

Introduction :

Après une étude approfondie du cahier des charges, une analyse bien détaillée, en prenant en considération toutes les contraintes existantes et la modification apportée au cahier des charges, on doit écrire un programme s'exécutant dans un automate programmable qui assure le bon fonctionnement du processus et configurer un pupitre opérateur "OP" qui assure la supervision et la commande des processus. Pour répondre à ces exigences on a choisit le matériel Siemens.

III-1- Siemens Simatic :

La famille Simatic est une branche issue de la société mère Siemens qui est responsable sur la production et le développement de produits destinés pour l'automatisation industrielle. Elle offre plusieurs gammes d'automates programmables industrielles, parmi lesquelles on cite S7-200, S7-300 et S7-400.

III-1-1- Automate S7-200 :

La famille S7-200 est un micro-automate programmable utilisé dans des applications d'automatisation variées. La figure (III.1) présente un micro- automate S7-200. Son dessin compact, son faible prix et son important jeu d'opération en font une solution idéale pour la commande de petites applications, il a une possibilité d'extension jusqu'à 7 modules [8].



Fig-III.1 – Automate S7-200

III-1-2 Automate S7-300:

La famille S7-300 est un mini-automate programmable utilisé dans des applications d'automatisation variées. La figure (III.2) présente un mini- automate S7-300, son dessin modulaire, son prix est élevé par rapport à S7-200, son important jeu d'opération en font une solution idéale pour la commande de moyens applications, il a une possibilité d'extension jusqu'à 32 modules [13] .



Fig-III.2- Automate S7-300

III-1-3 Automate S7- 400:

Le S7-400 est un automate haute performance pour les applications de milieu et haute gamme, possibilité d'extension à plus de 300 modules. Ces modules se présentent sous forme de boîtiers que l'on adapte sur un châssis. Des châssis d'extension sont à disposition pour faire évoluer le système [14].

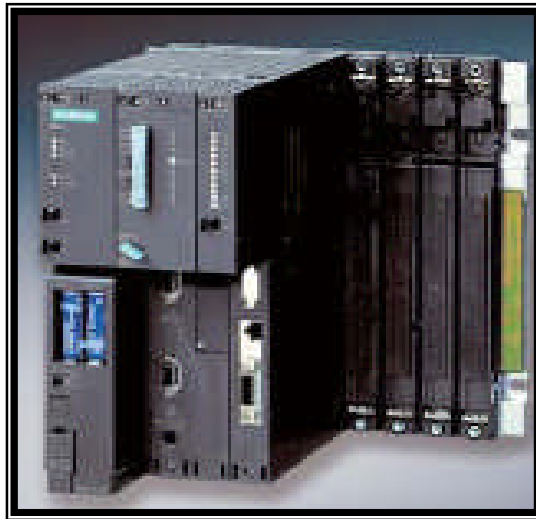


Fig-III.3- Automate S7-400.

III-2 Choix d'automate :

Après le recensement des Entrées / Sorties de processus (voir l'annexe), on a trouvé :

Les entrées :

- Logiques : $95 + 20\%$ (réserves) = 114.
- Analogiques : $17 + 20\%$ (réserves) = 21.

Les sorties :

- Logiques : $80 + 20\%$ (réserves) = 96.
- Analogiques : $5 + 20\%$ (réserves) = 6.

On a plus de 240 E / S. à partir de nombre et les caractéristiques des S7-200, S7-300, et S7-400. on a choisit l'automate S7-300.

III-3- Architecture de l'automate S7-300:

L'automate S7-300 comprend les composants suivants [14]:

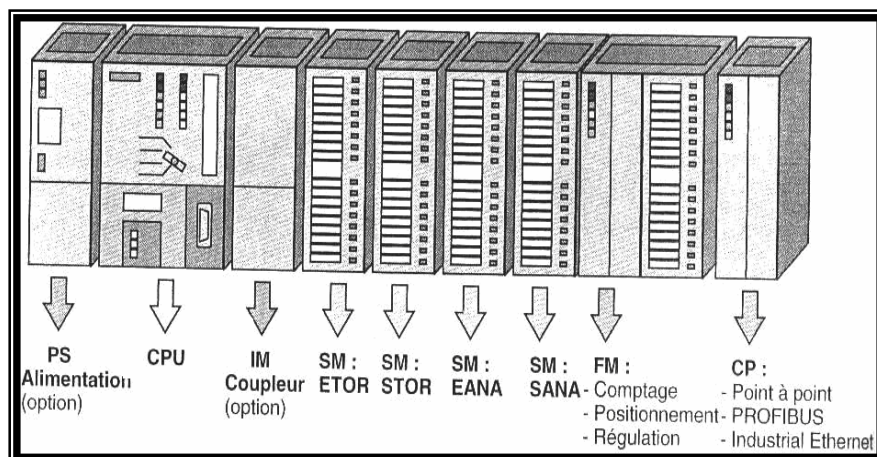


Fig-III.4- Composants de l'automate S7-300.

a- Rail profile :

L'alimentation électrique, la CPU, la carte de couplage IM et 8 modules des signaux maximum sont montés sur le rail profile [14].

b- Alimentation : PS

Le module d'alimentation délivre sous une tension 24V un courant de sortie assigne de 2A, 5A et 10A. La tension de sortie, à séparation galvanique, est protégée contre les courts circuits et la marche à vide. Une LED indique le bon fonctionnement de module d'alimentation. Un sélecteur permet de sélectionner la valeur de la tension primaire (120/230V) [14].

c- Unité centrale : CPU

La CPU regroupe les éléments suivants en face avant : [14]

- Signalisation d'état et de défaut.
- Commutateur à clé amovible à 4 positions.
- Raccordement pour tension 24V.
- Interface multipoint MPI pour console de programmation ou couplage à un autre système d'automatisation.
- Compartiment pour pile de sauvegarde (non disponible dans la CPU 312 IFM).
- Logement pour carte mémoire (non disponible dans la CPU 312 IFM, 314IFM).

d- Carte couplage : IM

Les coupleurs permettent de disposer d'une configuration à plusieurs châssis [14].

e- Module de signaux : SM

Ces modules sont sélectionnés en fonction de la plage de la tension d'entrée ou de la tension de sortie

- Module ETOR : 24V=, 120/230V ~ [14].
- Module STOR : 24V=, relais.
- Module EANA : tension, courant, résistance, thermocouple...
- Module SANA : tension, courant.

f- Modules de fonction : FM

Les modules de fonction offrent " des fonctions spéciales " : comptage, positionnement et régulation [14].

g- Modules de communication : CP

Processeur de communication pour le système de bus PROFIBUS [14].

