

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Projet de Fin d'Études

présenté par

SADEK Mohamed Abdou

&

MISSERAOUI Ishak

pour l'obtention du diplôme Master 2 en Électronique option Signaux en Ingénierie des
Système et Informatique Industrielle (SISII)

Thème>

Contribution à la modernisation d'une station de Nettoyage En Place (NEP) de la société VitaJus

Proposé par : BENNILA Norddine

Année Universitaire 2012-2013

Remerciements

Tout d'abord on remercie dieu tout puissant pour la bonne santé, la volonté et de la patience qu'il nous a données tout au long de notre étude.

Nous remercions très sincèrement Mr. BENNILA Norddine notre promoteur, pour ses conseils pertinents, et ses orientations judicieuses, sa patience et diligence, et pour ses suggestions qui ont grandement facilité notre travail.

Nous remercions également Mr. SAHLAOUI Fodhil notre Co-promoteur, pour leurs conseils, et leur disponibilité lors des différentes sollicitations.

Nous remercions aussi Mr. BENCHERCHALI, Mr. BENSLAMA, Me. CHENTIR et Mme .ELMIRAOU, pour leurs conseils, et leur orientation.

Nous tenons à exprimer notre gratitude aux membres de jury qui ont bien voulu examiner ce travail.

Notre profonde gratitude à tous les enseignants du département d'électronique, qui nous ont encadrés et donner le meilleur d'eux même en nous assurant une formation aussi meilleure que possible.

Nous remercions vivement toutes les personnes qui ont contribué de près ou de loin, à la réalisation de ce mémoire.

Dédicaces

Je dédie ce modeste travail à ceux qui sont les plus chers êtres au monde, Mon père, et ma mère, à qui je n'arriverais jamais à exprimer ma gratitude et ma reconnaissance, pour leurs amours et leurs soutiens tout au long de mes études.

A mes frères Aniss, Kacem, Saad, Krimou et Abd-Erahmenne

A mes sœurs Azza et Soundous

A mes amis Brahim, Lotfi, Amine, djawad et bilel

A toute ma famille et amis

MOHAMED ABDOU SADEK

Je dédie ce modeste travail à la mémoire de mon cher père et à l'être la plus chère sur cette terre, pour ma mère qui je n'arriverais jamais à exprimer ma gratitude et ma reconnaissance, pour leurs amours et leurs soutiens tout au long de mes études.

A ma grande mère et mon oncle mourade

A mes frères Moncef, Sidali, Mohamed et chakib

A mes sœurs

A tous mes neveux et nièces

A mes amis IBRAHIM, Abdou, RANJA, Amine, SAMIR, NASSIM et RABAHE

A toute ma famille et amis

ISHAK MISSERAOU

ملخص:

قمنا في عملنا هذا بتحديث محطة التنظيف ناب التابعة لشركة فيتاجو وذلك بتزويدها بنظام المراقبة والتحكم بواسطة من نوع سيمنس من نوع سيمنس أيضا .

يجب أن نستعمل و نظام التواصل HMI على المستوى مشغل جميع الأوامر، حالة الجهاز، والتنبيه

كلمات المفاتيح: محطة التنظيف ناب ، المبرمج الألي أبيي ، نظام التواصل بين الانسان والآلة.

Résumé :

Dans notre travail nous avons modernisé la station de nettoyage en place NEP, au sein de la société VitaJus, en la dotant d'un système de commande et de contrôle entièrement automatisé, par l'appui d'un automate programmable API et d'une interface homme machine HMI de marque SIEMENS. Au niveau opérateur toute les commandes, état de la machine, et alarme doivent passer par l'interface HMI.

Mots clés : NEP; API; l'interface HMI.

Abstract :

In our work we have modernized the station cleaning up NEP to society VitaJus, endowing it with a control system and fully automated control by supporting a PLC programmable controller and a human Machine Interface HMI SIEMENS brand. The operator level all orders, machine status, alarm and must go through the HMI.

Keywords: NEP; PLC; the HMI.

Listes des acronymes et abréviations

NEP : Nettoyage En Place

IHM : Interface Homme Machine

API : Automate Programmable Industrielle

GRAFCET : Graphe Fonctionnel de Commande Etape Transition

LIST : Le langage à instruction (machine)

CONT : Le langage à contact

LOG : Le langage graphique

MPI : L'interface multipoint

PROFIBUS : Bus de terrain propriétaire

PT100 : Sonde de température à résistance platine

P : Action Proportionnelle

I : Action intégral

D : Action Dérivé

Ta : Constante du temps

Tu : Temps de retard

BDL : Bac de Lancement

Liste des figures

Figure 1.1: Diagramme de production de l'usine.....	7
Figure 2.1 : Schéma synoptique de la station.....	11
Figure 2.2 : L'API S7-200 (CPU224).....	12
Figure 2.3 : Pompe de pression électrique.....	13
Figure 2.4 : Pompe doseuse pneumatique.....	14
Figure 2.5 : Echangeur à plaques.....	14
Figure 2.6 : Conductivimètre.....	15
Figure 2.7 : Sonde de température PT100.....	16
Figure 2.8 : Capteur de niveau.....	16
Figure 2.9 : Vanne de régulation (modulante).....	18
Figure 2.10 : Purgeur de vapeur.....	19
Figure 2.11 : Clapet anti-retour.....	19
Figure 2.12 : Régulateur de température PID (OMRON).....	20
Figure 3.1 : Structure générale d'un API.....	24
Figure 3.2 : Fenêtre du SIMATIC manager.....	26
Figure 3.3 : Organigramme pour la création de projets sous STEP 7.....	28
Figure 3.4 : Création d'un projet sans l'assistant.....	29
Figure 3.5 : Configuration du matériel.....	29
Figure 3.6 : La fenêtre d'édition de mnémoniques.....	30
Figure 3.7 : L'hierarchie d'un projet STEP 7.....	31
Figure 3.8 : Création de blocs de données.....	34

Figure 3.9:	Exemple d'un « OU » logique avec le langage ladder.....	35
Figure 3.10:	Exemple d'un « OU » logique avec le langage FBD.....	36
Figure 3.11:	Exemple d'un « OU » logique avec le langage.....	37
Figure 3.12:	Système avec pupitre opérateur.....	40
Figure 3.13:	Système avec plusieurs pupitres opérateur.....	41
Figure 3.14:	Fenêtre configuration variables.....	43
Figure 3.15:	Création d'une vue.....	44
Figure 4.1:	la CPU 314C-2DP.....	47
Figure 4.2:	Câble de communication MPI.....	49
Figure 4.3:	Pupitre digital MP 277 10 " Touch 1.1.4.0.....	50
Figure 4.4:	Capteur de proximité inductif.....	50
Figure 4.5:	Détecteur de niveau VEGACAP 69.....	51
Figure 4.6:	L'équation de l'étape M8.1.....	52
Figure 4.7:	L'équation de l'étape M9.0.....	53
Figure 4.8:	Conversion de l'étape M8.1 en langage Ladder (LD).....	54
Figure 4.9:	Conversion de l'étape M9.0 en langage Ladder (LD).....	54
Figure 4.10:	Le bit mémoire M1.0.....	55
Figure 4.11:	Le bit mémoire M2.0.....	56
Figure 4.12:	Le bit mémoire M3.0.....	56
Figure 4.13:	Le bit mémoire M6.0.....	57
Figure 4.14:	Le bloc fonctionnel FB1.....	58
Figure 4.15:	Le bloc fonctionnel FB2.....	58

Figure 4.16:	Le bloc fonctionnel FC1.....	59
Figure 4.17:	L'étape Initial M0.0.....	60
Figure 4.18:	Temps de vidange complet du BDL (T3).....	61
Figure 4.19:	La fonction MOVE.....	62
Figure 4.20:	La première étape d'auto-préparation M14.0.....	62
Figure 4.21:	L'appel des blocs FC3 et FC4.....	63
Figure 4.22:	L'étape M8.0.....	63
Figure 4.23:	La première étape de nettoyage machine M15.0.....	64
Figure 4.24 :	Les actions des étapes (sorties).....	64
Figure 4.25 :	L'architecture générale des Vues.....	65
Figure 4.26:	La vue principale station NEP.....	66
Figure 4.27:	La vue des alarmes.....	66
Figure 4.28:	La vue des mots de passe).....	67
Figure 4.29:	La vue de l'opérateur.....	67
Figure 4.30:	La vue de l'ingénieur.....	68
Figure 4.31:	La vue des paramètres pour opérateur.....	69
Figure 4.32 :	La vue d'auto-préparation pour opérateur.....	69
Figure 4.33 :	La vue des circuits pour opérateur.....	70
Figure 4.34 :	La vue des paramètres pour l'ingénieur.....	70
Figure 4.35 :	La vue d'auto-préparation pour l'ingénieur.....	71
Figure 4.36 :	La vue des circuits pour ingénieur.....	71
Figure 4.37 :	La vue d'auto-préparation départ pour opérateur.....	72

Figure 4.38 : La vue cycle circuit pour opérateur.....	72
Figure 4.39: La vue circuit Départ pour opérateur.....	73
Figure 4.40 : La vue machine.....	73
Figure 4.41 : La vue d'auto-préparation ingénieur.....	74
Figure 4.42 : La vue des cycles circuit ingénieur.....	74
Figure 4.43 : La vue circuit départ ingénieur.....	75
Figure 4.44 : La vue circuit marche.....	75
Figure 4.45 : . La vue man/Auto.....	76
Figure 4.46: L'intégration des variables automate dans le projet WinCC.....	77
Figure 4.47 : Liaison entre l'API et le pupitre	77
Figure 4.48 : Paramétrage du pupitre.....	78
Figure 4.49 : Schéma du système.	79
Figure 4.50 : Réponse indicielle.....	80
Figure 4.51: Échangeur thermique	82
Figure 4.52 : Schéma fonctionnel de l'installation	83
Figure 4.53 : Réponse indicielle du système étudié expérimentalement	85
Figure 4.54 : Bloc fonctionnel FB41.....	86
Figure 4.55 : Bloc d'organisation OB35.....	87

Liste des tableaux

Tableau 1 : Type de régulateur selon la méthode de Ziegler & Nichols.....81

Tableau 2 : Type de régulateur selon le rapport de réglabilité r82

Tableau 3 : Les mesures des températures en C° par rapport au temps.....84

Tableau 4: Les mesures des températures en % par rapport au temps.....84

1.1 Présentation de l'usine VitaJus

VITA JUS est une société algérienne privé, qui date depuis plusieurs années.

Ses débuts ont commencé par le jus de fruits dont le conditionnement est en carton stérile et aseptique, En forme de paquets d'1 litre et des briquettes de 20cl, dans une variété de 10 parfums qui lui ont permit de prendre son envol avec l'acquisition de nouvelles chaines de production à haute cadence pour le conditionnement en paquets et bouteilles en verre.

VITAJUS fonctionne avec six(06) lignes de production et de conditionnement modernes qui sont exploitées par un personnel hautement qualifié et continuellement formé.

1.2 L'unité de production

VitaJus possède plusieurs chaines de conditionnements capables de conditionner des jus de fruits en briques (Slim-1L) et briquettes (20cl) et en bouteilles de verre de divers formats (25cl).

L'entreprise dispose de six(06) lignes de production de capacité totale de 4000 litres/heure.

- Trois (03) lignes «en Tetra Pak :

Lignes de production	Capacités: unités / heure
01 ligne volume 1L et 1.5 L	7000 et 5500
02 lignes volume 20cl	7500/ligne

- Une (01) ligne bouteille en verre :

Lignes de production	Capacités: unités / heure
01 ligne volume 25cl	20000

- Deux (02) lignes en Combibloc :

Lignes de production	Capacités: unités / heure
02 lignes volume 20cl	24000/ligne

- Les utilités relatives aux chaînes de production sont indépendantes mais disposent de la possibilité d'interconnexion.
 - Eau, vapeur.
 - Electricité, générateurs, onduleurs de puissance.
 - Air comprimé.

 - le stockage des matières premières et produits finis.

1.3 Moyens matériels

1.3.1 Machines de conditionnements

VITAJUS dispose de six (06) lignes de production et de conditionnement (figure 1.1) assurant la fabrication de différents produits qui se répartissent comme suit :

- 01 machine "**A3 Speed**" (**Tetra Pak**) assurant la fabrication de 24.000 briquettes 20cl par heure.
- 01 machine "**TBA 19**" (**Tetra Pak**) assurant la fabrication de 7500 briquettes 20 cl par heure.
- 01 machine "**A3 Flex**" (**Tetra Pak**) assurant la fabrication de 7000 packs d'1 litre par heure.
- 02 machines "**Combibloc**" assurant la fabrication de 48 000 briquettes de 20cl par heure.
- 01 machine "**Verre**" assurant la fabrication de 20.000 bouteilles de 25 Cl par heure.

1.3.2 Cuves de préparation et stockage de jus

- Cuve de préparation T1 : est utilisée pour préparer le sirop de sucre.
- Cuve de stockage T2 : est utilisée comme intermédiaire de sirop et de reconstitution boissons.
- Cuves de reconstitution boissons T3, T4, T5etT6 : remplies par le concentrée pulpe purée
- Cuve de stockage T7 : est utilisée comme intermédiaire de nectar.

Remarques:

- 1- La connexion entre les cuves (tanks) est assurée par des tuyaux INOX.
- 2- La circulation des solutions (boisson, sirop, concentré, jus) est assurée par des pompes électriques ou pneumatiques.

1.3.3 Les pasteurisateurs

La pasteurisation est un procédé de [conservation des aliments](#) par lequel ceux-ci sont chauffés à une [température](#) et durée définie, puis refroidis rapidement. La pasteurisation tire son nom des travaux de [Louis Pasteur](#) sur la stabilisation des vins au 19^e siècle.

La [pasteurisation](#) utilisée pour le [lait](#), les jus et les semi-conserves, est une technique qui consiste à soumettre les aliments à une température comprise entre 65 et 100 °C et à les refroidir brutalement.

Le pasteurisateur est un échangeur de chaleur qui comporte plusieurs sections avec un chambreur, Le chambreur est un tube calorifugé dont le volume permet selon le débit, de maintenir la température de pasteurisation durant le temps nécessaire selon l'équation suivante : $Q = V/t$.

Remarque:

Il y a d'autres équipements utilisés dans l'usine pour faciliter les tâches par exemple : laveuse, rinceuse, réchauffeuse des bouteilles, refroidisseur, dateur, étiqueteuse, cartonneuse, palettiseur...etc.

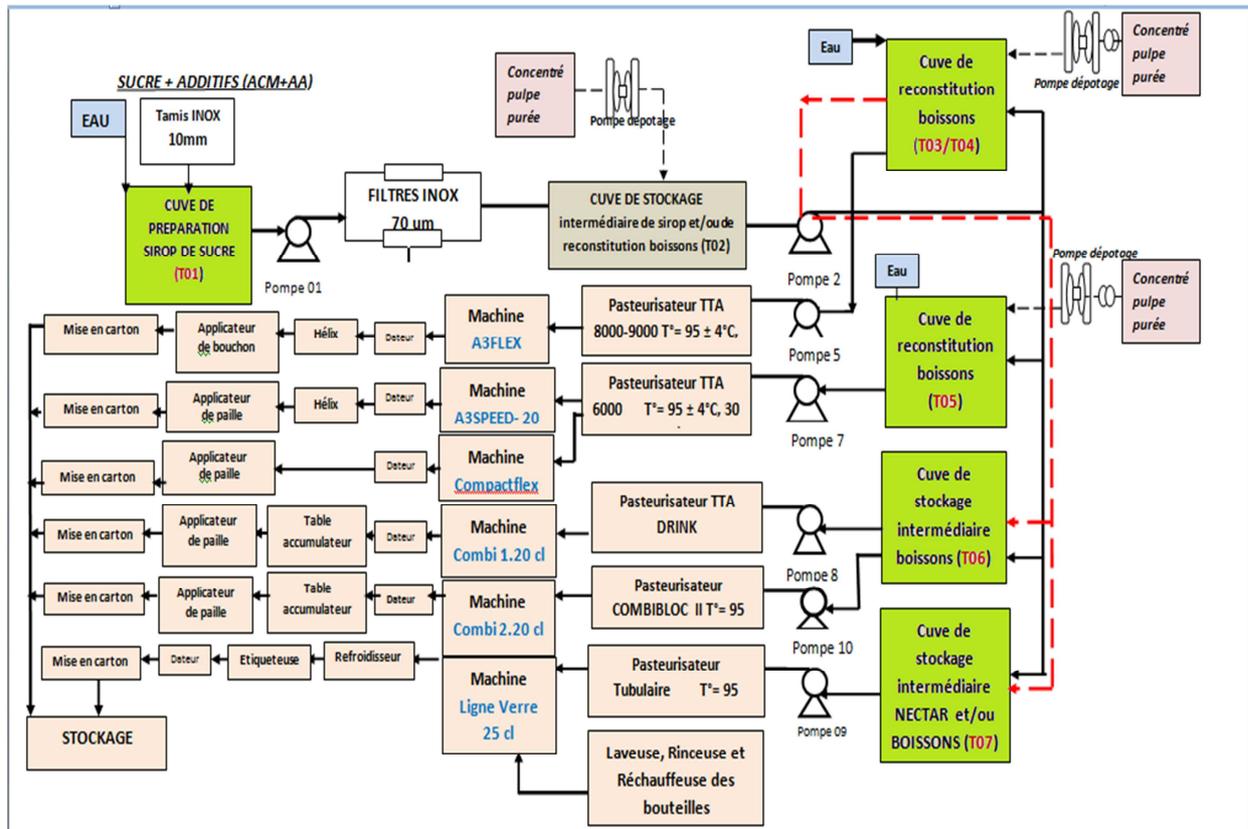


Figure 1.1. Diagramme de production de l'usine.

1.4 La station de nettoyage et son rôle dans l'usine

Tous les produits alimentaires exigent une norme d'hygiène de haut niveau en matière de procédé de fabrication. Les matières premières (par ex : lait ; jus de fruits...) sont en contact avec un certain nombre de surfaces constituant toutes les sources potentielles de contamination.

La stérilisation peut réduire, mais n'élimine pas entièrement les conséquences d'une telle contamination.

L'équipement de nettoyage et de désinfection représente donc un élément très important du processus de traitement dans une usine alimentaire.

