286.7	BOOL	SIGNAL_ON AGIT1
70	BAC_G	E 287.0	BOOL	
71	BAC_H	E 287.1	BOOL	
72	LT_H	EW 274	INT	signal du niveau bac H
73	TIC_G	EW 276	INT	entree du signal temperature G
74	TIC_H	EW 278	INT	entree du signal temperature H
75	LT_G	EW 280	INT	signal du niveau bac G
76	TEMPERATURE	FC 1	FC 1	BLOC DE FONCTON DE TEMPERATURE
77	TRANSMETTEUR DE NIVEAU	FC 2	FC 2	TRANSMETTEUR DE NIVEAUX G ET H
78	CAPTEURS DE NIVEAUX	FC 3	FC 3	CAPTEURS DE NIVEAUX
79	Condition de choix	FC 4	FC 4	BLOC DE FONCTION POUR LE CHOIX DES BACS ET POMPES
80	CONTROL VANNES	FC 5	FC 5	BLOC DE FONCTION STANDARD POUR LE CONTROL DES VANNES
81	APPEL VANNES	FC 6	FC 6	BLOC DE FONCTION APPEL VANNES
82	AGITATEUR	FC 8	FC 8	BLOC DE FONCTION POUR LES AGITATEURS
83	POMPE	FC 9	FC 9	BLOC DE FONCTION POUR LES POMPES
84	MOTEUR HUILES BRUTES	FC 10	FC 10	MOTEUR HUILES BRUTES
85	CONTROL LES POMPES	FC 77	FC 77	BLOC DE FONCTION STANDARD POUR LE CONTROLE DES POMPES ET AGITATEURS
86	SIMULATION	FC 99	FC 99	BLOC DE FONCTION POUR LA SUPERVISION
87	SCALE	FC 105	FC 105	Scaling Values
88	ETAT_SIGNAL_LTG	M 0.5	BOOL	ETAT DU SIGNAL LTG
89	ETAT_SIGNA_LTH	M 0.6	BOOL	ETAT DU SIGNAL LTH
90	ETAT_SIGNAL_TICG	M 0.7	BOOL	ETAT DU SIGNAL TICG
91	ETAT_SIGNAL_TICH	M 1.0	BOOL	ETAT DU SIGNAL TICH
92	CHOIX_BAC_G	M 4.2	BOOL	CHOIX DU BAC G
93	CHOIX_BAC_H	M 4.3	BOOL	CHOIX DU BAC H
94	CHOIX_PG	M 12.0	BOOL	CHOIX DE LA POMP PG
95	CHOIX_PGs	M 12.1	BOOL	CHOIX DE LA POMP PGs
96	CHOIX_PH	M 12.2	BOOL	CHOIX DE LA POMP PH
97	CHOIX_PHs	M 12.3	BOOL	CHOIX DE LA POMP PHs
98	NIVEAU_BAC_G	MD 270	REAL	NIVEAU DU BAC G_LT

Etat	Mnémonique	Opéran	Type de d	Commentaire
64	VH1_SIGNAL_ON	E 286.2	BOOL	SIGNAL_ON AGIT1
65	VH1_SIGNAL_OFF	E 286.3	BOOL	SIGNAL_ON AGIT1
66	VHs2_SIGNAL_ON	E 286.4	BOOL	SIGNAL_ON AGIT1
67	VHs2_SIGNAL_OFF	E 286.5	BOOL	SIGNAL_ON AGIT1
68	VHs1_SIGNAL_ON	E 286.6	BOOL	SIGNAL_ON AGIT1
69	VHs1_SIGNAL_OFF	E 286.7	BOOL	SIGNAL_ON AGIT1
70	BAC_G	E 287.0	BOOL	
71	BAC_H	E 287.1	BOOL	
72	LT_H	EW 274	INT	signal du niveau bac H
73	TIC_G	EW 276	INT	entree du signal temperature G
74	TIC_H	EW 278	INT	entree du signal temperature H
75	LT_G	EW 280	INT	signal du niveau bac G
76	TEMPERATURE	FC 1	FC 1	BLOC DE FONCTON DE TEMPERATURE
77	TRANSMETTEUR DE NIVEAU	FC 2	FC 2	TRANSMETTEUR DE NIVEAUX G ET H
78	CAPTEURS DE NIVEAUX	FC 3	FC 3	CAPTEURS DE NIVEAUX
79	Condition de choix	FC 4	FC 4	BLOC DE FONCTION POUR LE CHOIX DES BACS ET POMPES
80	CONTROL VANNES	FC 5	FC 5	BLOC DE FONCTION STANDARD POUR LE CONTROL DES VANNES
81	APPEL VANNES	FC 6	FC 6	BLOC DE FONCTION APPEL VANNES
82	AGITATEUR	FC 8	FC 8	BLOC DE FONCTION POUR LES AGITATEURS
83	POMPE	FC 9	FC 9	BLOC DE FONCTION POUR LES POMPES
84	MOTEUR HUILES BRUTES	FC 10	FC 10	MOTEUR HUILES BRUTES
85	CONTROL LES POMPES	FC 77	FC 77	BLOC DE FONCTION STANDARD POUR LE CONTROLE DES POMPES ET AGITATEURS
86	SIMULATION	FC 99	FC 99	BLOC DE FONCTION POUR LA SUPERVISION
87	SCALE	FC 105	FC 105	Scaling Values
88	ETAT_SIGNAL_LTG	M 0.5	BOOL	ETAT DU SIGNAL LTG
89	ETAT_SIGNA_LTH	M 0.6	BOOL	ETAT DU SIGNAL LTH
90	ETAT_SIGNAL_TICG	M 0.7	BOOL	ETAT DU SIGNAL TICG
91	ETAT_SIGNAL_TICH	M 1.0	BOOL	ETAT DU SIGNAL TICH
92	CHOIX_BAC_G	M 4.2	BOOL	CHOIX DU BAC G
93	CHOIX_BAC_H	M 4.3	BOOL	CHOIX DU BAC H
94	CHOIX_PG	M 12.0	BOOL	CHOIX DE LA POMP PG
95	CHOIX_PGs	M 12.1	BOOL	CHOIX DE LA POMP PGs
96	CHOIX_PH	M 12.2	BOOL	CHOIX DE LA POMP PH
97	CHOIX_PHs	M 12.3	BOOL	CHOIX DE LA POMP PHs
98	NIVEAU_BAC_G	MD 270	REAL	NIVEAU DU BAC G_LT

134	TEMPO_VG	T	70	TIMER
135	TEMPO_VB	T	72	TIMER
136	TEMPO_VG2	T	73	TIMER
137	TEMPO_VG1	T	74	TIMER
138	TEMPO_VGS2	T	75	TIMER
139	TEMPO_VGS1	T	76	TIMER
140	TEMPO_VH	T	77	TIMER
141	TEMPO_VH2	T	78	TIMER
142	TEMPO_VH1	T	79	TIMER
143	TEMPO_VHs2	T	80	TIMER
144	TEMPO_VHs1	T	81	TIMER