III-4- Présentation de la console de programmation:

La console de programmation est un ordinateur complet spécialement conçu pour s'intégrer aux diverses techniques d'automatisation. Ses caractéristiques, sa conception ergonomique et ses équipements en font une console particulièrement bien adaptée à la maintenance ainsi qu'à la conception/ programmation, au test et à la mise en service des automates programmables SIMATIC [12].

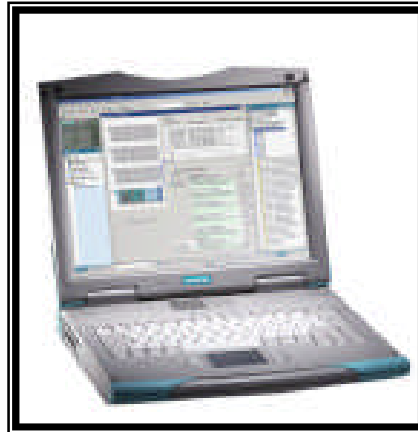


Fig-III.5- Console de programmation PG 720

III-5- Présentation du Logiciel step7 :

Step7 est le logiciel permettant de configurer et de programmer les automates programmables SIMATIC S7-300/400 et les systèmes d'automatisation SIMATIC M7-300/400, ainsi que les systèmes intégrés compact C7.

Step7 est constitué d'un logiciel de base et de logiciels optionnels s'exécutant sous système d'exploitation [9].

Le logiciel de base step7 nous assiste dans toutes les phases du processus de la création de nos solutions d'automatisation, comme par exemple :

- La création et la gestion de projets.
- La configuration et le paramétrage du matériel et de la communication.
- La gestion de mnémoniques.
- La création de programmes pour systèmes cibles S7.
- Le chargement de programmes dans des systèmes cibles.
- Le test de l'installation d'automatisation.
- Le diagnostic lors de perturbations de l'installation.

La conception de l'interface utilisateur du logiciel step7 répond aux connaissances ergonomiques modernes et son apprentissage est très facile.

Le langage de programmation step7 dispose de plusieurs modes de représentation, selon les goûts et l'état des connaissances. En respectant certaines règles, le programme peut être conçu sous forme de liste d'instructions puis converti en un autre mode de représentation [9].

III-5-1- Langage contact (CONT) :

Le schéma à contact s'apparente aux schémas de circuits électriques, il fait appel à des symboles, tels que les contacts et les bobines.

Ce mode de représentation convient particulièrement à tous les familiers de la technologie des contacteurs[9].

III-5-2- Langage liste (LIST):

La liste d'instructions se compose d'opération step7. LIST permet de programmer librement (et parfois de manière complexe).

Ce mode de représentation est particulièrement destiné aux programmeurs qui maîtrisent déjà d'autres langages de programmation [9].

III-5-3- Langage logigramme (LOG):

Le logigramme est un langage de programmation graphique qui utilise des boites logiques.

Le symbole placé dans la boîte définit la fonction à manipuler par exemple & (opération logique ET). Ce mode de représentation permet à des non programmeurs, par exemple des spécialistes des procédés industriels, d'accéder à la programmation [9].

Le logigramme est disponible à partir de la version 3.0 de step7.

Exemple :

La lampe doit s'allumer lorsque S1 est activé et S2 inactivé

Le matériel :

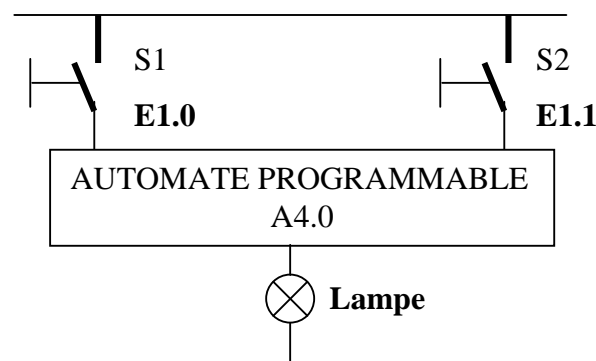


Fig-III.6- Schéma du câblage

Le Programme :

Langage CONT :

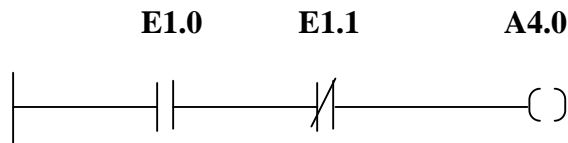


Fig-III.7- Schéma d'une programmation avec langage CONT

Langage LOG :

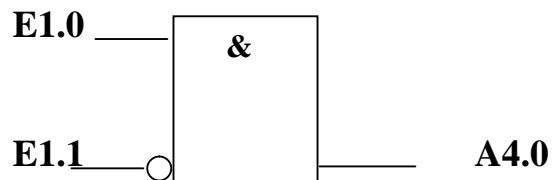


Fig-III.8- Schéma d'une programmation avec langage LOG

Langage LIST :

```

U    E1.0
UN   E1.1
=    A4.0
  
```

III-6- Choix du matériel :

A partir de :

- L'alimentation au niveau de l'armoire.
- Les caractéristiques des équipements (capteurs, fin de courses ...).
- Les circuits de puissance des équipements (vannes, pompes...).
- Le recensement des E/S.

On a choisie les composants suivants :

- Quatre (04) modules de : DI 32× DC 24V.
- Sept (07) modules de : DO 16×Rel.AC 120/230V.
- Trois (03) modules de : AI 8×16 Bits.

- Un (01) module de : AO 8× 12 Bits.
- Trois (03) Rails profilés.
- Une (01) unité de traitement : CPU 314.
- Une (01) alimentation : PS 307 5A.
- Deux (02) coupleurs : IM360S, IM361R.

III-7- Configuration du matériel :

Au sens logiciel du terme, on entend par configuration l'organisation des modules dans une table de configuration.

A l'image de la réalité matérielle de notre S7-300, il s'agit de ranger les modules sur un châssis de façon virtuelle avec le logiciel step7. Nous pouvons choisir les modules dans un catalogue électronique et le reporter dans la table de configuration à l'endroit correspondant à leur emplacement sur le profilé-support. L'emplacement dans la table de configuration doit correspondre à l'emplacement réel sur profilé-support.

Dans la table de configuration, step7 affecte automatiquement une adresse à chaque module. Par paramétrage, on entend le fait de choisir et de fixer les propriétés et le comportement des modules paramétrables [13].