La conception de nettoyage du matériel entrant en contact avec les produits constitue un des éléments essentiels d'une installation alimentaire. Il ne faut pas oublier que les fabricants de produits alimentaires ont pour obligation le maintien permanent de niveaux d'hygiène

élevés. Ceci vaut, bien sûr, à la fois pour l'équipement et pour le personnel de production. Cette obligation peut revêtir trois aspects :

1. Obligation commerciale :

De bons produits, sains et propres, se conservent correctement et ne présentent aucun risque pour la santé sont à l'évidence excellent pour le commerce : les clients achèteront à nouveau le même produit.

Si par contre un produit est souillé se conserve mal ou fait l'objet de plaintes auprès des autorités compétentes, c'est l'inverse qui se vérifie et la publicité qui en découle est très préjudiciable. On devra garder à l'esprit en permanence les effets potentiels d'un nettoyage, d'une qualité et de normes élevée.

2. Obligation morale :

La plupart des consommateurs utilisant les produits ne voient jamais l'usine ou le mode de fabrication de ces derniers. Ils font confiance à l'entreprise, se reposent sur sa réputation et considèrent comme allant de soi que toutes les opérations sont effectuées dans les conditions les plus hygiéniques, par du personnel qualifié, averti et conscient de ces facteurs.

3. Obligation légale :

La législation s'efforce de protéger le consommateur et l'acheteur en matière d'hygiène et de qualité. Tout manquement aux obligations légales, nationales ou locales, risque d'entraîner de graves inculpations. Les poursuites peuvent s'avérer très préjudiciables. Il vaut mieux prévenir que guérir et les entreprises sont donc tenues de satisfaire aux exigences légales et de respecter en permanence des normes élevées.

1.5 Conclusion

Dans ce chapitre nous avons présenté la société VitaJus ainsi que les moyens matériels de production de l'entreprise, en mettant en relief la station NEP dont fait l'objet de notre travail.

2.1 Introduction

Souvent, on perd de vue que le nettoyage des équipements de production n'est pas seulement indispensable pour des raisons d'hygiène. Le nettoyage contribue également à prolonger la durée de vie des équipements et permet d'économiser sur les frais d'entretien. Cependant, de nombreuses entreprises considèrent les procédures de nettoyage comme un mal nécessaire qui prend beaucoup de temps. L'une des solutions pour effectuer ces tâches de manière rapide et efficace est l'application du 'Nettoyage-En-Place'(NEP).

2.2 Principe de fonctionnement de nettoyage en place NEP

Un Nettoyage en place NEP ou Clean-in-Place CIP est un système automatique de nettoyage des installations sans démontage. Le plus souvent intégrées à la machine lors de la conception, les cuves (Tanks), tuyaux ou autres machines sont lavées à l'aide d'un circuit d'eau parallèle. Il est utilisé pour des systèmes fermés composés de tuyauteries reliant différents équipements et cuves.

Le circuit d'eau qui sert au nettoyage ainsi que les alimentations et l'éventuel retraitement des eaux de lavage sont intégrés dès la conception de l'outillage. Dans les machines automatisées, les cycles et les programmes sont également intégrés dès le début. Dans ce cas, on peut utiliser les pompes principales ou des pompes auxiliaires pour déplacer l'eau et la solution de nettoyage (par ex : la soude caustique, l'acide nitrique...).

La commande du NEP est normalement effectuée par un système de nettoyage entièrement automatique.

Avant de commencer le NEP, il faut sélectionner le circuit à nettoyer et le cycle de nettoyage (court ou long).

2.3 Les cycles de nettoyage

On peut nettoyer avec deux cycles ; le cycle court ou le cycle long.

2.3.1 Cycle court

a. Rinçage à eau

Avant de rincer le circuit à nettoyer (Tank, Ligne, machine), on vide le BDL (bac de lancement), après on le remplit par l'eau jusqu'au niveau haut (action du capteur de niveau) ; puis on rince le circuit (Tank, ligne, machine...) par l'eau froide et on remplit le BDL par l'eau, après avoir terminé le remplissage du BDL, on chauffe l'eau remplie par l'actionnement de la vanne liée à la chaudière, laisser l'eau chaude circuler pendant une durée déterminée expérimentalement, après on vide le circuit vers l'égout.

b. Rinçage par la soude (NaOH) ou l'acide de 2%

Après avoir accompli le rinçage à eau, la poussée de la soude ou l'acide commence jusqu'à ce que la conductivité étudiée soit atteinte, puis le remplissage du BDL sera actif. La circulation commence lors de l'actionnement du capteur de niveau haut BDL, pendant un temps prédéfini, après avoir fini les étapes on récupère la soude ou l'acide de concentration supérieure à 0.5%.

c. Rinçage finale par l'eau

Dernière opération, on rince l'échangeur à plaque et le circuit par l'eau froide et l'eau chaude puis on vidange vers l'égout.

2.3.2 Cycle Long

- Rinçage à eau (froide et chaude).
- Rinçage par la soude caustique (NaOH) à 2%.
- Rinçage intermédiaire par l'eau.
- Rinçage par l'acide nitrique (HNO₃) à 2%.
- Rinçage finale par l'eau.

Remarque:

- 1- On utilise la soude caustique NaOH pour éliminer les matières organiques.
- 2- On utilise l'acide nitrique HNO₃ pour éliminer les matières minérales.

La figure 2.1 représente le schéma synoptique de la station NEP

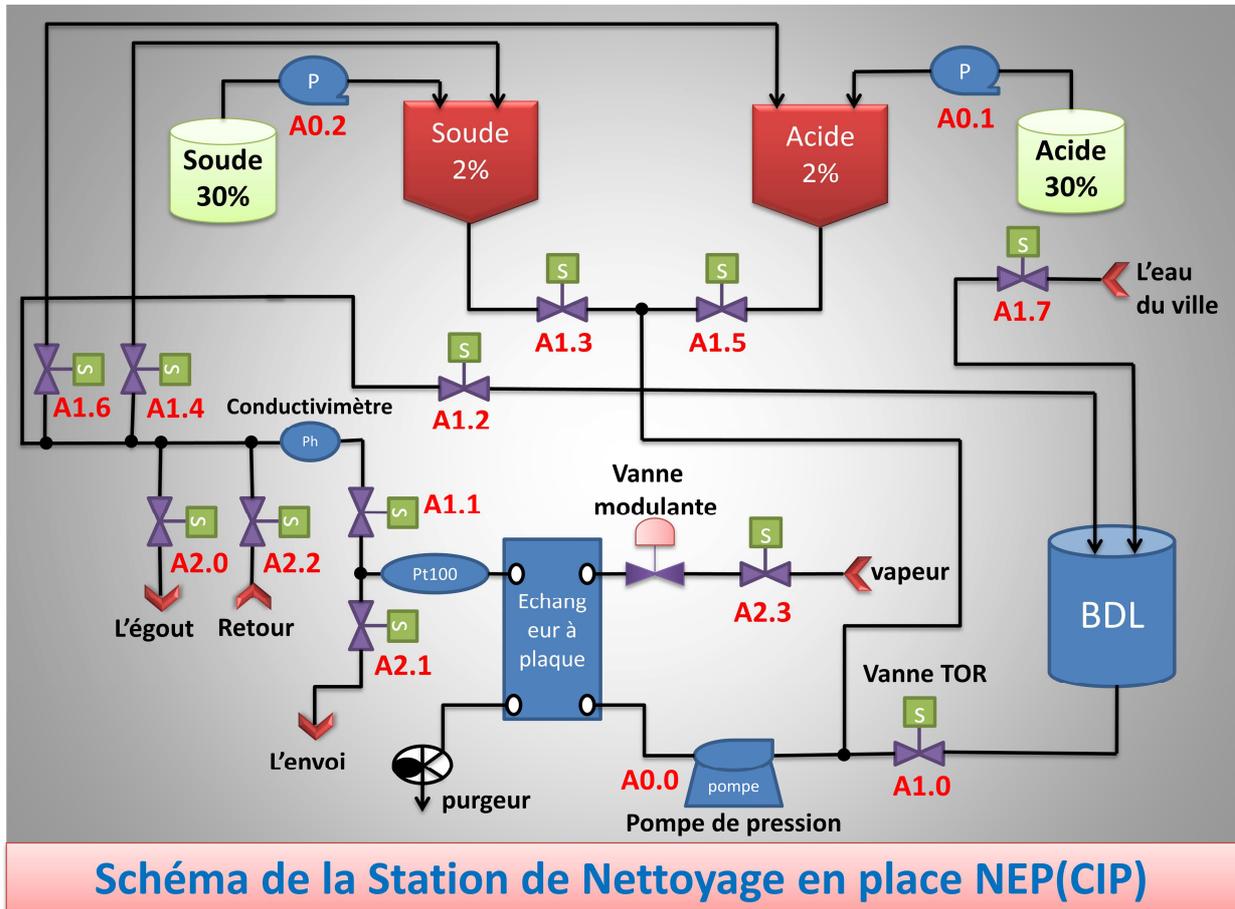


Figure 2.1. Schéma synoptique de la station NEP.

2.4 Description des composants utilisés

L'automatisation du système CIP de Vita jus est basée sur les dispositifs suivants :

2.4.1 Automate programmable industriel S7-200 (SIEMENS)

L'automate programmable industriel S7-200 /CPU 224 de marque SIEMENS (figure 2.2) situé dans l'armoire de contrôle de la station NEP (Vita jus), c'est un API compact à boîtier unique est constituée d'unité d'alimentation, de processeur, de mémoire de données égale 8192 octets, de nombre d'entrée/sortie limité (14 E /10S), et une possibilité de mètre sept modules d'extension.

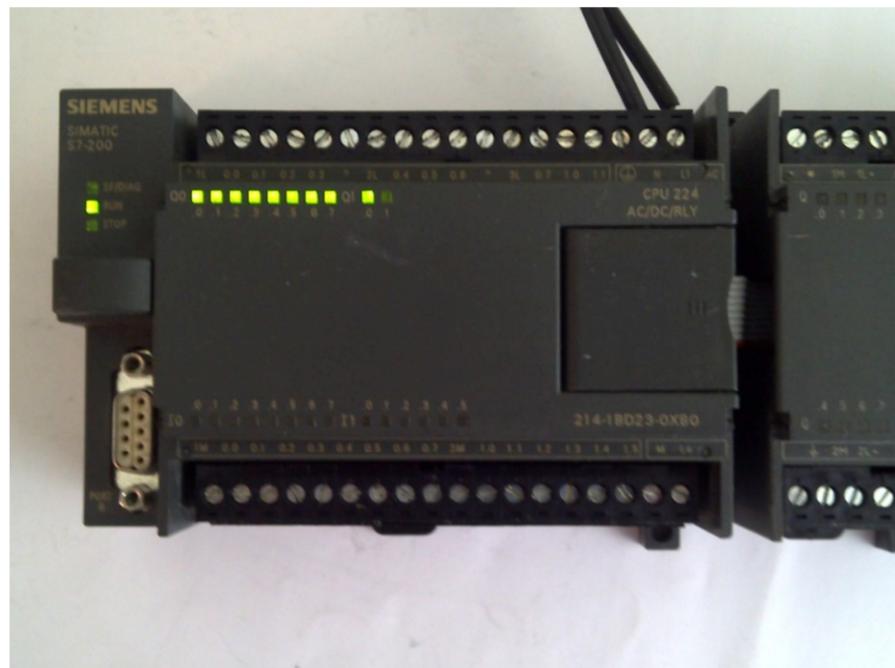


Figure 2.2. API S7-200 (CPU224).

2.4.2 Les Pompes

a. Pompe de pression

La pompe de pression c'est une pompe électrique à turbine (figure2.3) qui est situé juste après le BDL (bac de lancement) utilisée pour le pompage d'eau et de la solution de nettoyage (soude/acide) vers les circuits.

Le débit maximal de cette pompe 15000 L/h.



Figure 2.3. Pompe de pression électrique.

b. Pompe de retour CIP

C'est une pompe électrique à turbine (figure2.3), elle est utilisée pour récupérer à partir des tanks l'eau et la solution de nettoyage, son emplacement est juste après les tanks.

c. Pompes doseuses de la soude et l'acide

Ce sont des pompes pneumatiques (figure2.4), situés devant les tanks d'auto-préparation pour la préparation des solutions de nettoyage (soude/acide).

Le débit maximal de ses pompes est 1500 L/h.



Figure 2.4. Pompe doseuse.

2.4.3 Le réservoir de circulation (BDL)

Le réservoir de circulation ou bac de lancement placé au début de la station NEP est utilisé pour maintenir l'arrivée d'eau et du liquide (solution de nettoyage) en circulation au niveau souhaité. Un système à contacteur de niveau surveille le niveau du réservoir.

2.4.4 L'Échangeur de chaleur à plaques

L'Échangeur de chaleur à plaques (figure 2.5), situé près de la pompe de pression est utilisé pour réchauffer l'eau et la solution de nettoyage en utilisant de la vapeur.

Le cœur de l'échangeur est constitué d'un paquet de plaques avec des ouvertures pour la circulation des fluides (l'eau et la vapeur). Ces fluides circulent entre les plaques, mélangés alternativement tête-bêche pour créer les turbulences nécessaires.



Figure 2.5. Échangeur à plaques (*FUNKE*).

2.4.5 Transmetteur de conductivité (conductivimètre)

L'appareil de contrôle et transmetteur de conductivité (figure 2.6), situé dans la conduite de retour est utilisé pour mesurer la conductivité des solutions de nettoyage (la soude et l'acide) dans le circuit de nettoyage.

Le conductivimètre a une grande importance également dans la boucle de tri, c'est-à-dire lors du changement de solution nettoyante. Par exemple, lorsque le premier rinçage est terminé, la solution de soude est envoyée dans les installations. La soude va donc pousser les eaux de rinçage qui seront envoyées à l'égout. Dans la conduite de retour, le conductivimètre va détecter le moment où la solution de soude sera pure (soude ou acide supérieure à 0.5%) et pourra être utilisée pour le nettoyage, par conséquent son mauvais fonctionnement ou une trop faible sensibilité peut entraîner une surconsommation des produits chimiques qui seront chauffés et ensuite traités.



Figure 2.6. Conductivimètre E+H.

2.4.6 Les capteurs

a Capteur de température Pt100

La PT 100 est une sonde de température (figure 2.7) placée dans la conduite d'envoi, elle est régulièrement utilisée dans l'industrie pour la mesure des températures. PT représente le mot platine qui est le matériel principal de la sonde, PT100 détermine la valeur ohmique (100 ohms) de la sonde à zéro degré centigrade, La résistance de la sonde s'élève en fonction de l'augmentation de la température ambiante. le model utilisé varie sur une plage de 0c° à +150c°.



Figure 2.7. Sonde de température Pt100.

b Capteur de niveau

Le type de capteur utilisé c'est un détecteur de niveau conductif (figure2.8), qui est constitué de deux électrodes. Lorsque le liquide atteint le seuil déterminé par la position des électrodes, le milieu liquide ferme le circuit électrique. Un signal de commutation est généré par la soudaine augmentation de courant, il est placé dans le BDL(bac de lancement).



Figure 2.8. Capteur de niveau.

La Détection par conductivité apporte une solution simple et économique pour la détection de niveaux pour les produits conducteurs d'électricité, même si leur conductivité est faible.

2.4.7 Les électrovannes

Une électrovanne ou électrovalve est un dispositif commandé électriquement, permettant d'autoriser ou d'interrompre par une action mécanique la circulation d'un fluide ou d'un gaz dans un circuit.

Il existe deux types d'électrovannes : « tout ou rien » et « proportionnelle ».

- Les électrovannes dites de « tout ou rien » ne peuvent s'ouvrir qu'en entier ou pas du tout. L'état change suivant qu'elles sont alimentées électriquement ou non.

- Les électrovannes proportionnelles sont celles qui peuvent être ouvertes avec plus ou moins d'amplitude en fonction du besoin.

a Les vannes TOR

La vanne tout ou rien que l'on trouve sur le procédé est une vanne pneumatique. C'est donc une pression d'air comprimé qui lui permet de bloquer ou de libérer la canalisation. L'air comprimé est admis ou non dans le servomoteur de la vanne procédé grâce à une électrovanne dite vanne de commande ou pilote.

Cette vanne ne peut prendre que deux positions, ouverte ou fermée. On traduit en termes de course du clapet 0% ou 100% d'ouverture. Son rôle principal est d'assurer des fonctions de sécurité et d'utilités, on peut aussi dans certains cas les utiliser pour réguler une grandeur physique (Régulation discontinue). Lorsqu'elles sont fermées, ces vannes sont étanches.

Symbolisation :



Vanne TOR
Pneumatique
Simple effet



Vanne TOR
Pneumatique
Double effet



Vanne TOR
Electromagnétique

b Les vannes modulantes ou de régulation

La vanne automatique ou vanne de régulation (figure 2.9) située avant l'échangeur est commandée par un servomoteur, utilisé en régulation industrielle comme organe modulant le débit ou la pression d'un fluide.

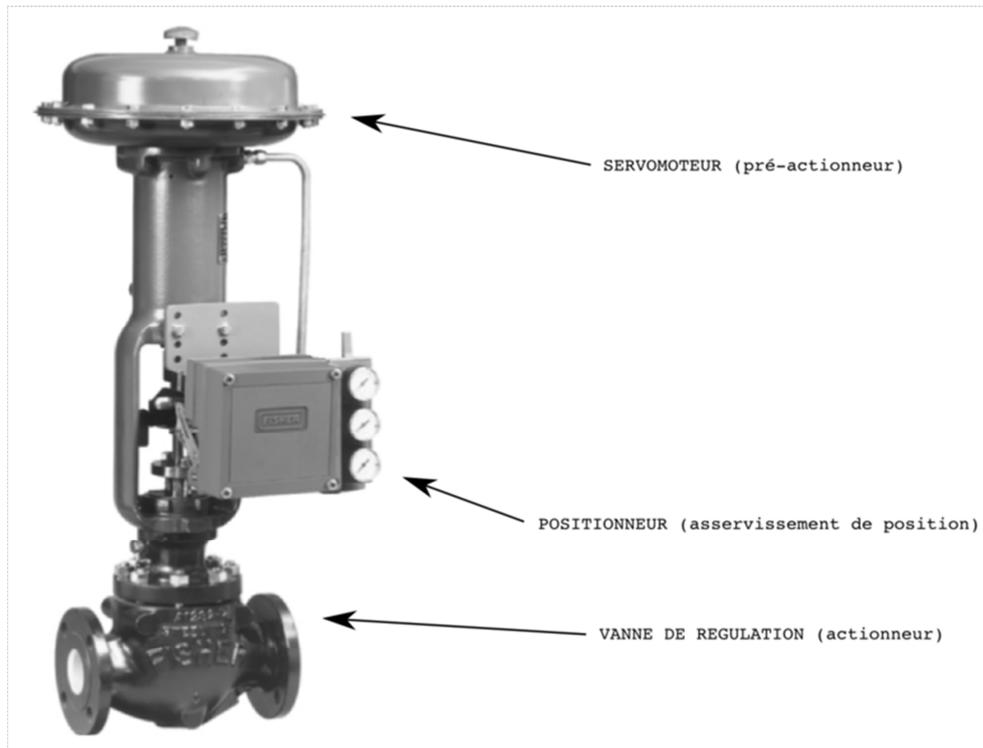


Figure 2.9. Vanne de régulation (modulante).