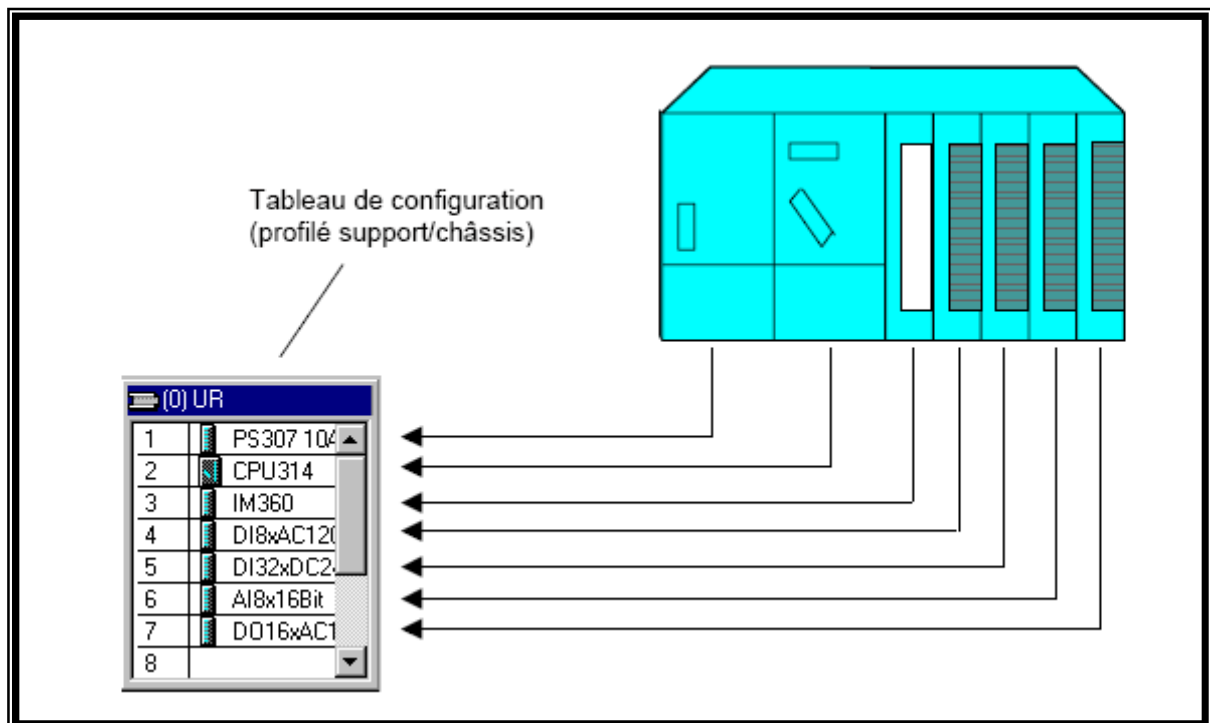


Fig-III.9- Configuration du matériel

III-8- Mnémoniques :**III-8-1-Généralités :**

Un mnémonique (nom symbolique) nous permet d'utiliser des désignations parlantes à la place d'adresse. On distingue les mnémoniques locaux et les mnémoniques globaux [9].

III-8-2- Domaine de validité :

Un mnémonique global est connu dans le programme utilisateur entier : tous les blocs de programme peuvent y faire appel. Son nom doit être univoque dans l'ensemble du programme utilisateur.

Un mnémonique local est connu uniquement dans le bloc où il est défini. Nous pouvons utiliser le même nom dans différents blocs à des fins différentes [9].

III-8-3- Utilisation de mnémoniques globaux :

Nous pouvons définir des mnémoniques globaux pour des entrées, sorties, compteurs, mémentos et blocs. Les adresses suivantes sont autorisées [9]:

- | | |
|----------------------------------|----------------------|
| - Signaux d'E/S (mémoire image) | E, A |
| - Entrées, sorties de périphérie | PE, PA |
| - Mémentos | M |
| - Temporisation, compteurs | T, Z |
| - Blocs de code | FB, FC, SFB, SFC, OB |
| - Bloc de données | DB |
| - Types de données utilisateurs | UDT |
| - Table de variable | VAT |

Nous définissons les mnémoniques globaux dans la table des mnémoniques.

III-8-4- Utilisation de mnémoniques locaux :

Nous pouvons utiliser des mnémoniques locaux pour les paramètres de blocs (paramètres d'entrée, de sortie et d'entrée / sortie) et pour les données temporaires ou statiques d'un bloc.

Les mnémoniques locaux sont définis lors de la saisie du programme dans les déclarations des variables du bloc [9].

III-9- Adressage absolu et adressage symbolique :

Dans un programme step7, nous utilisons des opérandes comme des signaux d'E/S, des mémentos, des compteurs, des temporisations, des blocs de données et des blocs fonctionnels. Nous pouvons accéder à ces opérandes par adressage absolu dans notre programme. Toutefois, la lisibilité de nos programmes sera grandement améliorée si nous faisons plutôt appel à des mnémoniques [9].

III-9-1- Adresse absolue :

Une adresse absolue est composée d'un identificateur d'opérande et d'une adresse (par exemple A4.0. E 1.1, M 2.0, ..) [9].

III-9-2- Adressage symbolique :

Nous pouvons structurer notre programme de manière plus lisible et faciliter ainsi la correction d'erreurs en affectant des noms symbolique (mnémoniques) aux adresses absolues.

Step7 est en mesure de convertir automatiquement les mnémoniques dans les adresses absolues requise. Si nous préférons adresser des ARRAY, STRUCT, bloc de données locales, blocs de code et types de données utilisateur de manière symbolique, nous devons cependant d'abord affecter un mnémonique aux adresses absolues, avant de pouvoir réaliser l'adressage symbolique.

Nous pouvons par exemple affecter le mnémonique Moteur –Marche à l'opérande A 4.0, puis en utiliser Moteur –Marche comme adresse dans une instruction de programme.

L'adressage symbolique nous permet de déterminer plus aisément dans quelle mesure des éléments de programme correspondent aux composants de notre projet de commande du processus [9].

II-10- Adressage des modules :

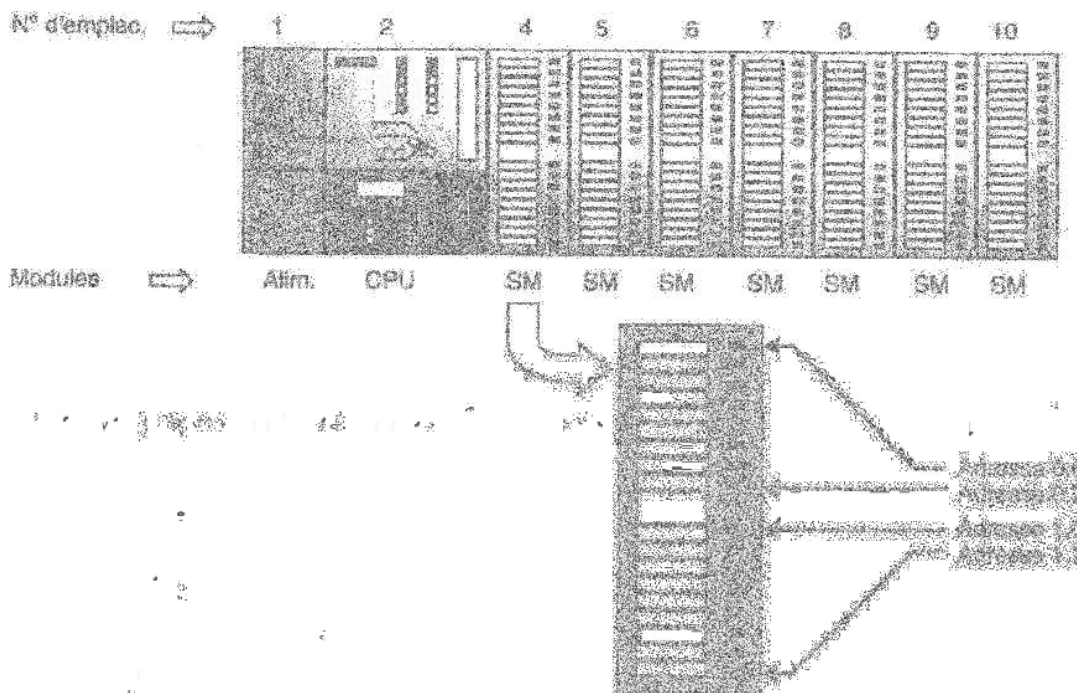


Fig-III.10- Adressage des modules S7-300

Numéros d'emplacement :

Les numéros d'emplacement dans le châssis d'un S7-300 facilitent l'établissement du schéma d'adressage dans l'environnement S7-300. La première adresse dans le module est déterminée par sa position dans le châssis [14].

Emplacement 1 :

Alimentation électrique. En standard, elle occupe le premier emplacement. Un module d'alimentation n'est pas nécessairement obligatoire. Un système S7-300 peut aussi fonctionner en 24V directement [14].

Emplacement 2 :

C'est l'emplacement de la CPU [14].

Emplacement 3 :

Il est logiquement réservé à un coupleur (IM) pour une configuration multi châssis avec châssis d'extension.

Même lorsqu'aucun coupleur IM n'est monté, il faut en tenir compte dans le schéma d'adressage [14].

Emplacement 4 à 11 :

L'emplacement 4 est le premier emplacement disponible pour des modules d'E/S, des processeurs de communication (CP) ou des modules de fonction (FM) [14].

Exemple d'adressage :

- Un module d'entrée TOR à l'emplacement 4 commence avec l'adresse d'octet 0.
- La LED supérieure sur un module de sortie TOR à l'emplacement 6 correspond à A8.0.

Notation :

4 adresses d'octet sont réservées à chaque emplacement. En cas d'utilisation de module d'entrée/sortie TOR 16 voies, on perd 2 adresses d'octet par emplacement [14].

III-11- Programmation :

Nous pouvons programmer notre automate très simplement, en créant un programme utilisateur que nous chargeons ensuite dans la CPU de notre S7-300, ce programme utilisateur à créer comprend différents blocs avec lesquelles nous allons pouvoir structurer notre programme [9].

Dans notre programme, On a utilisé les blocs (OB1) et (FC) [9].

- Un bloc d'organisation (OB1) pour gérer le traitement cyclique du programme.
- Une fonction (FC) dans laquelle nous entrons le programme proprement dit.

Bloc d'organisation(OB) :

Un bloc d'organisation (OB) constitue l'interface entre le système d'exploitation de la CPU et le programme utilisateur. L'OB spécifie l'ordre selon lequel sont traités les blocs de programme utilisateur [9].

La fonction(FC) :

Une fonction (FC) est un bloc de code qui est sans « mémoire », mais que peut transmettre des paramètres. L'utilisation de ce bloc est particulièrement intéressante pour programmer des fonctions qui reviennent fréquemment [9].

III-12- Description du pupitre opérateur “ OP “ :

Les pupitres de contrôle-commande “OP” nous permet de représenter non seulement sous forme graphique les états de service, les valeurs actuelles du processus et les dérangements d’un automate y étant accouplé, mais également de commander aisément la machine ou l’installation à surveiller. Pour se faire, les pupitres de contrôle-commande nous proposent une série de fonction standard. L’adaptation de l’affichage et de la manipulation des pupitres de contrôle-commande aux diverses exigences du processus peut être réalisée de manière optimale à l’aide de logiciel de configuration Protool [10].

Les pupitres de contrôle-commande nous permettent de [10]:

- Piloter et surveiller le processus grâce à un guidage par menu. Nous pouvons ainsi entrer des valeurs de consigne ou commander des actionneurs, par exemple en saisissant des valeurs ou en appuyant sur les touches de fonction configurées.
- Visualiser des processus, machines et installations, grâce à des images graphique ou semi-graphique ;
- Visualiser des messages d’événement ou d’alarme ainsi que visualiser des variable processus, par exemple sous forme de champ d’affichages, d’histogrammes, de courbes d’affichages d’état.
- Intervenir directement dans le déroulement du processus au moyen du clavier intégré.



Fig-III.11- Pupitre Opérateur “ OP”.

III-13- Présentation du logiciel Protool :

Protool est un outil convivial destiné à la configuration du pupitre opérateur (OP). Il fonctionne sous Microsoft Windows. La plupart des opérations à effectuer dans Protool peuvent l’être aussi bien avec la souris qu’avec le clavier [10].

III-14- Configuration du pupitre opérateur :

La configuration de pupitre opérateur consiste à créer des images et des messages et à établir les liaisons correspondantes avec le programme de l'automate. Il est ainsi possible de visualiser les opérations qui se déroulent dans l'automate et d'intervenir sur celles-ci [10].

III-14-1- Images :

Les images servent à afficher une représentation du processus. De cette manière, l'opérateur peut reconnaître rapidement le contexte du processus et y intervenir si nécessaire. Des textes expliquent les divers éléments des images. Des graphiques par exemple des courbes ou des histogrammes, représentent les variations, par exemple d'une température ou d'un niveau de remplissage.

Les images doivent être appelées par l'opérateur. Elles affichent des valeurs en provenance de l'automate. L'opérateur peut également entrer des valeurs qui sont ensuite transmises à l'automate [10].

III-14-2- Messages :

Les messages indiquent certains états de fonctionnement à l'opérateur ou attirent son attention sur des alarmes dans le déroulement du processus. Les messages sont affichés automatiquement [10].