2.4.8 Le purgeur de vapeur

Un purgeur de vapeur (figure 2.10), est un organe de l'échangeur, qui permet d'évacuer le condensat (c.-à-d. la vapeur d'eau condensée) et autres gaz non condensables sans laisser échapper la vapeur. Dans le secteur industriel, la vapeur d'eau est souvent utilisée comme moyen de chauffage.

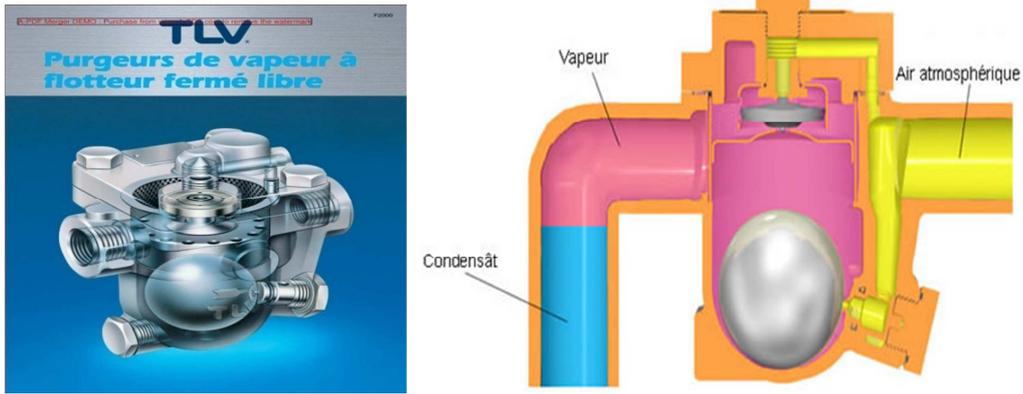


Figure 2.10. Purgeur de vapeur.

2.4.9 Le clapet anti-retour

Un clapet anti-retour (figure 2.11) est un dispositif permettant de contrôler le sens de circulation d'un fluide quelconque placé dans les interconnexions des tuyaux. Il permet le passage d'un liquide, d'un gaz, d'air comprimé, ... dans un sens et bloque le flux si celui-ci venait à s'inverser.



Figure 2.11. Clapet anti-retour.

2.4.10 Le régulateur de température PID externe

C'est un régulateur de température PID (figure 2.12) situé dans l'armoire de contrôle de la station NEP (Vita jus) qu'on va remplacer dans le cadre de notre projet par un PID interne propre à l'API, il est utilisé pour réguler la température au niveau du circuit de nettoyage.



Figure 2.12. Régulateur de température PID (OMRON).

2.5 Préparation des solutions de nettoyage (l'Auto-préparation)

Avant de commencer le nettoyage des cuves, lignes ou machines, on prépare la solution de nettoyage (soude/acide).

Pour obtenir un résultat de nettoyage optimal, il est primordial d'avoir la puissance et la température correcte de la solution de nettoyage à utiliser.

L'auto-préparation se fait comme suit:

Etape 1 : remplissage de tank soude ou acide par l'eau jusqu'à l'arrivée au niveau moyen du tank (3000L).

Etape 2 : dosage de la soude ou l'acide dans le tank qui est remplie par l'eau pendant une durée bien déterminé.

Etape 3 : mélange des deux solutions (eau+soude) ou (eau+acide) et vérification de la conductivité ou la concentration de solution.

Etape 4 : si la concentration de la solution mélangée est entre 1.7% et 2.3%, on rince le circuit avec l'eau et on l'évacue vers l'égout.

Etape 5 : si la concentration de la solution mélangée est inférieure à 1.7%, on ajoute la soude ou l'acide pendant un temps et on vérifie la concentration.

Etape 6 : si la concentration de la solution mélangée est supérieure à 2.3%, on ajoute l'eau pendant un temps et on vérifie la concentration.

Remarques :

1- La méthode utilisée pour calculer le temps de dosage de la soude ou l'acide est :

$$\left. \begin{array}{l} 1500 \text{ L} \longrightarrow 60 \text{ min} \\ 204 \text{ L} \longrightarrow T \end{array} \right\} T = 8 \text{ min et } 16 \text{ s (temps de dosage).}$$

- pour obtenir une solution de la soude ou l'acide à 2%, il faut mélanger 204L de la soude ou l'acide à une quantité de 3000L d'eau.

- le débit de la pompe doseuse est 1500L/h.

2- Le transmetteur de conductivité (conductivimètre) c'est lui qui assure la vérification de concentration des solutions de nettoyage.

2.6 Conclusion

Nous avons au cours de ce chapitre présenté le fonctionnement de la station NEP et l'importance de son rôle dans l'unité VITAJUS.

Et nous avons passé en revue tous les éléments techniques employés pour le contrôle du mélange utilisé pour le nettoyage.

3.1 Introduction

Dans le cadre d'une évolution conduisant à une automatisation de plus en plus globale, l'automate est de moins en moins acquis seul, et même si c'est le cas, il doit pouvoir se connecter à d'autres matériels à processeur, et dialoguer avec les agents d'exploitation.

Les types de communication supportés par les API modernes sont : la communication avec un opérateur par un pupitre ou un terminal industriel : ils permettent une communication homme-machine, et ce dans les deux sens (clavier, ...).

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. Et c'est pour cette raison qu'on voulait remplacer les dispositifs de commande placés dans l'industrie (API S7-200 + pupitre manuel) par un autre dispositif moderne (API S7-300 + pupitre digital ou tactile).

3.2 Généralités sur les API

3.2.1 Historique

Au début des années 50, les ingénieurs étaient déjà confrontés à des problèmes d'automatismes, les composants de base de l'époque étaient les relais électromagnétique à un ou plusieurs contacts. Les circuits conçus comportaient des centaines voire des milliers de relais. Le transistor n'était connu que comme un composant d'avenir et les circuits intégrés étaient inconnus.

Vers 1960, les semi-conducteurs (transistors, diodes) sont apparus dans les automatismes sous forme de circuits digitaux. Ce n'est quelques années plus tard, que l'apparition des circuits intégrés a amorcé une révolution dans la façon de concevoir les automatismes. Ceux-ci étaient très peu encombrants et leur consommation était des plus réduite. On pouvait alors concevoir des fonctions de plus en plus complexes à des coûts toujours décroissants.

C'est en 1969 que les constructeurs américains d'automobiles (General Motors en particulier) ont demandé aux firmes fournissant le matériel d'automatisme des systèmes plus évolués et plus souples pouvant être modifiés simplement sans coût exorbitants.

Les ingénieurs américains ont résolu le problème en créant un nouveau type de produit nommé automates programmables. Ils n'étaient rentables que pour des installations d'une

certaine complexité, mais la situation a très vite changée, ce qui a rendu les systèmes câblés obsolètes.

De nombreux modèles d'automates sont aujourd'hui disponibles ; depuis les nano automates bien adaptés aux machines et aux installations simples avec un petit nombre d'entrées/sorties, jusqu'aux automates multifonctions capables de gérer plusieurs milliers d'entrées/sorties, et destinés au pilotage de processus complexes.

3.2.2 Définition

Un Automate Programmable Industriel API dit Programmable Logic Controller PLC dans le langage industriel, est une machine électronique spécialisée dans la conduite et la surveillance en temps réel de processus industriels et tertiaires. Il exécute une suite d'instructions introduites dans ses mémoires sous forme de programme, et s'apparente par conséquent aux machines de traitement de l'information.

Trois caractéristiques fondamentales le distinguent des outils informatiques tels que les ordinateurs utilisés dans les entreprises et le tertiaire :

- il peut être directement connecté aux capteurs et pré actionneurs grâce à ses Entrées/sorties industrielles.
- il est conçu pour fonctionner dans des ambiances industrielles sévères, (Température, vibration, micro-coupures de la tension d'alimentation, Parasite, etc.).
- enfin, sa programmation à partir des langages spécialement développés pour le Traitement des fonctions d'automatisme facilite son exploitation et sa mise en œuvre.

Selon la norme française EN 61131-1, un automate programmable est un :

« Système électronique fonctionnant de manière numérique, destiné à être utilisé dans un environnement industriel, qui utilise une mémoire programmable pour Le stockage interne des instructions orientées utilisateurs aux fins de mise en œuvre des fonctions spécifiques, telles que des fonctions de logique, de mise en séquence, de temporisation, de comptage, et de calcul arithmétique, pour commander au moyen d'entrées et de sorties tout ou rien ou analogiques divers types de machines ou de processus. L'automate programmable et ses périphériques associés sont conçus pour pouvoir facilement s'intégrer à un système d'automatisme industriel et être facilement utilisés dans toutes leurs fonctions prévues. »

De manière générale, un API est structuré autour de plusieurs éléments de base (figure3.1) que sont l'unité de traitement, la mémoire, l'unité d'alimentation, les interfaces d'entrées-sorties, l'interface de communication et ce périphérique de programmation.

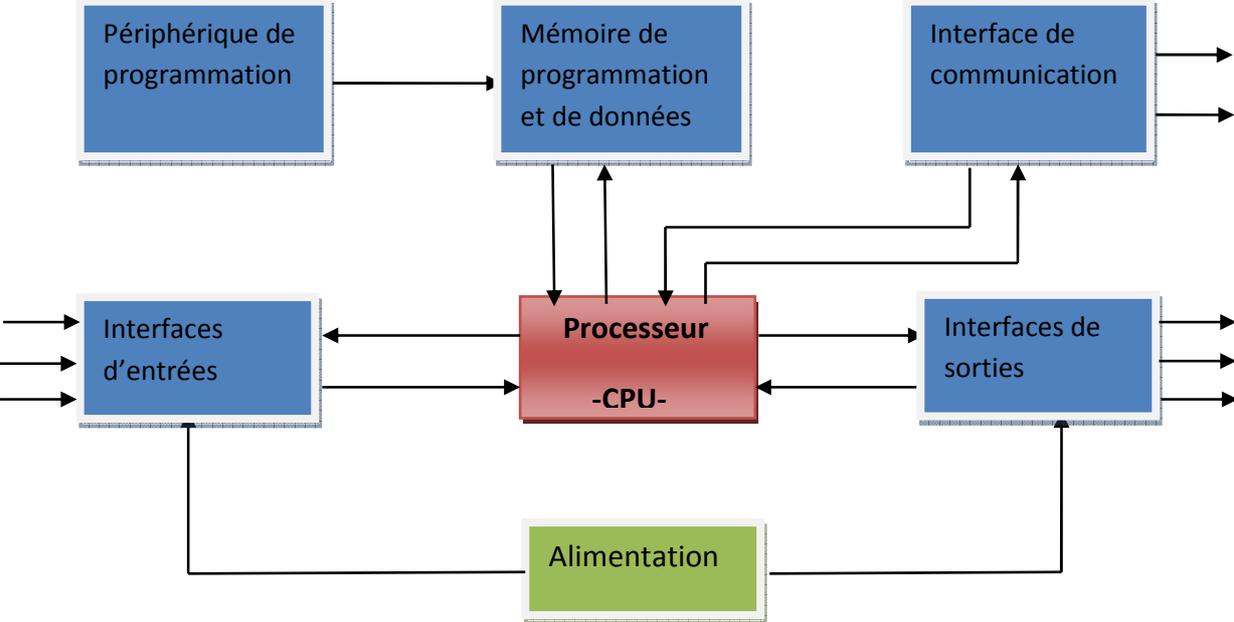


Figure 3.1. Structure générale d'un API.

3.2.3 Logiciel de programmation des API SIEMENS <<Step7>>

STEP7 fait partie de l'industrie logicielle SIMATIC. Il représente le logiciel de base pour la configuration et la programmation de systèmes d'automatisation.

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

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

Il s'exécute sous les systèmes d'exploitation de Microsoft à partir de la version Windows 95. Et s'adapte par conséquent à l'organisation graphique orientée objet qu'offrent ces systèmes d'exploitation.

3.2.4 Applications de STEP7

Il est constitué de deux types d'application ; applications de bases et applications en options.

a. Applications de bases STEP7

Le logiciel STEP7 met à disposition les applications de bases suivantes

- Le gestionnaire de projets.
- La configuration du matériel.
- L'éditeur de mnémoniques.

- L'éditeur de programmes CONT, LOG, LIST.

➤ **Gestionnaire de projets SIMATIC Manager**

Le gestionnaire de projets SIMATIC Manager (figure3.2) gère toutes les données relatives à un projet d'automatisation, il démarre automatiquement les applications requises pour le traitement des données sélectionnées.

Le schéma suivant représente la fenêtre qui apparaît à l'ouverture du SIMATIC Manager.

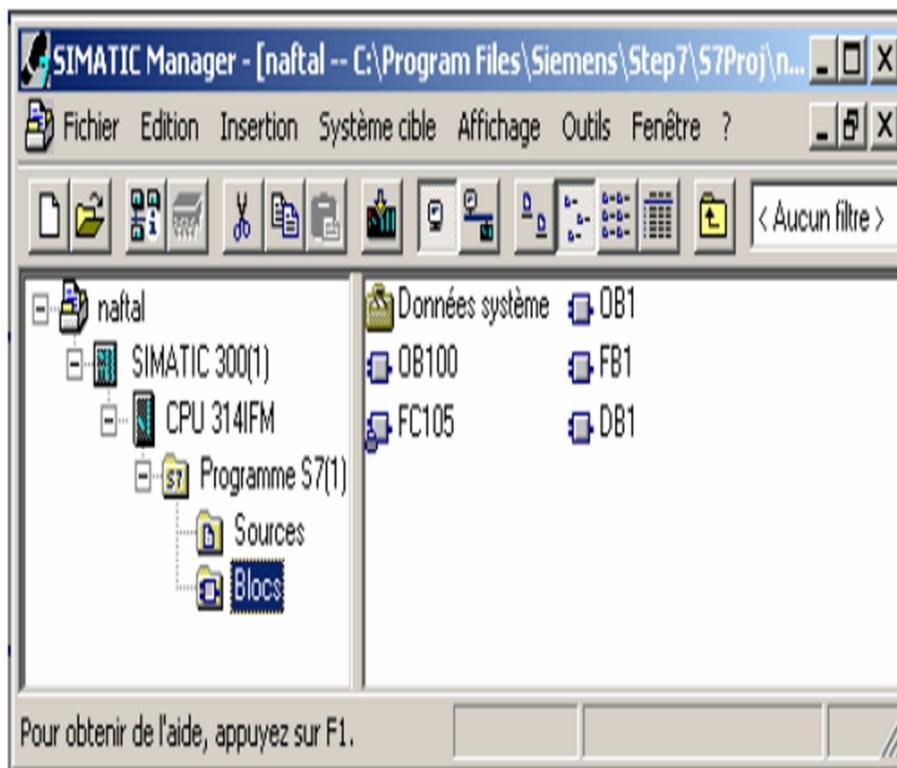


Figure 3.2. Fenêtre du SIMATIC manager.

➤ **Configuration du matériel HW Config**

HW Config est utilisé pour configurer et paramétrer le support matériel dans un projet d'automatisation.

➤ **Editeur de mnémoniques**

Il permet la gestion de toutes les variables globales. En effet, il définit des désignations symboliques et des commentaires pour les signaux du processus (Entrées/Sorties), les mémentos, les blocs de données, les temporisations et les compteurs.

La table des mnémoniques qui en résulte est mise à disposition à toutes les applications. La modification de l'un des paramètres d'une mnémonique est de ce fait reconnue automatiquement par toutes les applications.

➤ ***Editeur de programme***

Les langages de base proposés sont :

- Le schéma à contact (CONT), langage graphique similaire aux schémas de circuits à relais, il permet de suivre facilement le trajet du courant.
- La liste d'instruction (LIST), langage textuel de bas niveau, à une instruction par ligne, similaire au langage assembleur.
- Le logigramme (LOG), langage de programmation graphique qui utilise les boîtes de l'algèbre de BOOLE pour représenter les opérations logiques.
- L'éditeur de programmes permet aussi la visualisation et le forçage des variables.

b. La programmation STEP7

Pour concevoir un projet avec STEP7, il existe 2 approches (figure 3.3):

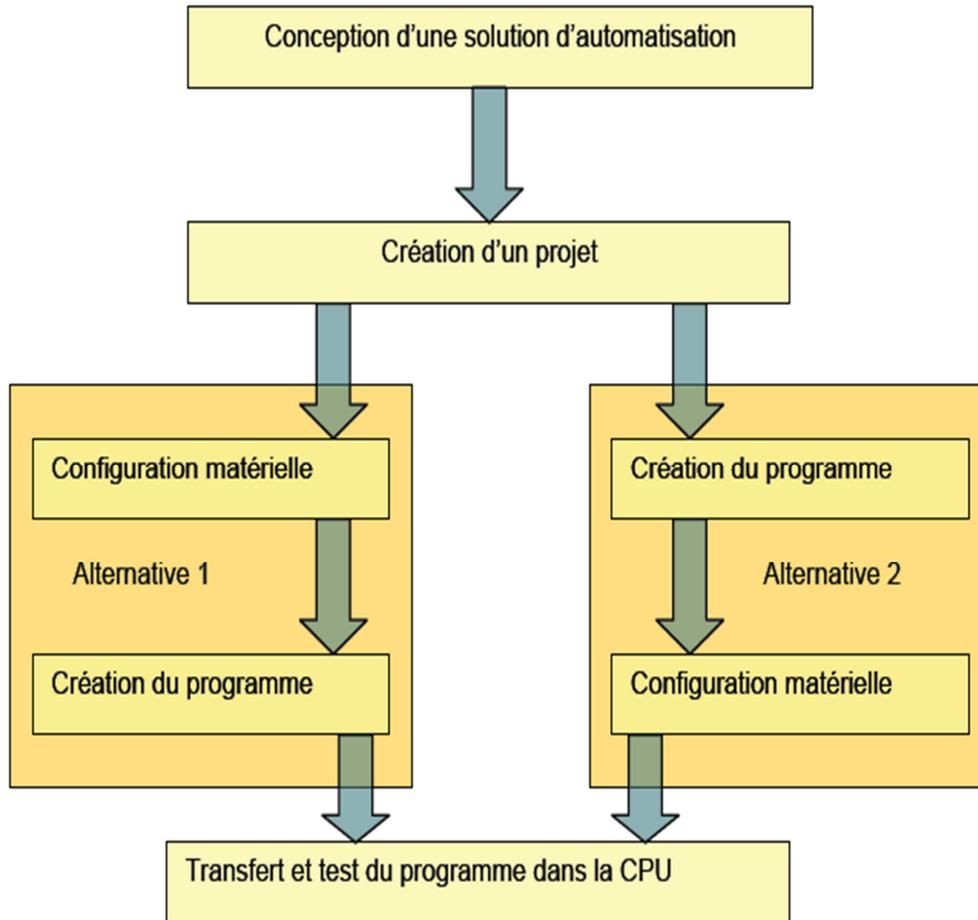


Figure 3.3. Organigramme pour la création de projets sous STEP 7.

Il faut noter que pour un système contenant beaucoup de variables, la seconde alternative n'est pas très pratique.

➤ **Création du projet avec STEP7 :**

Pour créer un projet avec STEP 7, on peut lancer l'assistant de création de projet de STEP7 (figure 3.4), ou créer directement un projet que l'on configurera soi-même.

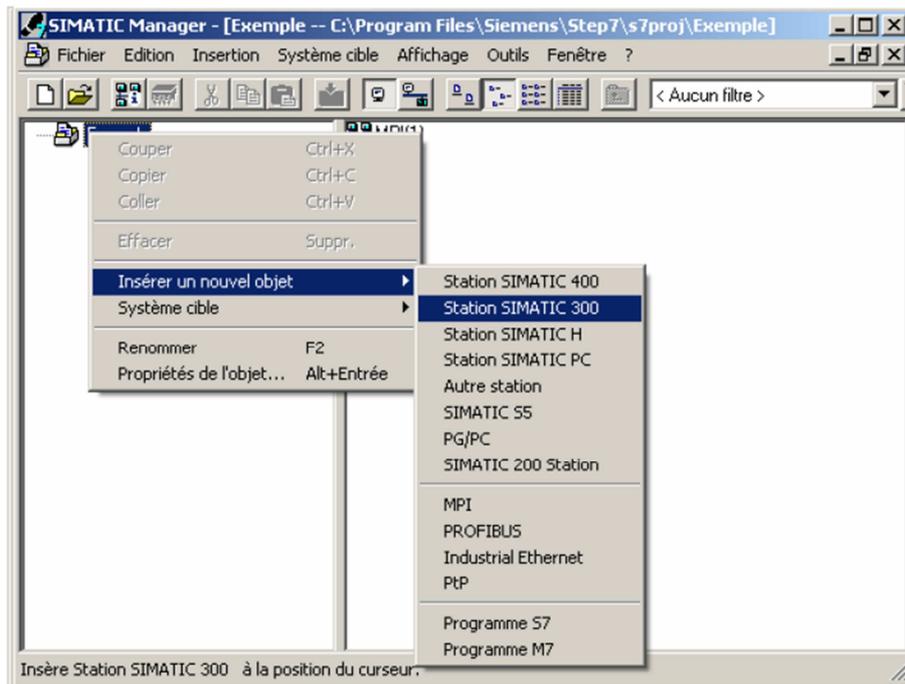


Figure 3.4. Création d'un projet sans l'assistant.

c. Configuration matérielle :

La configuration matérielle (figure3.5) est une étape très importante, elle permet de reproduire à l'identique le système utilisé (alimentation, CPU, modules etc....).

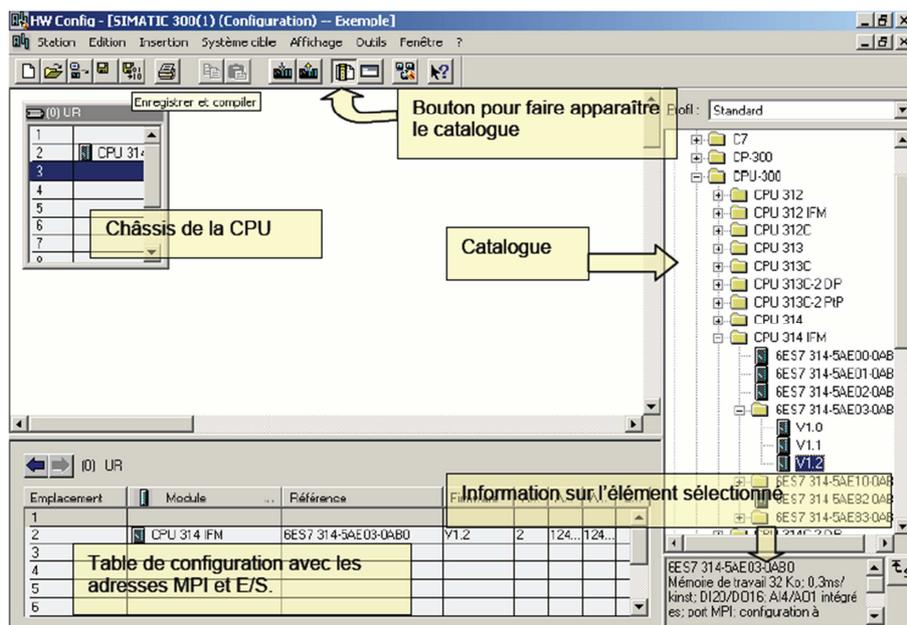


Figure 3.5. Configuration du matériel.

Le paramétrage des modules est réalisé automatiquement au démarrage de la CPU. Ainsi, le remplacement d'un module est ainsi possible sans nouveau paramétrage.

d. Définition des mnémoniques

Il faut définir les variables qui vont être utilisées lors des étapes de programmation. L'utilisation de noms communs est plus aisée que la manipulation de chiffres (ex : utiliser « moteur » au lieu du bit de sortie A0.0).

L'utilisation de la table des mnémoniques (figure3.6) consiste à :

- Donner un nom à la mnémonique dans la première colonne.
- Donner la variable associée à cette mnémonique dans la seconde colonne.
- Le type de la donnée est automatiquement généré par STEP 7.
- Ecrire éventuellement un commentaire dans la colonne prévu à cet effet.
-

Etat	Mnémonique	Opérande	Type de données	Commentaire
2	auto_préparation/acide	M 41.1	BOOL	
3	auto_preparation/soude	M 41.2	BOOL	
4	autopréparation	FB 1	FB 1	
5	bouton arret	M 41.3	BOOL	
6	bouton auto_preparation	M 41.4	BOOL	
7	bouton depart	M 40.4	BOOL	
8	bouton nett ligne pack	M 40.2	BOOL	
9	bouton nett ligne verre	M 40.3	BOOL	
10	bouton nettoyage Tank1	M 40.0	BOOL	
11	bouton nettoyage Tank5	M 40.1	BOOL	
12	bouton rearmement	M 40.5	BOOL	
13	CIP en marche	A 0.4	BOOL	
14	CIP en panne	A 0.5	BOOL	
15	CIP pret	A 0.3	BOOL	
16	circulation de l'acide	T 8	TIMER	
17	circulation de la soude	T 7	TIMER	

Figure 3.6. La fenêtre d'édition de mnémoniques.

3.2.5 Organisation d'un programme utilisateur

Le logiciel de base STEP 7, permet de structurer le programme utilisateur.

Cette structuration est réalisée par la subdivision du programme en différentes parties autonomes. Il en résulte les avantages suivants :

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

a. Hiérarchisation dans un projet

Chaque programme utilisateur est structuré sous forme de projet (figure3.7).

Un projet représente l'ensemble des données et programmes d'une solution d'automatisation et se trouve à la tête d'une hiérarchie d'objets qui sont :

- objet projet.
- objet station.
- objet Modules programmables.
- objet Programme S7/M7.
- objet dossier Sources.
- objet dossier Blocs.

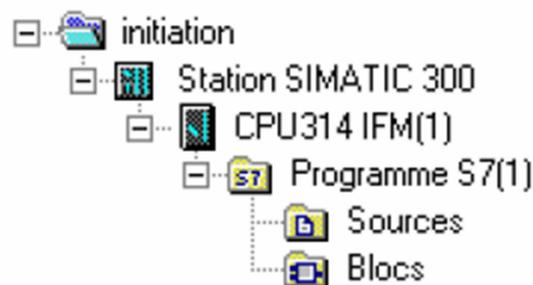


Figure 3.7. L'hiérarchie d'un projet STEP 7.

b. Objet dossier Blocs

Il englobe les blocs de code (OB, FB, SFB, FC, SFC) qui contiennent les programmes qu'on doit charger dans la CPU, et les blocs de données (DB d'instance et DB globaux) qui contiennent les paramètres du programme.

➤ ***Les blocs d'organisation (OB)***

Ils constituent l'interface entre le système d'exploitation et le programme utilisateur. Ils sont appelés par le système d'exploitation et gèrent le traitement de programme cyclique et déclenché par alarme, ainsi que le comportement à la mise en route de l'automate programmable et le traitement des erreurs. On peut programmer les blocs d'organisation et déterminer ainsi le comportement de la CPU.

Le bloc d'organisation **OB1** est généré automatiquement lors de la création d'un projet, il représente le programme principal (MAIN dans le langage C). En effet, c'est le programme appelé cycliquement par le système d'exploitation.

Les autres blocs, existant dans le projet seront exécutés à leur appel par l'OB1.

➤ ***Les blocs fonctionnels (FB)***

▪ **Le FB :**

C'est un sous programme écrit par l'utilisateur, il facilite la programmation de fonctions complexes souvent utilisées. Il exécute par l'appel d'autre bloc de code.

Un bloc de données d'instance, qui constitue sa mémoire, lui est associé. Ce dernier contient les paramètres transmis au FB ainsi que les variables statiques.

▪ **Le bloc fonctionnel système (SFB) :**

C'est un bloc fonctionnel intégré à la CPU S7. Les SFB font partie du système d'exploitation, par conséquent, ils ne sont pas chargés en tant que partie du programme. Comme les FB, les SFB sont des blocs avec mémoire. On doit donc également créer pour les SFB des blocs de données d'instance que l'on charge dans la CPU en tant que partie du programme.

Ils sont utilisés pour des fonctions spéciales intégrées de la CPU 314 IFM, comme ils peuvent être utilisés pour la communication via des liaisons configurées.

➤ *Les fonctions (FC)*

▪ **La Fac :**

Elle contient des routines pour les fonctions fréquemment utilisées, comme le renvoi d'une valeur au bloc appelant. Elle est sans mémoire et contient uniquement des variables temporaires qui sont sauvegardées dans la pile de données locales et perdues à l'achèvement de cette fonction.

Mais elle peut faire appel à des blocs de données globaux pour la sauvegarde de ses données.

▪ **La fonction système (SFC) :**

C'est une fonction intégrée dans la CPU S7, pré-programmée et testée. Elle est appelée à partir du programme. Comme ces fonctions font partie du système d'exploitation, elles ne sont pas chargées en tant partie du programme. Comme les FC, les SFC constituent des blocs sans mémoire.

Parmi les fonctionnalités qu'elles proposent :

Le contrôle du programme, la gestion des alarmes horaires et temporisées, la mise à jour de la mémoire image du processus, l'adressage de modules et la création de messages relatifs aux blocs.

➤ **Les blocs de données d'instance (DB d'instance) :**

Associé à chaque bloc fonctionnel, il contient les paramètres effectifs et les données statiques du FB.

On peut utiliser plusieurs DB pour un même FB ; par exemple, un FB pour la commande de plusieurs moteurs, les données de chaque moteur sont sauvegardées dans différents DB.

➤ **Les blocs de données globaux (DB) :**

A l'opposé des DB d'instance qui ne sont associés qu'aux blocs fonctionnels, les DB globaux (figure3.8) servent à l'enregistrement de données utilisateur pouvant être utilisées par tous les autres blocs de code.

Remarque :

A l'appel d'un FB, le DB d'instance lui correspondant est automatiquement généré une fois qu'il est inséré à son emplacement.

Mais si on veut créer un bloc de données à partir du dossier bloc, on procède comme suit :

- On clique avec le bouton droit de la souris, puis sur Insérer un nouvel objet>Bloc de données,
- Ou Dans la fenêtre SIMATIC Manager, on clique sur le menu Insertion> BlocS7>bloc de données.

Dans les deux cas, le type du bloc de données sera demandé (DB d'instance ou DB global).

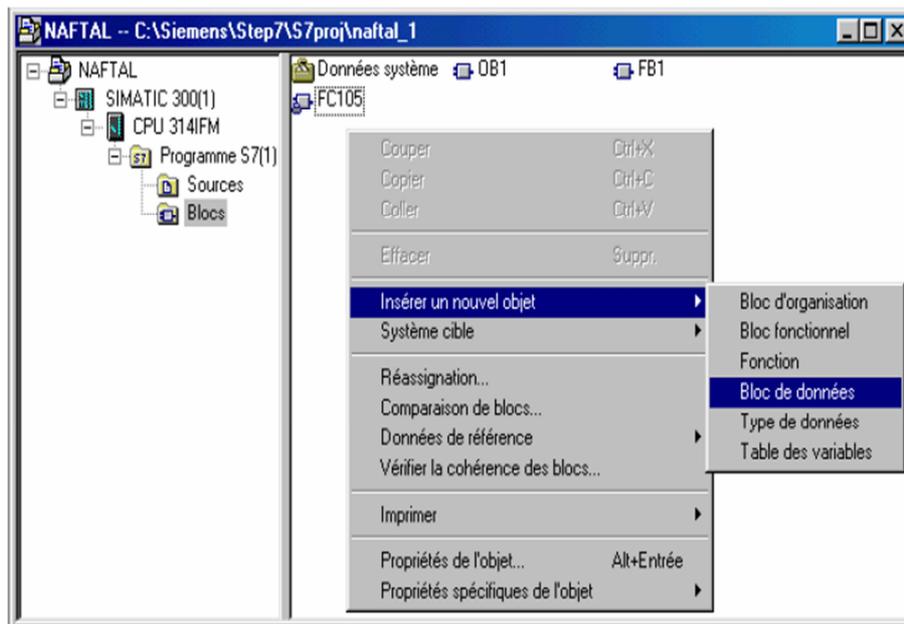


Figure 3.8. Création de blocs de données.

3.2.6 Les langages de programmation

a. *ladder*

C'est un langage entièrement graphique, bien adapté au traitement logique simple de type combinatoire. Il utilise les symboles graphiques comme des contacts à ouverture ou à fermeture et des bobines. Ainsi un programme écrit en langage à contacts (LD) ne se présente pas sous la forme d'une liste d'instructions, mais comme une liste de schémas développés classique.

Des blocs d'opérations logiques ou arithmétiques pré-programmés peuvent être insérés dans les réseaux de contacts.

Les éléments principaux d'un réseau en langage Ladder (figure 3.9) sont :

- Contacte à fermeture 
- Contacte à ouverture 

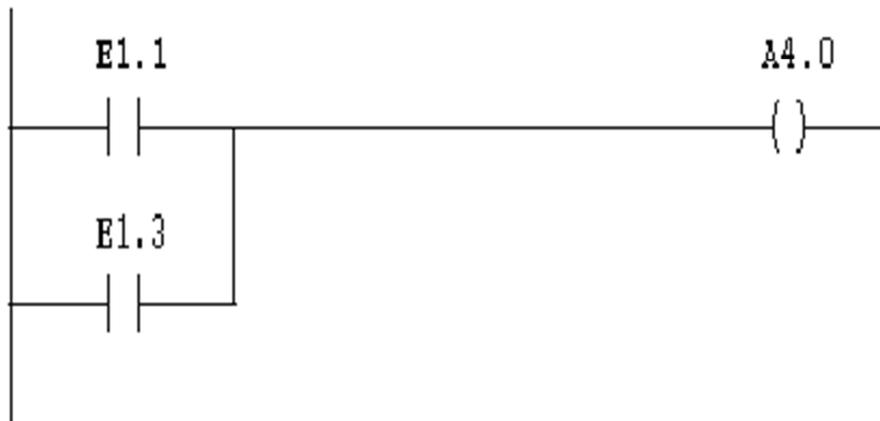


Figure 3.9. Exemple d'un « OU » logique avec le langage ladder

(A4.0 n'est mise à 1 que si : E1.1 **ou** E1.3 sont à 1).

b. *Le LOG (FBD)*

Le langage LOG est un langage graphique (figure 3.10) qui utilise des boîtes logiques d'algèbre de BOOL. La base de ce langage est la logique binaire, mais on peut aussi faire des opérations plus complexes telles que les opérations mathématiques à l'aide de blocs.

Les instructions LOG peuvent être sous formes :



- sous forme d'éléments : comme une inversion logique.
- sous forme de boîte avec opérande : comme une affectation.

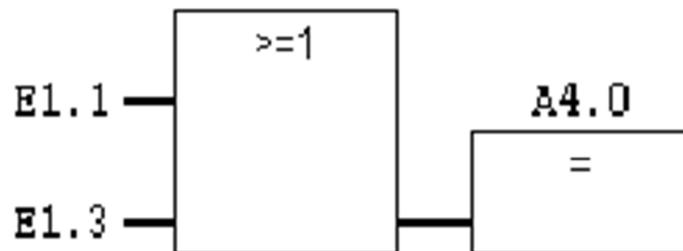


Figure 3.10. Exemple d'un « OU » logique avec le langage FBD

c. *Le LIST*

Le langage LIST figure parmi les langages de base du logiciel STEP7, sa syntaxe (figure3.11) est similaire à celle de l'assembleur.

C'est le langage le plus proche du langage machine MC7, des CPU S7, ce qui lui donne l'avantage d'être le langage le plus adapté pour la programmation avec optimisation d'espace mémoire et de temps d'exécution.