III-14-3- Touches :

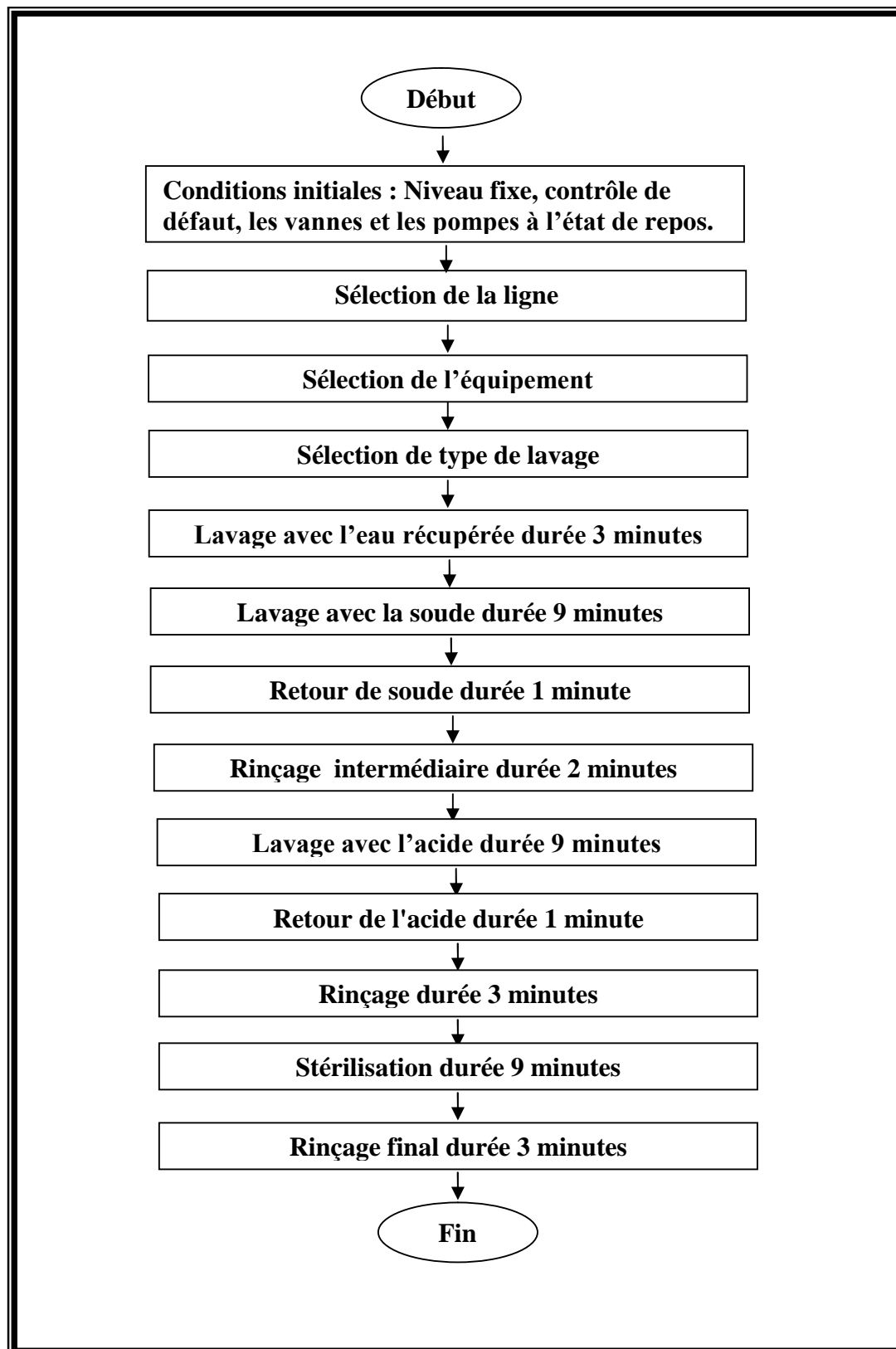
Les pupitres opérateurs possèdent un clavier de touche système et un clavier de touches de fonction. Le clavier de touches système comportant la commande du pupitre opérateur, par exemple le déplacement du curseur ou la saisie. Des fonctions peuvent être affectées aux touches de fonction au cours de la configuration, c'est ce qui permet de réaliser l'enchaînement des commandes proprement dit [10].

III-14-4- Données de configuration :

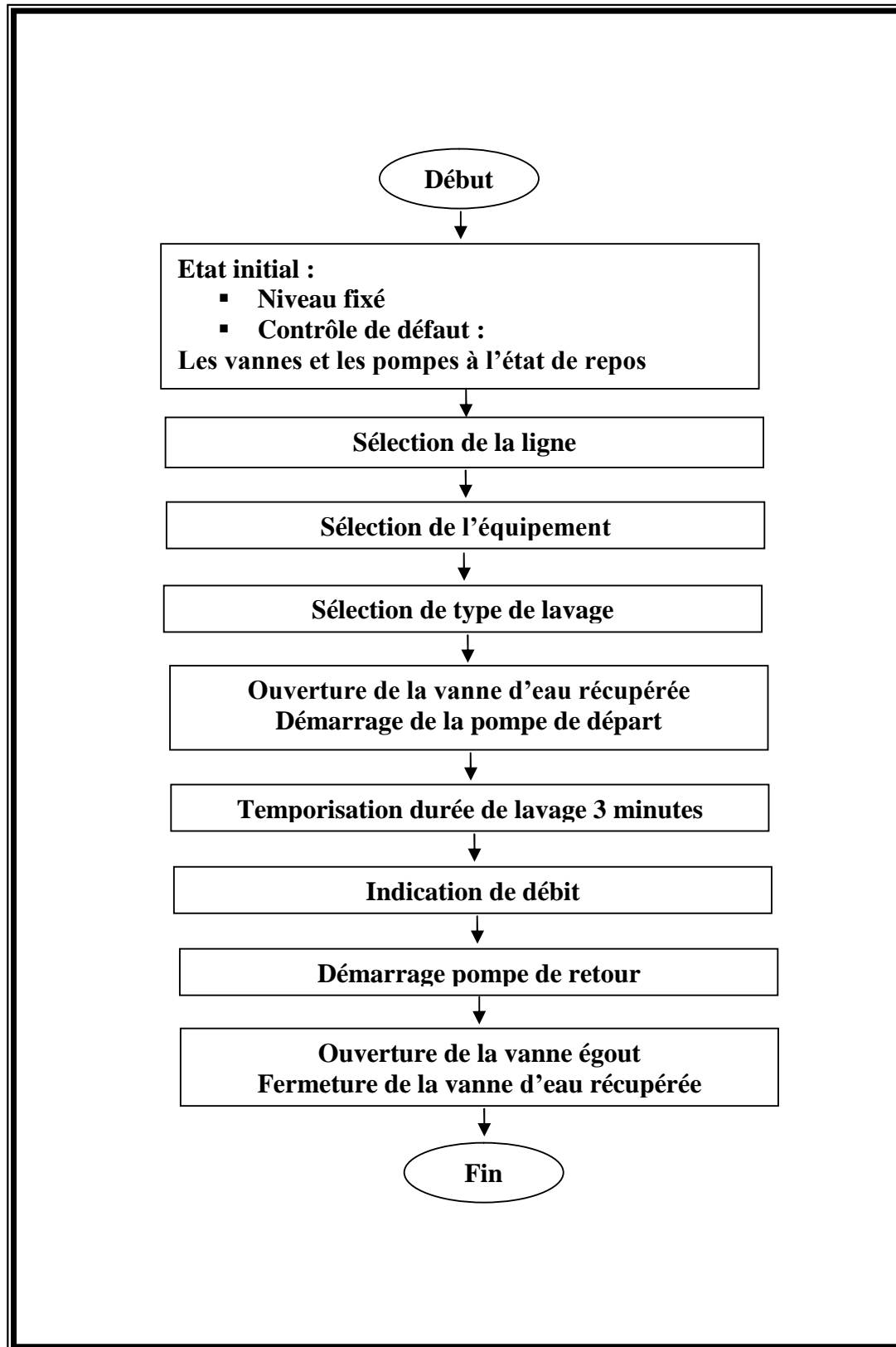
La configuration est réalisée sur un PC ou sur une console de programmation (PG). La configuration doit ensuite être générée dans Protool, puis transférée vers le pupitre opérateur. Lors qu'un couplage à l'automate est établi, l'OP affiche les valeurs actuelles [10].