Il dispose d'un jeu d'instruction très important permettant la création de programmes utilisateur complets.

Tout programme écrit en CONT ou en LOG peut être réécrit en LIST.

➤ *Structure et éléments de LIST :*

Dans le langage LIST, l'éditeur de programme divise l'espace de programmation en deux parties ; partie programme et partie commentaire, ces deux parties sont séparées par *II*.

➤ *Structure d'une instruction :*

Une instruction LIST est constituée soit :

- En une opération seulement,
- En une opération + opérande.

L'opérande peut indiquer :

- une constante.
- un bit du mot d'état.
- une mnémonique.
- un bloc de données.
- un identificateur d'opérande d'une zone mémoire précise + une adresse.

```
      O      E      1.1
      O      E      1.3
      =      A      4.0
```

Figure 3.11. Exemple d'un « OU » logique avec le langage LIST

3.3 Le système IHM

3.3.1 Historique

En quelques dizaines d'années, l'interface homme machine a connu une évolution très importante. Ainsi, dans les années 1950, il fallait recourir à des tableaux de connexion, sur lesquels on enfichait des câbles reliant deux opérateurs, pour programmer des opérations mathématique sur les tabulatrices électromécaniques, lointains ancêtres de nos calculatrices programmables.

Dans les années 1960, les systèmes sont devenus capables d'interpréter une ligne de commande : le clavier s'est imposé, bientôt accompagné de l'écran. Dès cette époque, des modes d'interaction plus conviviaux avec les machines ont été recherchés en particulier au Xerox Palo Alto Research Center, le Parc. En 1964, Douglas C. Engelbart avait conçu les principes de l'interface graphique moderne : au lieu d'afficher des lignes de commandes les unes à la suite des autres, l'écran pouvait accueillir des fenêtres dans lesquelles s'affichaient des menus, auxquels on pouvait accéder en déplaçant un pointeur avec une souris à deux roues de métal...). Ecran, clavier, souris : les trois éléments fondamentaux de l'interface de nos ordinateurs personnels étaient réunis. Il a suffi de la visite au Parc en 1979 d'un jeune homme appelé Steve Jobs pour que l'invention se répande dans le commerce : le jeune patron d'Apple allait doter le premier Macintosh, lancé en 1984, d'une interface graphique et d'une souris. L'interface homme machine n'a guère bougé depuis, mises à part bien entendu des améliorations ergonomiques (couleurs, définition, etc.). Cela malgré les avancées des interfaces vocales.

3.3.2 Définition

Lorsque la complexité des processus augmente et que les machines et installations doivent répondre à des spécifications de fonctionnalité toujours plus sévères, l'opérateur a besoin d'un maximum de transparence. Cette transparence s'obtient au moyen de l'Interface Homme-Machine (IHM).

Un système IHM constitue l'interface entre l'homme (opérateur) et le processus (machine/installation). Le contrôle proprement dit du processus est assuré par le système d'automatisation. Il existe par conséquent une interface entre l'opérateur et WinCC

flexible (sur le pupitre opérateur) et une interface entre WinCC flexible et le système d'automatisation.

3.3.3 Les avantages de système IHM

Un système IHM se charge des tâches suivantes:

- ❖ Représentation du processus

Le processus est représenté sur le pupitre opérateur. Lorsqu'un état du processus évolue p. ex : 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 p. ex : 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, p. ex : lorsqu'une valeur limite est franchie.

- ❖ Archivage de valeurs processus et d'alarmes

Les alarmes et valeurs processus peuvent être archivées par le système IHM. Vous pouvez ainsi documenter la marche du processus et accéder ultérieurement aux données de la production écoulée.

- ❖ Documentation de valeurs processus et d'alarmes

Les alarmes et valeurs processus peuvent être éditées par le système IHM sous forme de journal. Vous pouvez ainsi consulter les données de production à la fin d'une équipe p. ex.

3.3.4 Logiciel de programmation WinCC flexible

a. Présentation du système WinCC flexible

- *Eléments de WinCC flexible :*

1- WinCC flexible Engineering System :

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

2- WinCC flexible Runtime :

WinCC flexible Runtime est le logiciel de visualisation de processus. Dans Runtime, vous exécutez le projet en mode processus.

b. Concepts d'automatisation avec WinCC flexible

WinCC flexible assure la configuration de divers concepts d'automatisation. Avec WinCC flexible, vous pouvez en particulier réaliser les concepts suivants de manière standard.

➤ **Système d'automatisation avec un pupitre opérateur**

On appelle système monoposte, un pupitre opérateur directement relié à un automate via le bus système (figure3.12).

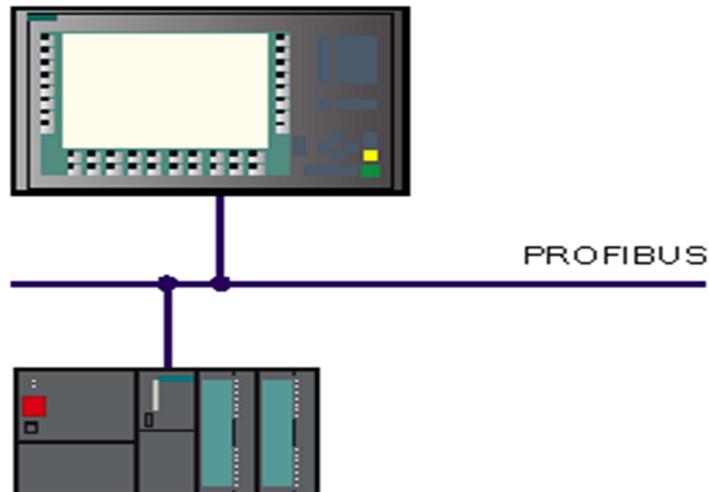


Figure 3.12. Système avec pupitre opérateur.

➤ **Système d'automatisation avec plusieurs pupitres opérateur**

Plusieurs pupitres opérateur sont reliés à un ou plusieurs systèmes d'automatisation via un bus système (figure3.13).

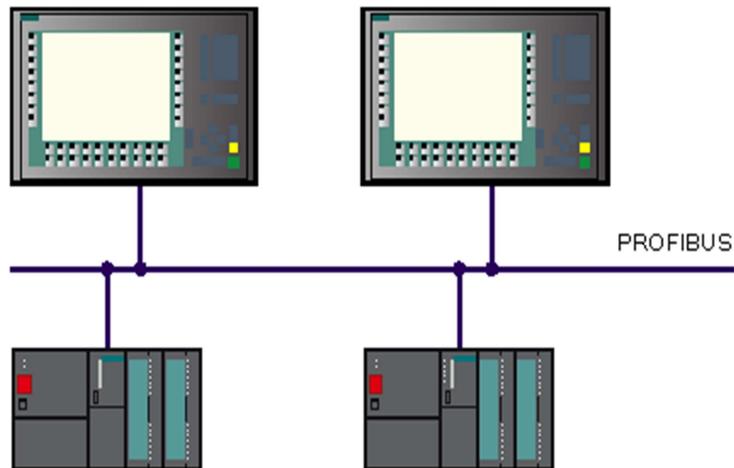


Figure 3.13. Système avec plusieurs pupitres opérateur.

De tels systèmes sont p. ex. mis en œuvre dans une chaîne de production pour permettre la commande de l'installation depuis plusieurs endroits.

c. Développement du projet

➤ **Projets sous WinCC flexible :**

Le logiciel de configuration WinCC flexible sert à configurer des interfaces utilisateur destinées à la commande de machines et d'installations. Il peut s'agir de simples panneaux de commande pour la lecture de paramètres mais aussi de pupitres opérateur complexes d'une chaîne de production p. ex.

WinCC flexible est un logiciel modulaire : Vous décidez, par le choix de l'édition WinCC flexible, des types de pupitre opérateur que vous serez à même de configurer.

L'édition WinCC flexible choisie pourra, si nécessaire, être mise à niveau sans problème.

Vous pouvez configurer 8 pupitres opérateur au plus dans un projet WinCC flexible.

Un projet WinCC flexible contient toutes vos données de configuration d'une installation ou d'un pupitre opérateur. Les données de configuration sont p. ex. :

- Des vues de processus qui représentent le processus.
- Des variables qui transmettent au runtime des données entre automate et pupitre opérateur.
- Des alarmes qui affichent au runtime des états de fonctionnement.
- Des archives qui sauvegardent des valeurs de processus et alarmes.

Toutes les données d'un projet sont enregistrées dans la base de données intégrée à WinCC flexible.

d. Utilisation de variables

➤ **Variables interne :**

Les variables internes ne possèdent aucun lien avec l'automate.

Les variables internes sont enregistrées dans la mémoire du pupitre. Le pupitre opérateur en question peut donc accéder en lecture et en écriture aux variables internes.

Les variables internes sont créées p. ex. pour exécuter des calculs locaux.

Vous pouvez utiliser pour les variables internes tous les types de données de base. Vous trouverez une liste détaillées des types de données sous "Types de données de base".

➤ **Variable externe :**

Les variables externes permettent de communiquer, c.-à-d. d'échanger des données entre les composants d'un process automatisé, entre un pupitre opérateur et un automate p. ex.

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.

Si vous configurez dans un environnement intégré avec STEP 7, vous pourrez créer des variables externes en accédant directement à toutes les variables qui ont été créées lors de la programmation de l'automate.

➤ **Editeur variable :**

Vous ouvrez l'éditeur Variables (figure3.14) en créant une nouvelle variable ou en éditant une variable affichée dans la fenêtre des objets.

Vous pouvez également ouvrir l'éditeur Variables en double-cliquant dans la fenêtre de projet sur l'entrée "Variables".

L'éditeur Variables affiche toutes les variables d'un dossier.

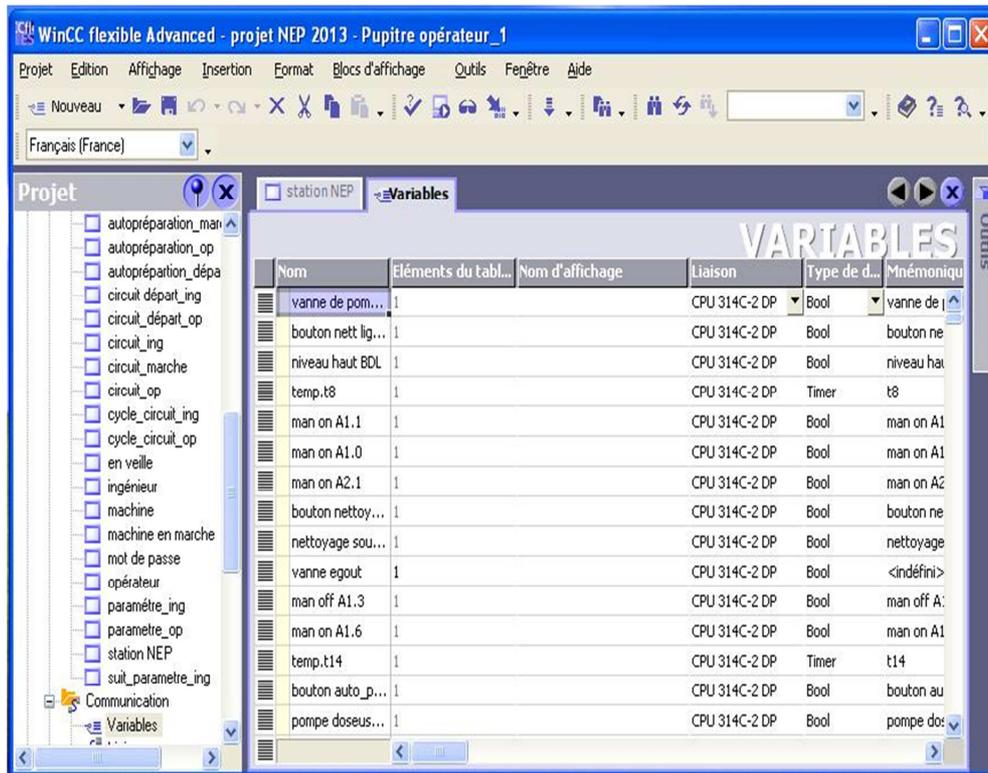


Figure 3.14. Fenêtre configuration variables.

e. Création des vues

Dans WinCC flexible, vous créez des vues pour le contrôle-commande de machines et d'installations. Pour créer des vues, vous disposez d'objets prédéfinis permettant de représenter votre installation, d'afficher des procédures et de définir des valeurs de processus (figure3.15).

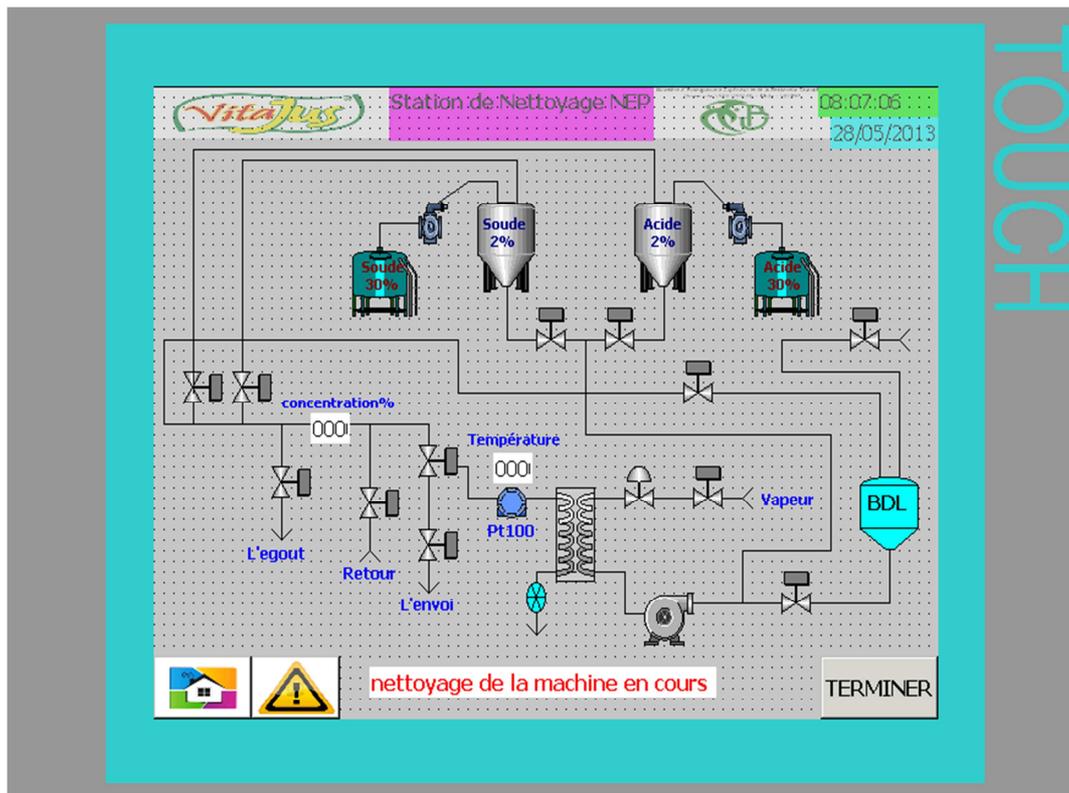


Figure 3.15. Création d'une vue.

f. Composition d'un système d'alarmes

➤ Affichage d'alarmes de process et d'alarmes système

1- Alarmes définies par l'utilisateur :

Vous configurez des alarmes pour afficher des états du processus ou pour saisir et lister, sur le pupitre opérateur, les données de process provenant de l'automate.

La procédure d'alarme détermine le type de l'information déclenchant une alarme et donc les propriétés de cette alarme.

WinCC flexible supporte les procédures d'alarme suivantes :

- Procédure par bit

Le pupitre opérateur déclenche une alarme lorsqu'un certain bit est mis à 1 dans l'automate. Pour cela, vous configurez des alarmes de bit dans WinCC flexible.

- Procédure analogique

Le pupitre opérateur déclenche une alarme lorsqu'une variable déterminée dépasse une valeur limite vers le haut ou vers le bas. Pour cela, vous configurez des alarmes analogiques dans WinCC flexible.

- Procédure par numéro d'alarme

L'automate transmet au pupitre opérateur un numéro d'alarme (accompagné éventuellement du texte d'alarme correspondant). Pour cela, vous pouvez configurer différentes alarmes dans le logiciel de configuration de l'automate :

dans SIMATIC STEP 7 : messages ALARM_S,

➤ **Alarmes système :**

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

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

Les alarmes système informent sur les états de fonctionnement du pupitre opérateur et des automates. Elles vont du simple renseignement à l'indication d'une erreur grave.

Le pupitre opérateur ou l'automate déclenche une alarme lorsqu'un état système déterminé ou une erreur survient dans l'un de ces appareils ou dans la communication entre eux.

Une alarme système se compose d'un numéro et du texte d'alarme. Le texte d'alarme peut aussi contenir des variables système internes qui précisent la cause du message d'erreur.

Vous ne pouvez configurer que certaines propriétés des alarmes système.

g. Intégration de WinCC flexible à Step7

➤ **Introduction :**

Si vous utilisez un automate SIMATIC et si vous avez installé le logiciel de configuration STEP 7 sur votre système, vous pouvez intégrer WinCC flexible dans STEP 7.

➤ **Avantages de l'intégration dans STEP 7 :**

Lors de la configuration intégrée, vous avez accès aux données de configuration que vous avez créées lors de la configuration de l'automate avec STEP 7. Vous profitez ce faisant des avantages suivants :

- Vous pouvez utiliser le gestionnaire SIMATIC Manager comme poste central de création, d'édition et de gestion des automates SIMATIC et des projets WinCC flexible.
- Les paramètres de communication de l'automate sont entrés par défaut lors de la

création du projet WinCC flexible. Toute modification sous STEP 7 se traduit par une mise à jour des paramètres de communication sous WinCC flexible.

- Lors de la configuration de variables et de pointeurs de zone, vous pouvez accéder sous WinCC flexible directement aux mnémoniques de STEP 7. Sélectionnez simplement sous WinCC flexible le mnémonique STEP 7 auquel vous voulez affecter une variable.
- Les modifications de mnémonique sous STEP 7 sont mises à jour sous WinCC flexible.
- Il vous suffit de définir les mnémoniques une seule fois sous STEP 7 pour pouvoir les utiliser sous ALARM_S et ALARM_D configurées sous STEP 7 sont prises en charge sous WinCC flexible et peuvent être affichées sur le pupitre opérateur.
- Vous pouvez créer un projet WinCC flexible sans intégration dans STEP 7 et intégrer ce projet ultérieurement dans STEP 7.
- Inversement, un projet intégré peut être désolidarisé de STEP 7 et être utilisé de façon autonome.
- Dans un multiprojet STEP 7, vous pouvez configurer des liaisons de communication sur plusieurs projets.