III-15- Organigrammes des cycles de fonctionnement :

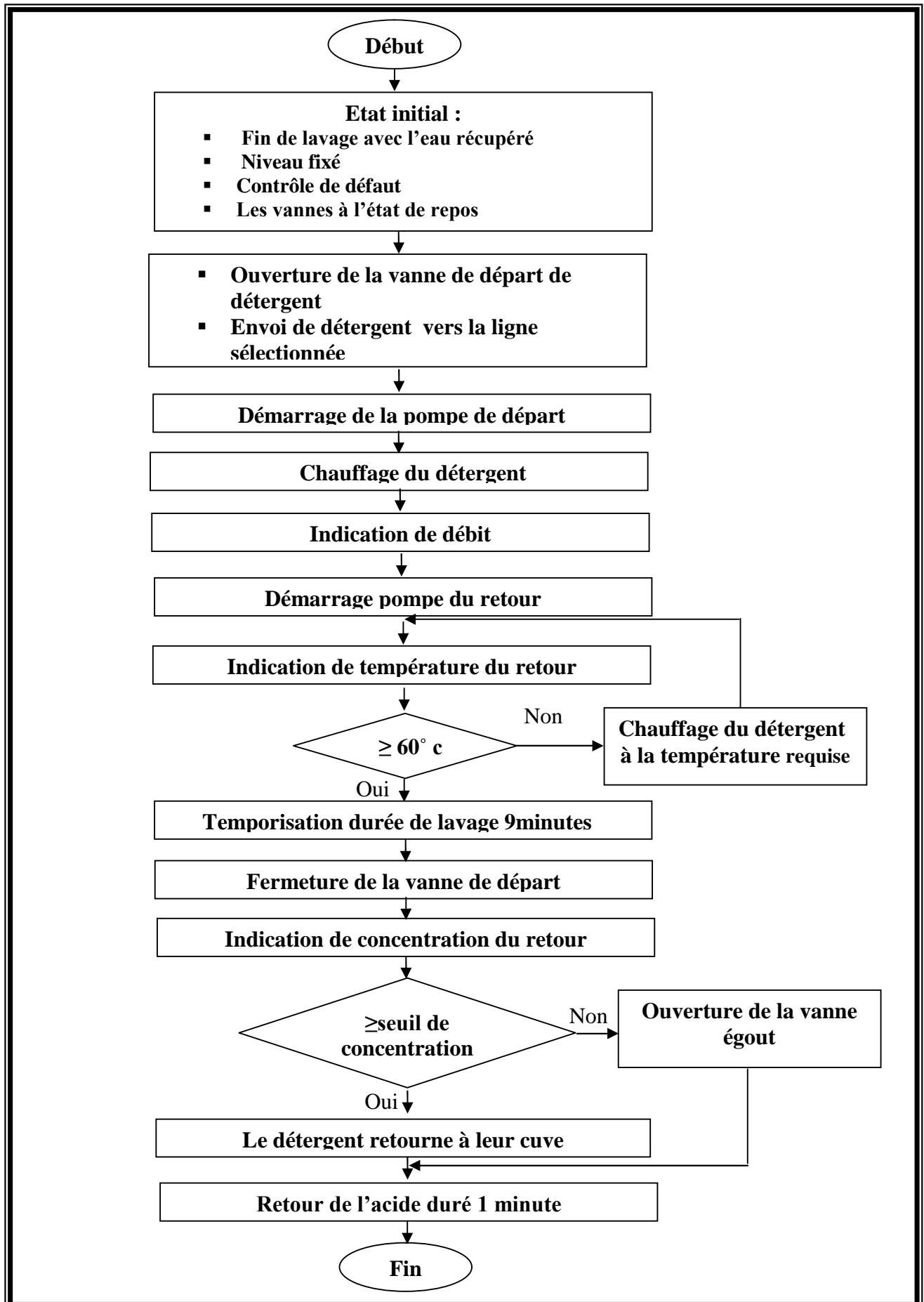
Afin d'illustrer le déroulement des séquences de fonctionnement du processus, on a représenté les organigrammes suivants :