3.4 Conclusion

Les API et l'IHM ont fait l'objet de ce chapitre. Nous avons introduit l'API SIEMENS et le pupitre digital MP277 10 Touch, et leurs environnements de programmation à savoir : STEP7 et WinCC Flexible.

4.1 Matériels proposés (Hardware)

Après avoir analysé les besoins de la station NEP on a opté pour le matériel suivant:

4.1.1 CPU 314C-2DP (SIEMENS)

a. Présentation de la CPU

SIEMENS propose plusieurs gammes de produits qui peuvent être utilisés dans ce projet. On a utilisé un automate programmable appartenant à la série S7-300 (figure4.1) Sa caractéristique principale est l'intégration de modules comportant entre autres des fonctions intégrées.



Figure 4.1. la CPU 314C-2DP

La CPU S7-314C-2DP se compose des éléments suivant :

➤ **LED de visualisation d'état et de défaut :**

- (Rouge) SF Défaut de matériel ou de logiciel.
- (Rouge) BATF Défaillance de la pile.
- (Verte) 5V cc L'alimentation 5V cc est correcte.
- (Jaune) FRCE Le forçage permanent est actif.
- (Verte) RUN CPU en RUN.
- (Jaune) STOP CPU en STOP ou ATTENTE ou en démarrage.

➤ **Commutateur de mode de fonctionnement :**

- **RUN -P** : la CPU traite le programme utilisateur, et on peut le modifier.
- **RUN** : la CPU traite le programme utilisateur, et on ne peut pas le modifier.
- **STOP** : la CPU ne traite aucun programme utilisateur.
- **MRES** : Position instable du commutateur, pour effectuer l'effacement général il faut respecter un ordre particulier de commutation.

➤ **Pile de sauvegarde ou accumulateur :**

L'utilisation de l'accumulateur ou de la pile de sauvegarde est nécessaire pour l'horloge temps réel.

La pile de sauvegarde est aussi utilisée pour :

- la sauvegarde du programme utilisateur s'il n'est pas enregistré dans la mémoire morte.
- Pour étendre la zone rémanente de données.

b. Caractéristiques technique de la CPU

En résumé les caractéristiques de la CPU 314C-2DP dans les points suivants :

- Une mémoire de travail 96 Ko.
- Un module d'entrées/sorties digital (tout ou rien) intégrées (24 E/16S).
- Un module d'entrées/sorties analogique intégrées (5 E/2S).
- 4 voies pour comptage et mesure avec codeurs incrémentaux 24V (60kHz).
- fonction de positionnement intégrée.
- ports MPI+ DP (Maître ou esclave DP).
- configuration multi-rangées jusqu'à 31 modules; échange de données direct possible (émetteur et récepteur); équidistance; routage.

4.1.2 Les modules

Nous avons choisi suivant les exigences de l'installation les modules suivant :

- Module d'entrées/sorties analogique (8 E*RTD) spécial pour les thermistances et les sonde de température à résistance platine Pt100.
- Module d'entrées/sorties tout ou rien (8E/8S).

4.1.3 Le câble de communication MPI

L'interface MPI (interface multipoint) est l'interface de la CPU utilisée pour la console de programmation (PG), le pupitre opérateur (OP) ou pour la communication au sein d'un sous réseau MPI (figure4.2). La vitesse de transmission typique est de 187,5 k bauds.



Figure 4.2. Câble de communication MPI.

4.1.4 Le pupitre digital MP 277 10" Touch 1.1.4.0

Le Multi Panel MP 277 10" Touch (figure4.3) est le représentant de la catégorie de produits appelées "Plate-forme multifonctionnelle". Les pupitres opérateurs offrent des possibilités de communication avec le monde de la bureautique élargies.

Les appareils sont équipés des composants suivants :

- Interface PROFIBUS.
- Interface Ethernet pour la connexion à PROFINET.
- 2 interfaces USB.

- Interface MPI.
- Ecran TFT(LCD) couleurs jusqu'à 64k.



Figure 4.3. Pupitre digital MP 277 Touch 1.1.4.0.

4.1.5 Les capteurs

a. Les capteurs de proximité inductive

Les capteurs de proximité inductifs (figure4.4) détectent tous les matériaux conducteurs à une distance définie. Si un métal se trouve dans le champ de la zone de couverture active, la sortie PNP ou NPN du capteur est activée.

On a utilisé les capteurs de proximité inductifs pour contrôler l'ouverture et la fermeture des vannes (feedback ou réaction).



Figure 4.4. Capteur de proximité inductif.

b. Les capteurs de niveau spéciaux

Le VEGACAP 69 (figure4.5) est un détecteur de niveau à deux tiges totalement isolées destinées à la détection de niveau de liquides dans des cuves non conductrices électriquement ou des produits corrosifs comme l'soude, l'acide.



Figure 4.5. Détecteur de niveau VEGACAP 69.

4.2 Programmation et mise en service (Software)

4.2.1 Les GRAFCETS

Dans La première étape de la programmation, nous avons hiérarchisé le programme, clarifier le cahier des charges et faciliter la tâche de maintenance, à l'aide du langage graphique connu par les automaticiens appelé GRAFCET (GRAPHE Fonctionnel de Commande Etape Transition), ou diagramme de fonctionnement.

Ce langage est utilisé quel que soit la technique de réalisation (électrique, pneumatique ou hydraulique).

4.2.2 Programmation STEP7

a. Les équations séquentielles des étapes

M8.1 (vidange BDL),(figure4.6) est activé si l'étape M8.0(étape initiale du nettoyage circuit) est active et la réceptivité ($M40.4$ (bouton départ)* $M3.0$ (bit de sécurité)* $\overline{M6.0}$ (bit d'alarme)) est à '1'et se désactive quand M8.2(remplissage BDL par L'eau) est active, M8.1 l'ors de sont activation provoque l'actionnement des vannes ($A0.0$ (pompe de pression), $A1.0$ (vanne pompe pression), $A1.1$ (vanne après échangeur), $A2.0$ (vanne égout)).

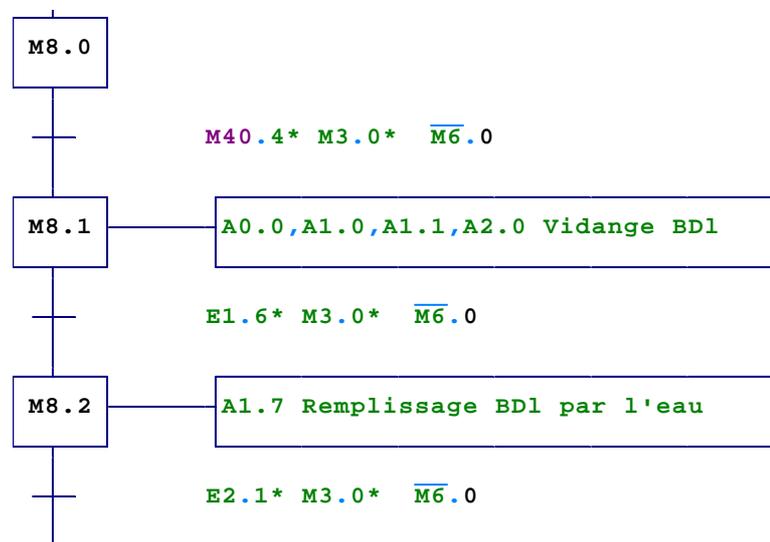


Figure 4.6. L'équation de l'étape M8.1.

$$M8.1 = M8.0 * (M40.4 * M3.0 * \overline{M6.0}) * \overline{M8.2}$$

M9.0 (vidange BDL), (figure4.7) est activé si l'étape M8.7(Rinçage avec l'eau chaude) est active et la réceptivité (T2(temps de rinçage)*M3.0(bit de sécurité)* $\overline{M6.0}$ (bit d'alarme)) est à '1' et se désactive quand M12.0(vidange complet) ou M9.1(vidange complet)est active, M9.0 (vidange BDL)l'ors de sont activation provoque l'actionnement des vannes (A0.0(pompe de pression, A1.0(vanne pompe pression), A2.0(vanne égout), A2.1(vanne d'envoi CIP), A2.2(vanne retour), A2.4(pompe retour CIP)).

Qui signifie le Vidange du BDL.

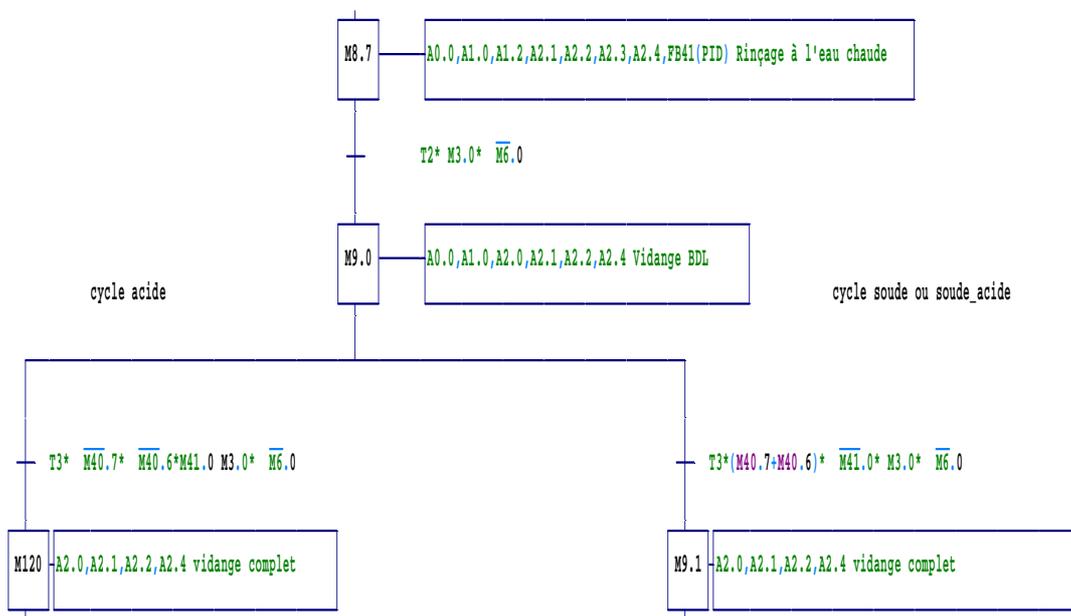


Figure 4.7. L'équation de l'étape M9.0.

$$M9.0 = M8.7 * (T2 * I1.0 * \overline{M6.0}) * (\overline{M12.0} + \overline{M9.1}) = M8.7 * (T2 * I1.0 * \overline{M6.0}) * (\overline{M12.0} * \overline{M9.1})$$

b. Conversion des équations vers le STEP7

La conversion de quelques équations en langage Ladder (figure4.8, figure4.9).

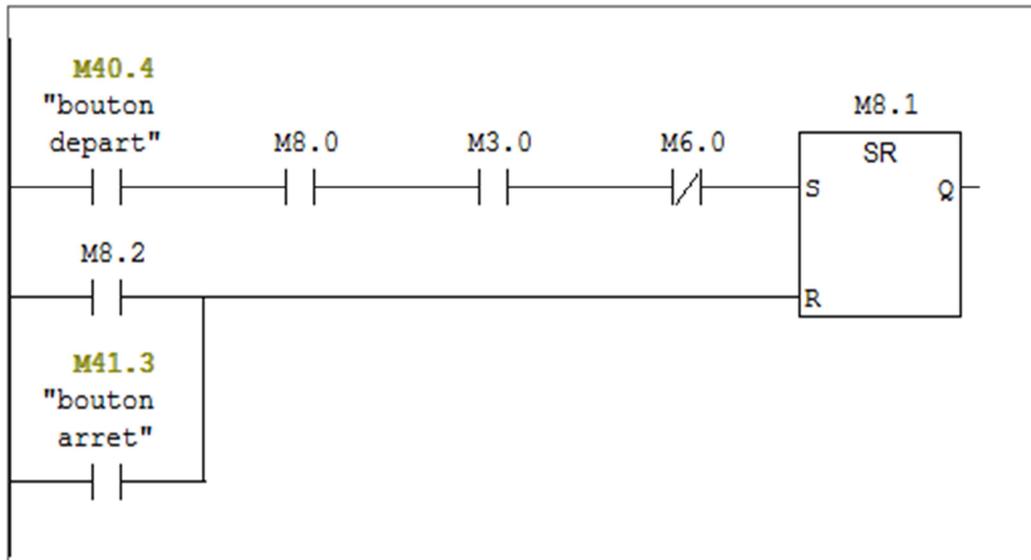


Figure 4.8. Conversion de l'étape M8.1 en langage Ladder (LD).

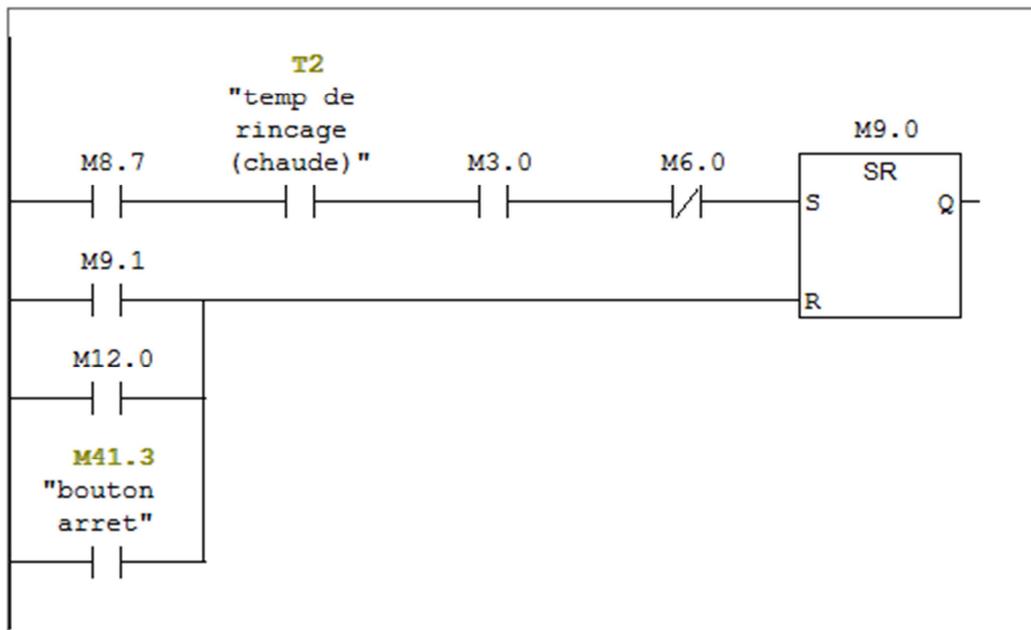


Figure 4.9. Conversion de l'étape M9.0 en langage Ladder (LD).

Remarque :

SR (Bascule mise à '0', mise à '1') propre au step7. Cette opération exécute la mise à '1' si l'état de signal est '1' à l'entrée S et '0' à l'entrée R. Si l'état de signal est '0' à l'entrée S et '1' à l'entrée R et si l'état de signal est à '1' à l'entrée S et R, la bascule est mise à '0'.

c. Hiérarchisation des blocs

➤ **Le bloc OB1 :**

Dans le bloc d'organisation OB1, se trouve :

- **Les bits mémoires de commande et de sécurité M1.0 (commande), M2.0 (commande) et M3.0 (sécurité).**

M1.0 : le bit mémoire est activé, si on appuie sur l'un des boutons existants dans la station de nettoyage (auto-préparation, nettoyage tank1/5 ou ligne verre/pack). La désactivation du bit est assurée si on appuie sur bouton d'arrêt ou le bit mémoire M2.0 est mis à 1 (figure4.10).

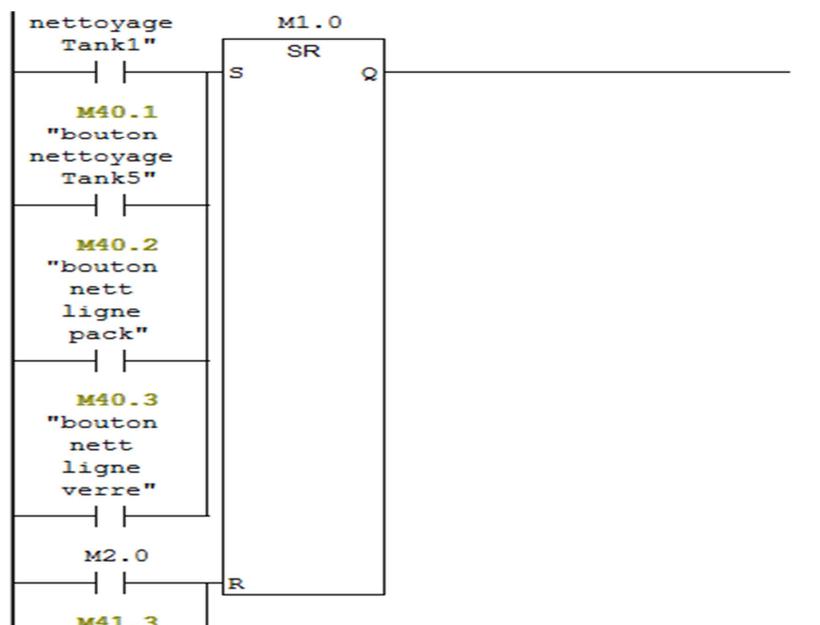


Figure 4.10. Le bit mémoire M1.0.

M2.0 : est un bit mémoire spécial pour l'auto-préparation, l'activation de ce bit est assurée si on appuie sur le bouton d'auto-préparation, et la désactivation est assurée si on appuie sur le bouton d'arrêt ou le bit mémoire M1.0 est mis à 1 (figure4.11).