Organigramme (1) : Nettoyage global de la station

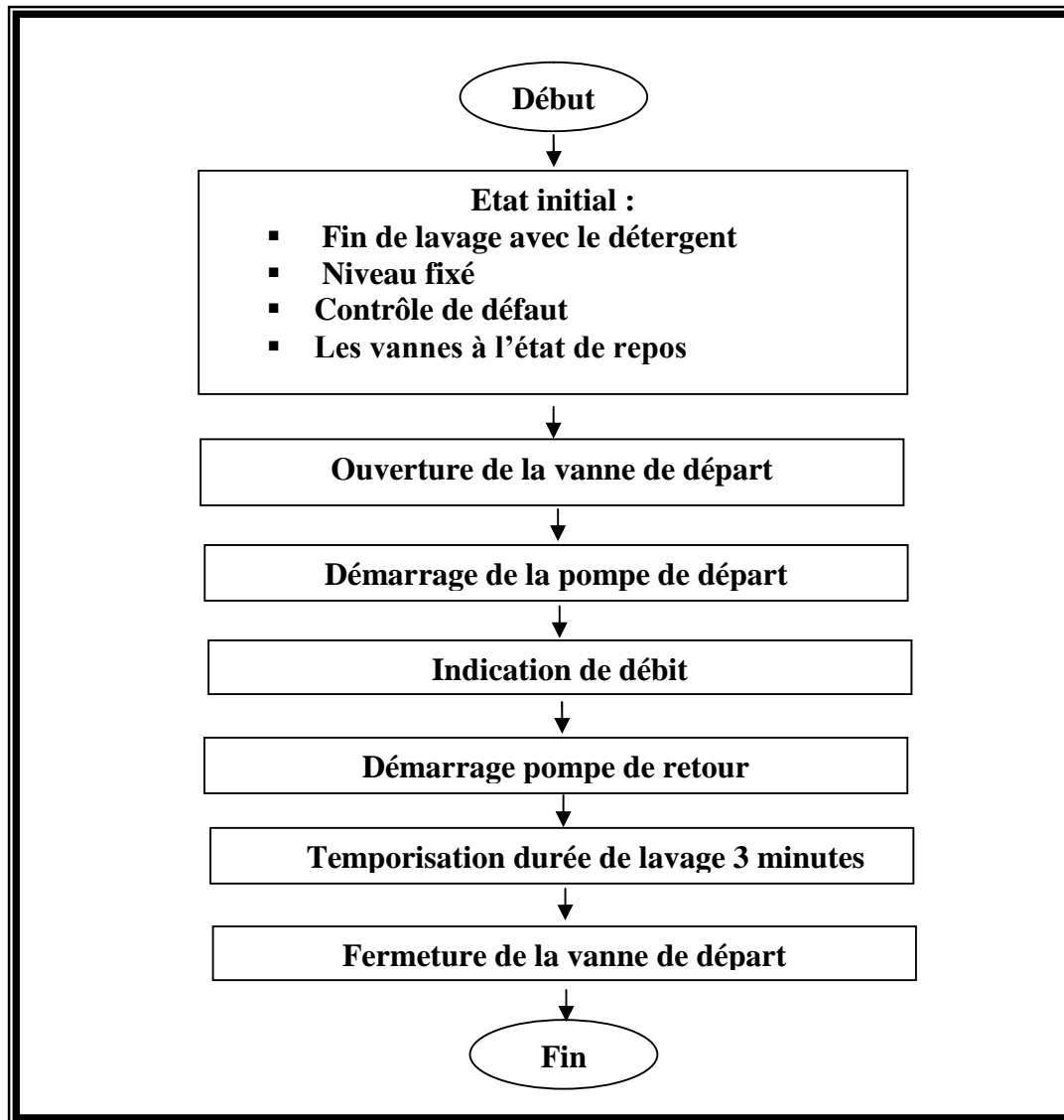
Organigramme (2) : Lavage avec l'eau récupérée



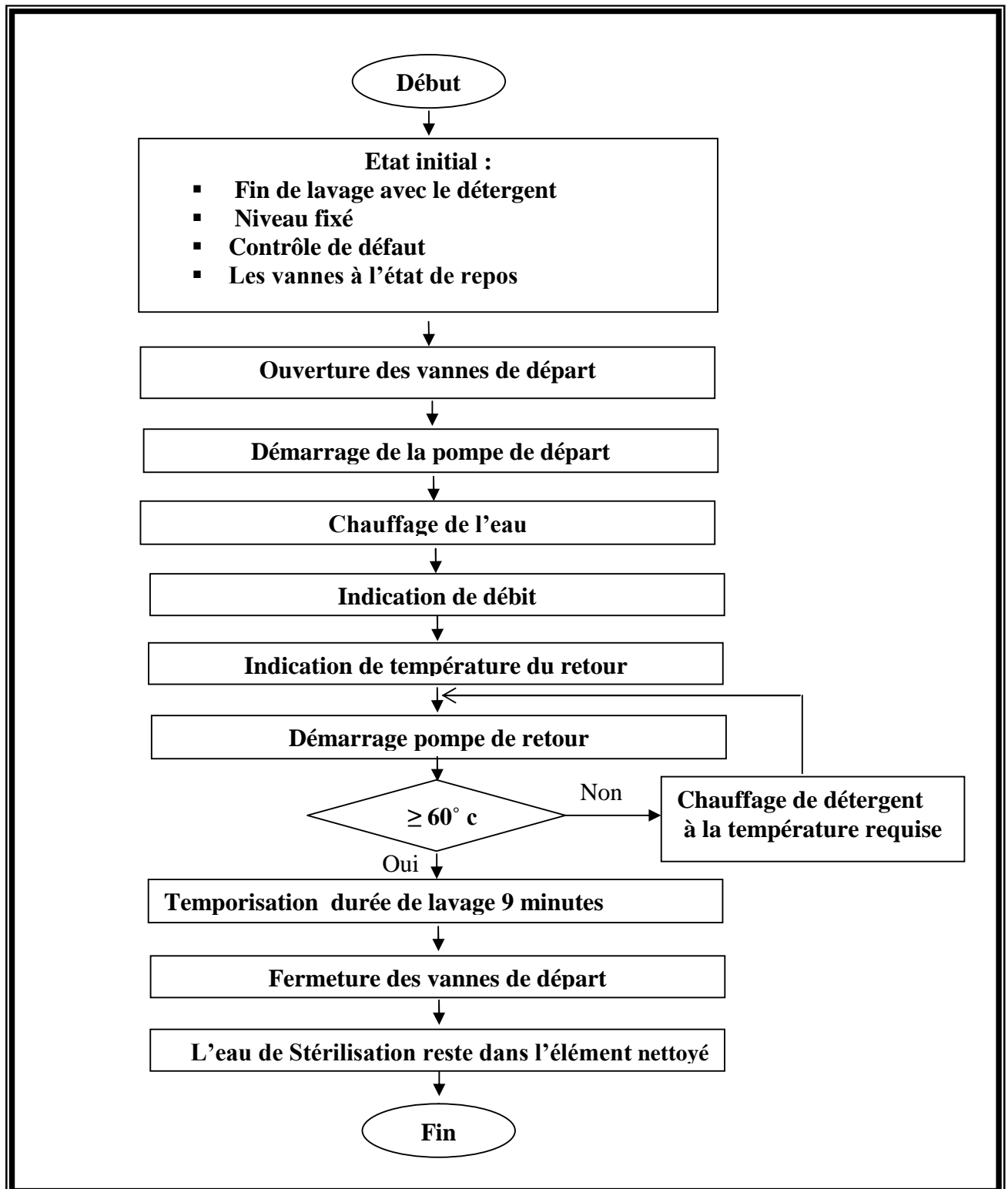
Organigramme (3) : Séquences de lavage avec les détergents concentrés



Organigramme (4) : Séquences du lavage avec l'eau neuve



Organigramme (5) : Séquences de la stérilisation – cuve d'eau neuve, tube de stérilisation –



Simulation :

Ce projet d'automatisation de la station de nettoyage de la laiterie Sidi Saâda- Relizane a été pris en charge par l'entreprise Sider Automation avec notre modeste collaboration pour la programmation de la mise en service de l'installation en mode automatique. La figure (III.12) représente l'affichage de l'OP du schéma synoptique de l'installation de nettoyage.

La simulation sera exposée par vidéo projecteur lors de la soutenance

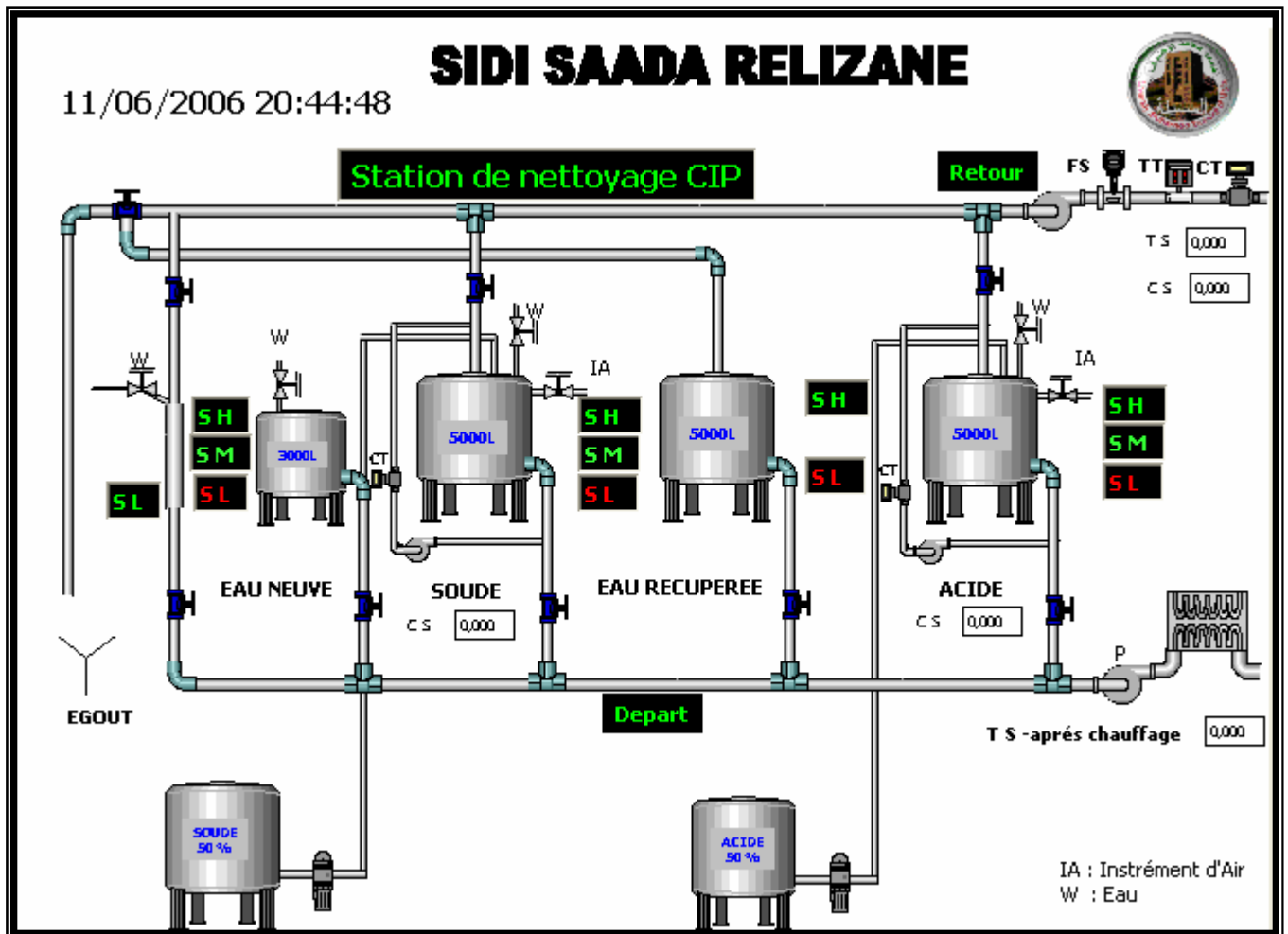


Fig-III.12- schéma synoptique de l'installation affiché sur l'écran de l'OP

Conclusion :

L'objectif de notre projet est l'automatisation de la station de nettoyage en place des 5 lignes, cette automatisation se fait en deux tâches essentielles, d'une part utilisation d'un automate dans le rôle de contrôle et de commande où il traite les informations que lui fournissent des détecteurs et des actionneurs (électrovanne, pompe, ...etc.), d'autre part la supervision, la conduite et le contrôle de l'instrumentation par un Pupitre Opérateur.

Conclusion générale

Dans le cadre de notre travail, nous avons présenté une solution programmée pour automatiser la station de nettoyage de la laiterie Sidi Saâda / Relizane. En premier lieu, nous avons abordé une étude théorique sur les systèmes automatisés et on a montré les performances d'une solution programmée par rapport à une celle câblée. Ensuite on a présenté le cahier des charges de la laiterie, ce dernier répond au besoin du client. Pour automatiser le processus de nettoyage, on a proposé avec la collaboration des ingénieurs de la société Sider Automation une solution programmée basée sur l'utilisation d'un automate S7-300, sa programmation, sa commande et sa supervision par un OP 270 10''.

Ce projet de fin d'études nous a été d'un grand apport d'une part, la connaissance et la maîtrise de nouveaux logiciels de contrôle et de commande qui sont très répandus à l'échelle mondiale et d'autre part, la participation aux différentes étapes de réalisation d'un projet en cours d'automatisation portant sur un thème d'actualité et d'avenir à savoir l'agroalimentaire.

Notre stage au sein de Sider Automation nous a permis de côtoyer le monde professionnel avec toutes ces spécificités et nous a ouvert une brèche sur l'évolution de la technologie en Algérie.

Enfin citons les avantages concrétisés avec cette réalisation :

- Amélioration de la qualité du produit.
- Optimisation de l'utilisation des produits coûteux.
- Protection de l'environnement contre les agressions chimiques (Soude, Acide).
- Amélioration des facteurs clés du processus à savoir : temps, produit, équipement.
- Pouvoir concurrencer le marché national et international.