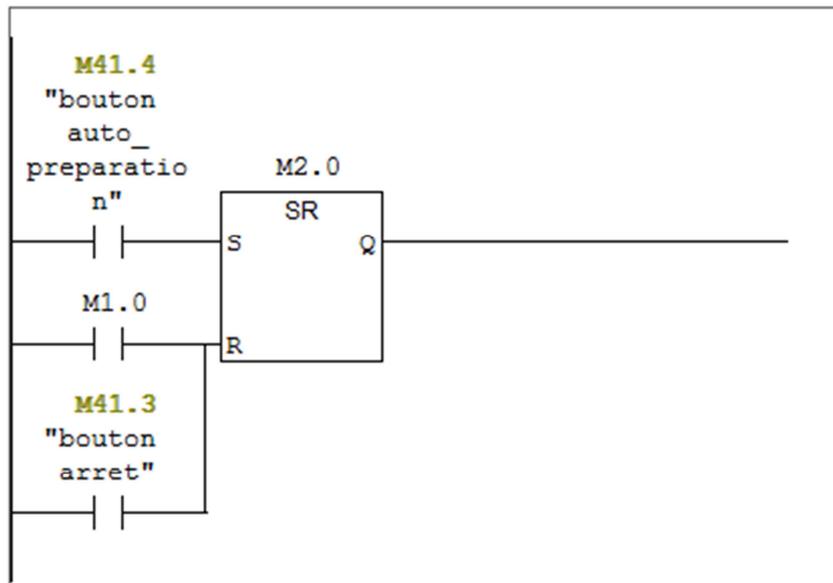


Figure 4.11. Le bit mémoire M2.0.

M3.0 : l'activation de ce bit est assurée si l'un des deux bits M2.0 et M1.0 sont mis à 1. La désactivation du bit est assurée si on appuie sur le bouton d'arrêt (figure4.12).

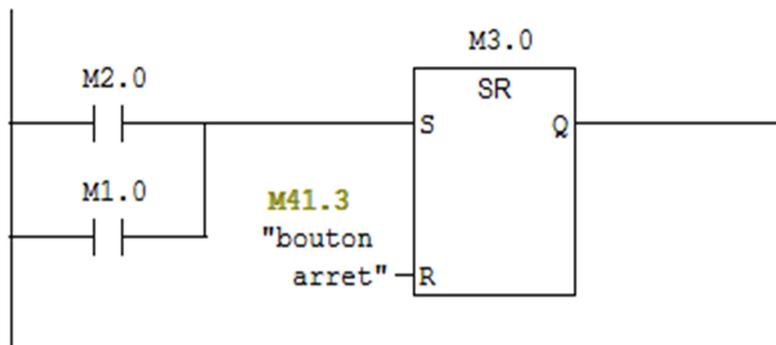


Figure 4.12. Le bit mémoire M3.0.

- **Les bits d'alarmes (système d'arrêt)**

M6.0 : ce bit mémoire de sécurité est mis à 1, s'il y a un défaut aux niveaux des pompes A0.0 (pompe de pression) et A2.4 (pompe de retour) ou aux niveaux des vannes (contrôle d'ouverture) ; dans ce cas la le système il s'arrê carrément (figure4.13).

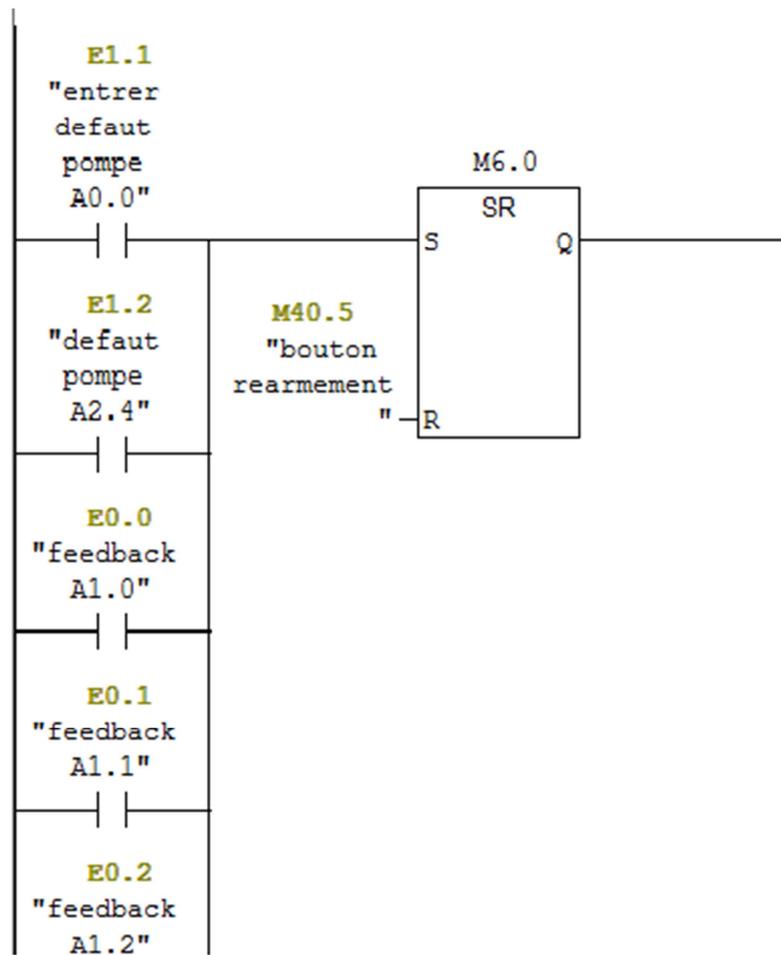


Figure 4.13. Le bit mémoire M6.0.

M3002.0 et **M3002.1** : ce sont des bits mémoires qui contrôlent les niveaux bas des tanks de soude et acide, si l'un des deux bits est activé une alarme s'affiche.

- L'appel des blocs de sous programme FB et FC
- **FB1** : ce bloc s'occupe de l'auto-préparation (figure4.14).

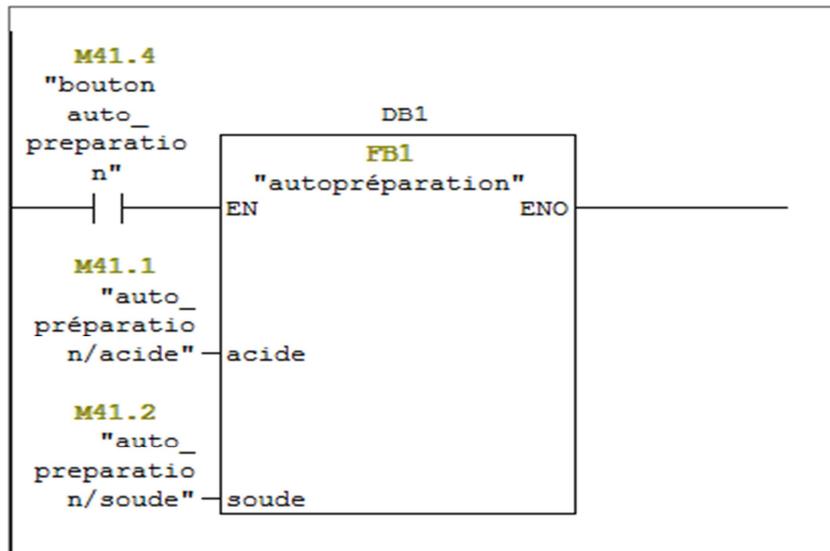


Figure 4.14. Le bloc fonctionnel FB1.

- **FB2** : ce bloc s'occupe du nettoyage des circuits tanks et lignes (figure4.15).

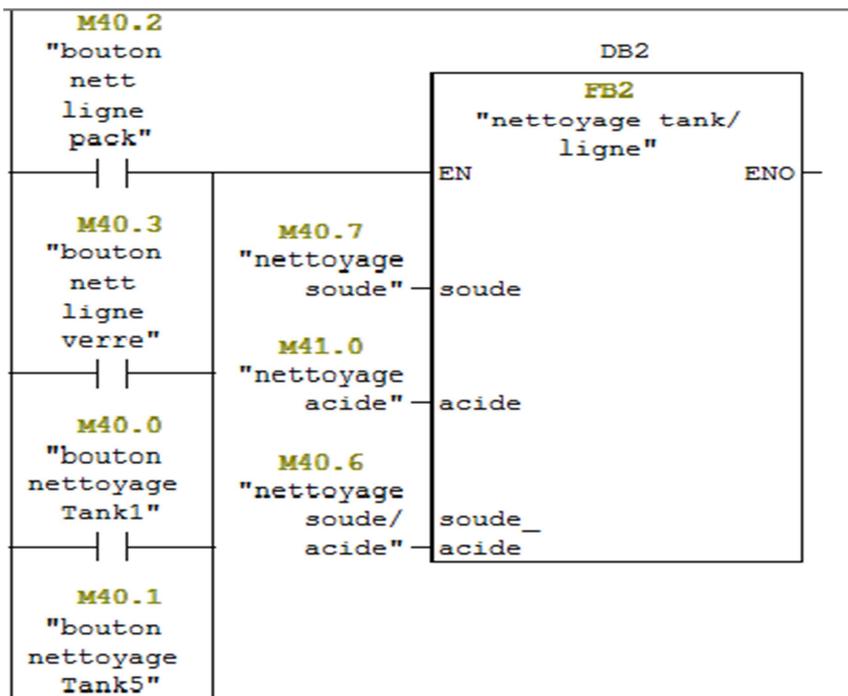


Figure 4.15. Le bloc fonctionnel FB2.

Description de l'opération **CALL_FB (Appeler FB)** :

Cette opération est exécutée si EN est à 1. Si l'opération CALL_FB a lieu, elle fonctionne comme suit :

- Elle sauvegarde l'adresse de retour au bloc appelant.
- Elle sauvegarde les sélecteurs pour les deux blocs de données en cours (DB et DI).
- Elle change la zone de données locales en cours en zone de données locales précédente.
- Ensuite, le traitement du programme se poursuit dans le bloc fonctionnel appelé.
- **FC1** : ce bloc s'occupe le nettoyage de circuit machine (figure4.16).

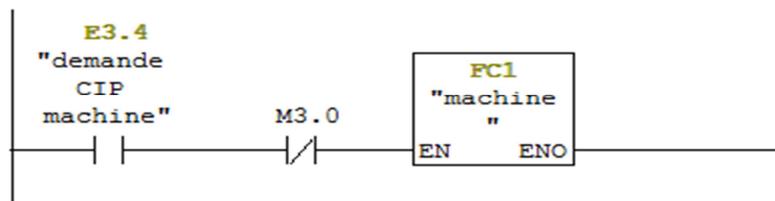


Figure 4.16. Le bloc fonctionnel FC1.

Description de l'opération **CALL_FC (Appeler FC)**

Cette opération permet d'appeler une fonction (FC). L'appel est activé par l'état de signal 1 à l'entrée de validation EN. Si l'opération a lieu, elle fonctionne comme suit :

- Elle sauvegarde l'adresse de retour au bloc appelant.
- Elle change la zone de données locales en cours en zone de données locales précédente.
- Elle crée la nouvelle zone de données locales pour la fonction appelée.

Ensuite, le programme poursuit le traitement dans la fonction appelée.

- **L'étape initiale (figure4.17)**

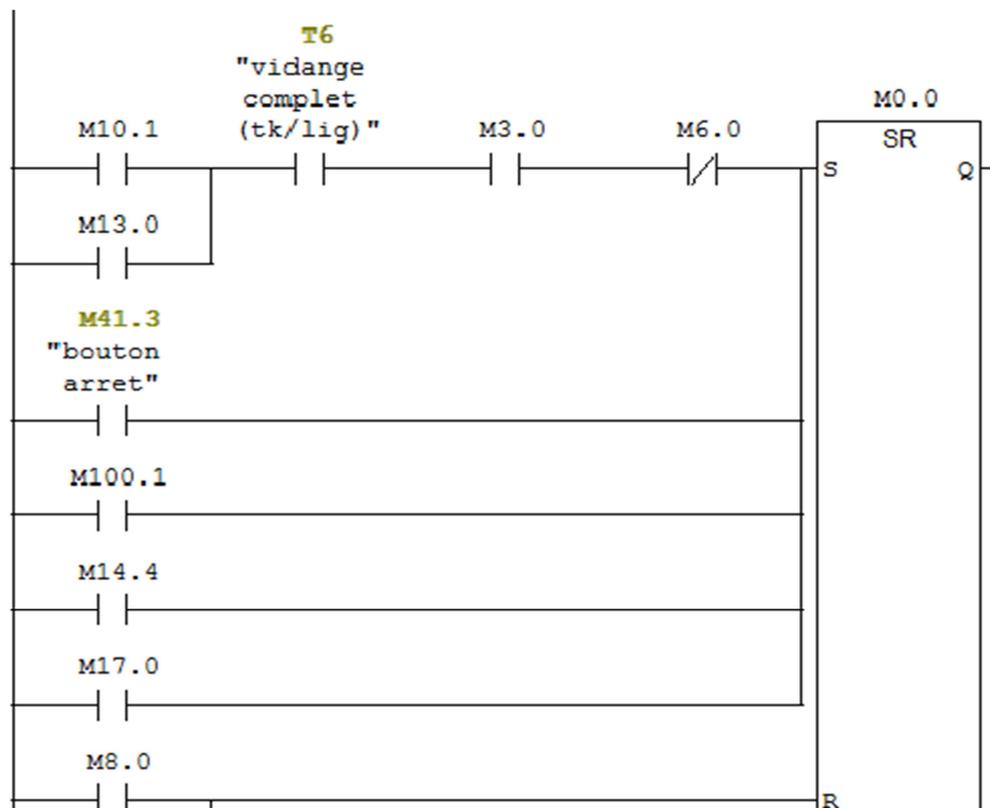


Figure 4.17. L'étape Initial M0.0.

Remarque :

Lorsque l'automate s'allume le bit M100.1 s'active et active l'étape initiale.

- **Les temps (Temporisateurs)**

Les temps (figure4.18) utilisés dans le programme sont :

T1 : temps de rinçage par l'eau froide pour tanks ou lignes.

T2 : temps de rinçage par l'eau chaude pour tanks ou lignes.

T3 : temps de vidange complet du BDL.

T6 : temps de vidange complet pour Tanks ou lignes.

T7 : temps de circulation de la soude pour tanks ou lignes.

T8 : temps de circulation de l'acide pour tanks ou lignes.

T10 : temps de dosage de la soude ou l'acide.

T11 : temps de mélange auto-préparation.

T12 : temps de rinçage après auto-préparation.

T13 : temps d'ajout si la concentration voulu n'est atteint.

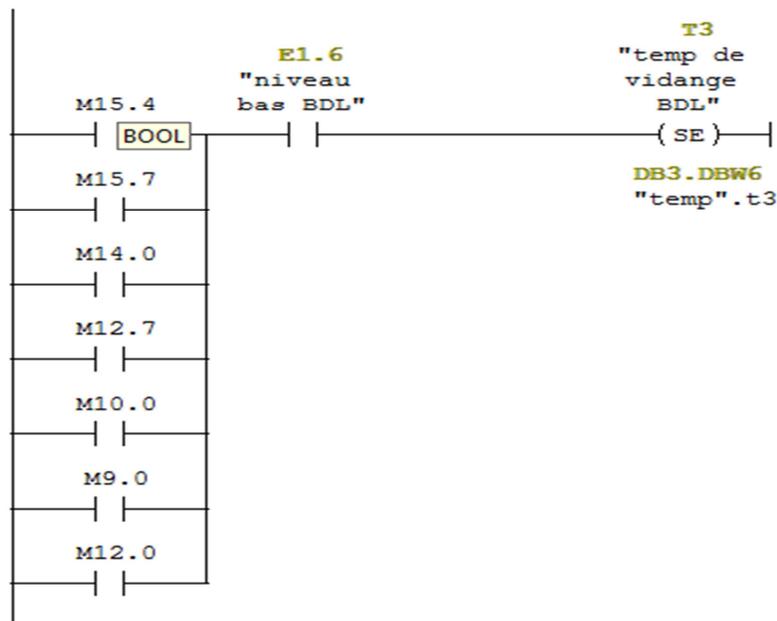


Figure 4.18. Temps de vidange complet du BDL (T3).

Description de l'opération ---**(SE) (Démarrer temporisation sous forme de retard à la montée)**

Cette opération démarre la temporisation indiquée avec la <valeur de temps> donnée si le RLG présente un front montant. L'interrogation à 1 de l'état de signal fournit un résultat égal à 1 lorsque la valeur de temps s'est écoulée sans erreur et que le RLG est toujours égal à 1. Si le RLG passe de 1 à 0 alors que la temporisation s'exécute, la temporisation est remise à zéro. Dans ce cas, l'interrogation à 1 de l'état de signal fournit un résultat égal à 0.

Remarque :

Les valeurs (variables) de temps de chaque circuit (temps vidange complet, temps circulation de l'acide, temps circulation de la soude, temps de rinçage par l'eau froide, temps de rinçage par l'eau chaude), seront transféré a l'aide de la fonction « **MOVE** » (figure4.19) lorsque l'opérateur ou l'ingénieur choisit le circuit.

Si l'opérateur choisi par exemple de nettoyer tank5 les valeurs (variables) de temps seront transférés par la fonction **MOVE** dans les fonctions « SE » (T1, T2, T6, T7, T8).

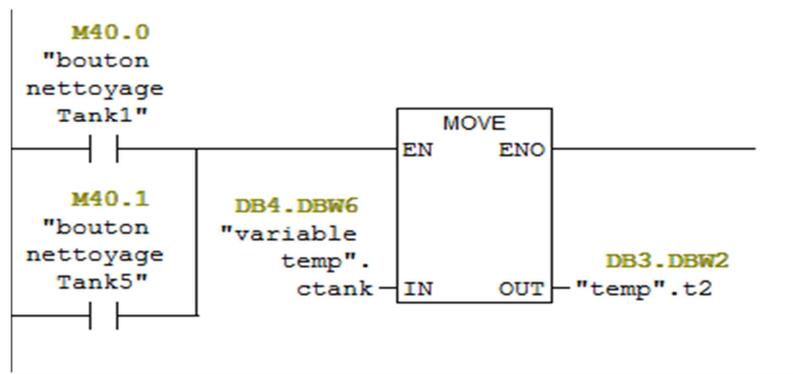


Figure 4.19. La fonction MOVE.

➤ **Le bloc FB1 :**

Dans le bloc fonctionnel FB1 (figure4.20) se trouve toutes les étapes d'auto-préparation (M14.0 ... M14.7).

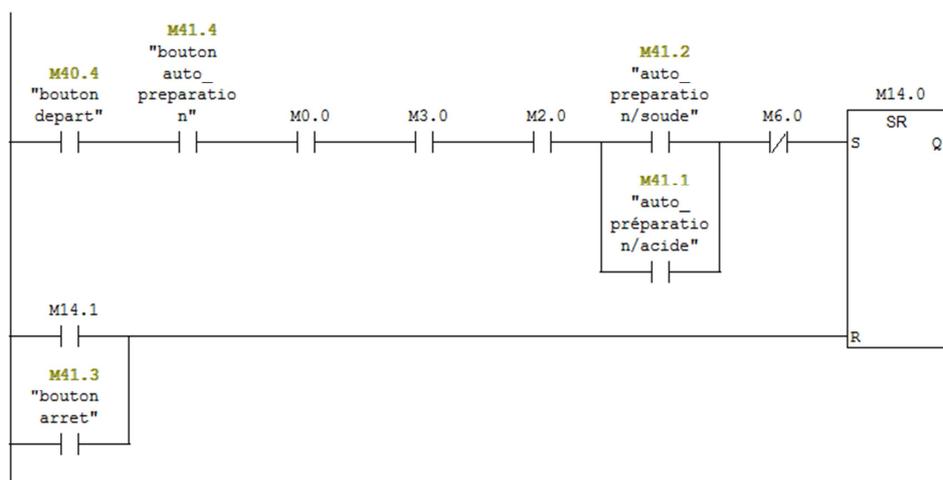


Figure 4.20. La première étape d'auto-préparation M14.0.

➤ **Le bloc FC1 :**

Dans le bloc fonctionnel FC1 (figure4.23) toutes les étapes de nettoyage machine.

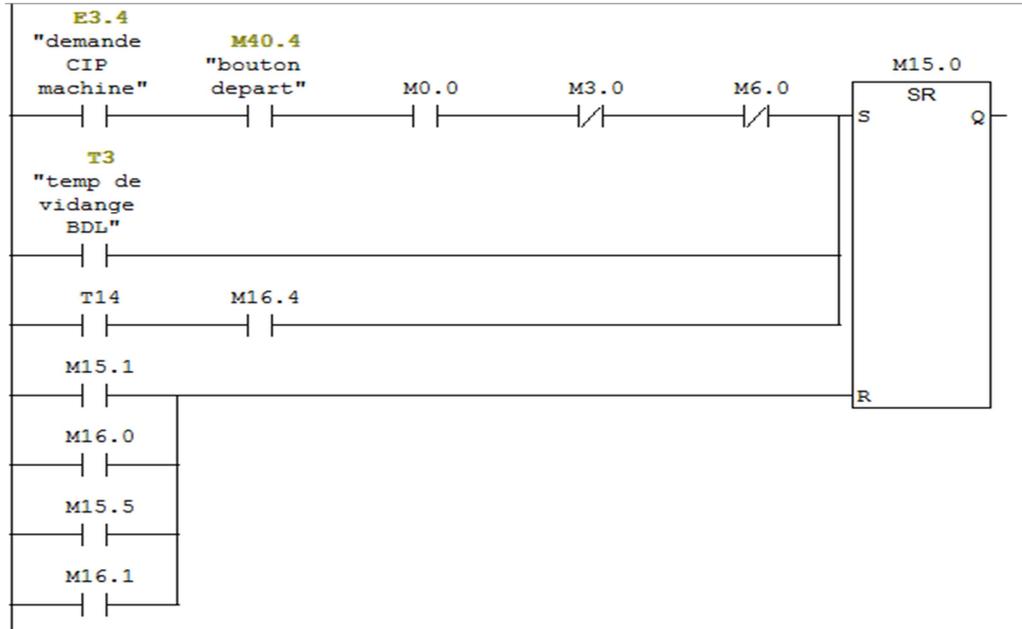


Figure 4.23. La première étape de nettoyage machine M15.0.

➤ **Le bloc FC2 :**

Dans le bloc fonctionnel FC2 (figure4.24) se trouve les actions (sorties) de toutes les étapes du programme.

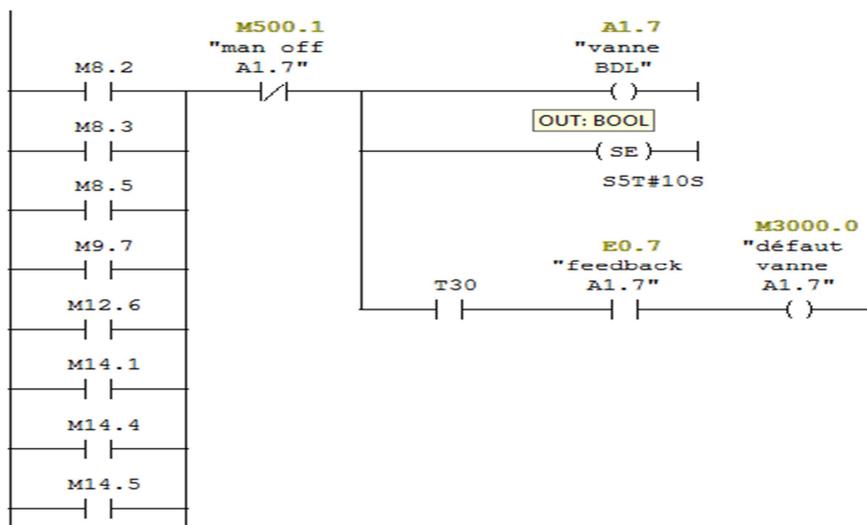


Figure 4.24. Les actions des étapes (sorties).

4.2.3 Programmation WinCC flexible

a. Architecture des vues

On a réparti les vues de façons étudié (figure4.25), pour que l'opérateur et l'ingénieur aient leurs propres privilèges.

Le schéma ci dessous représente la répartition des vues.

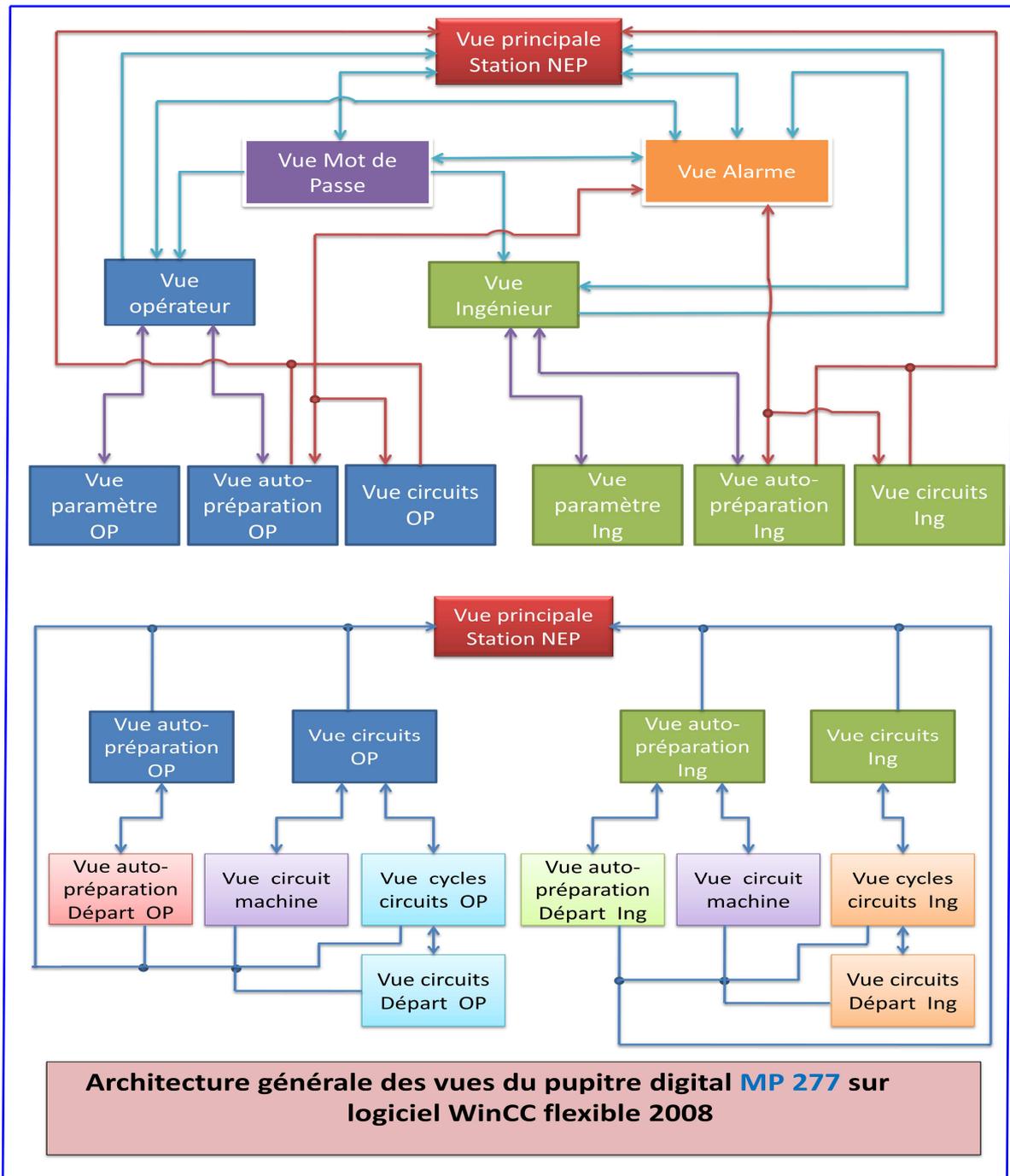


Figure 4.25. L'architecture général des Vues.

➤ **Vue principale Station NEP :**

Cette vue (figure4.26) est la représentation de la mise en page de l'interface utilisateur de la station.

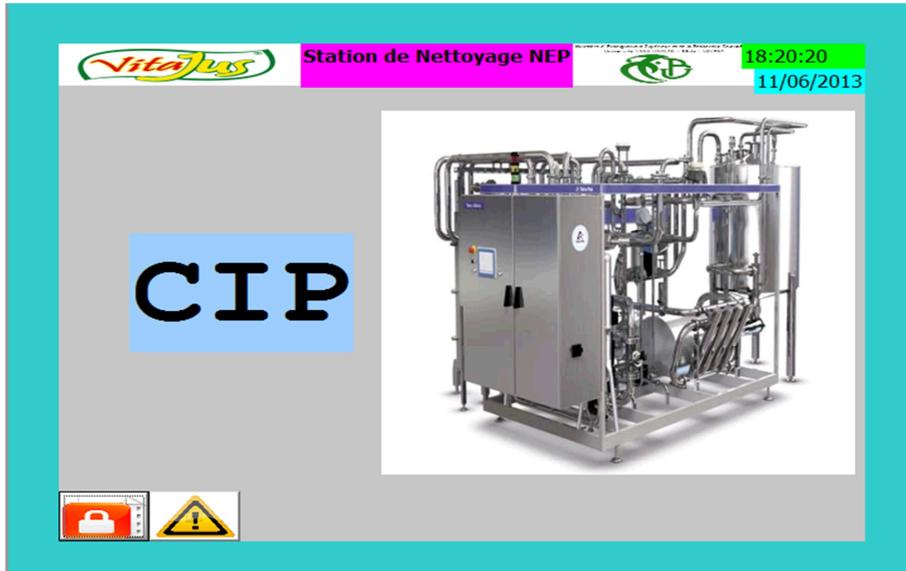


Figure 4.26. La vue principale station NEP.

➤ **Vue Alarme :**

La présente vue (figure4.27) a pour but de représenté les défauts et les mauvais fonctionnements des pompes et vannes.

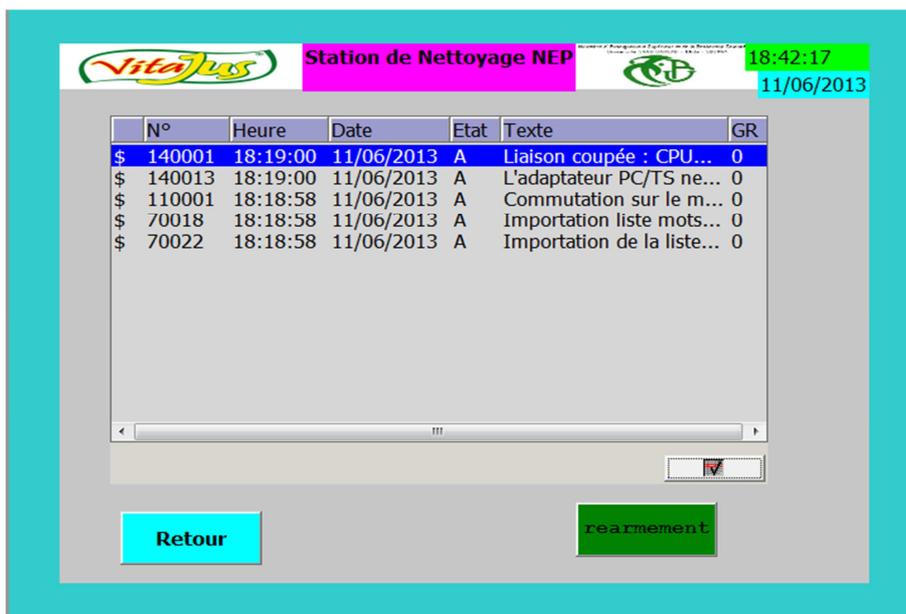


Figure 4.27. La vue des alarmes.

➤ **Vue mot de passe utilisateur :**

Les utilisateurs de l'installation effectuent avec cette vue (figure 4.28) le contrôle des mots de passe. Le privilège sera donné à chaque utilisateur ingénieur ou opérateur.



Figure 4.28. La vue des mots de passe.

➤ **Vue Opérateur (OP) :**

La vue ci-dessous (figure 4.29) c'est la transcription graphique de la station NEP avec le privilège opérateur, qui donne l'accès de forcé quelques vannes, paramétré la consigne et choisir le cycle.

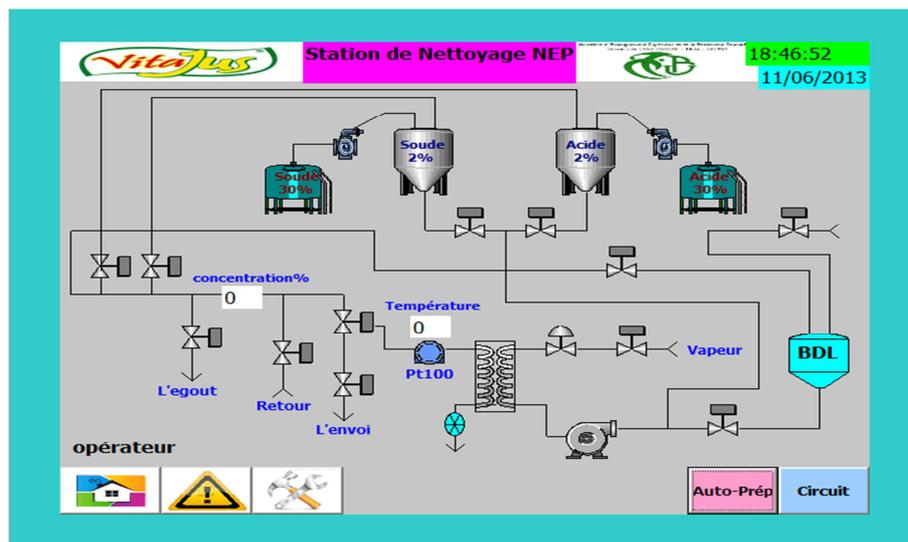


Figure 4.29. La vue de l'opérateur.

➤ **Vue Ingénieur (Ing) :**

Cette vue (figure4.30) est la représentation graphique de la station NEP avec privilège ingénieur, qui donne l'accès de forcé n'importe quelles vannes ou pompes et de paramétré les temps, la consigne et choisir le cycle.

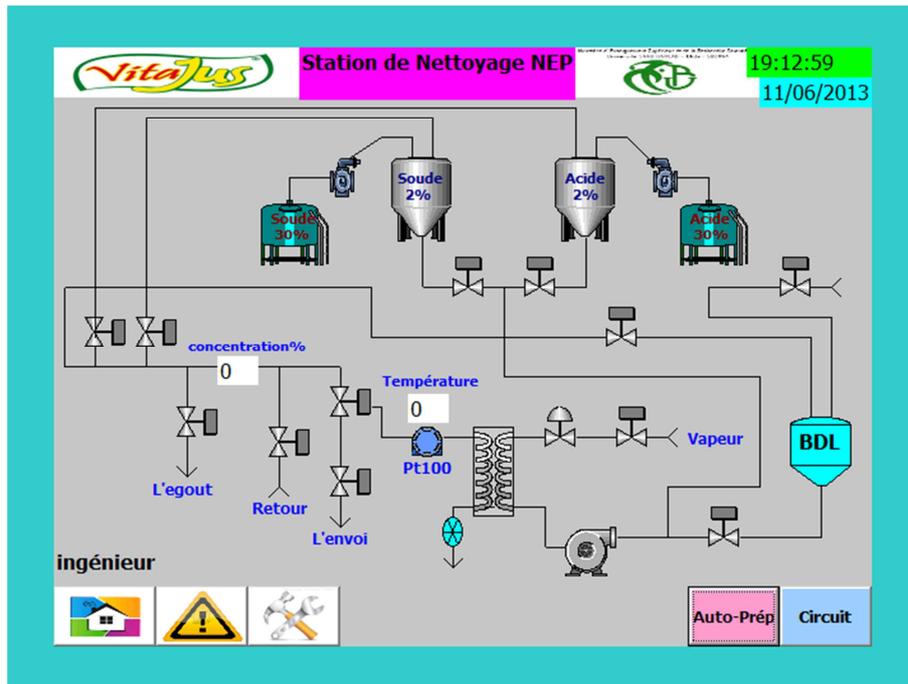


Figure 4.30. La vue de l'ingénieur.

Remarque :

La possibilité de forcer (vannes ou pompes) ou paramétré (consigne...) donne l'avantage de minimiser la communication physique avec PC à chaque fois.

➤ **Vue paramètre OP :**

Cette vue (figure4.31) donne la fonctionnalité de modifier la consigne pour l'opérateur.



Figure 4.31. La vue des paramètres pour opérateur.

➤ **Vue auto-préparation OP :**

Cette vue (figure4.32) nous donne le choix du cycle de l'auto-préparation (soude /acide) avec privilège opérateur.

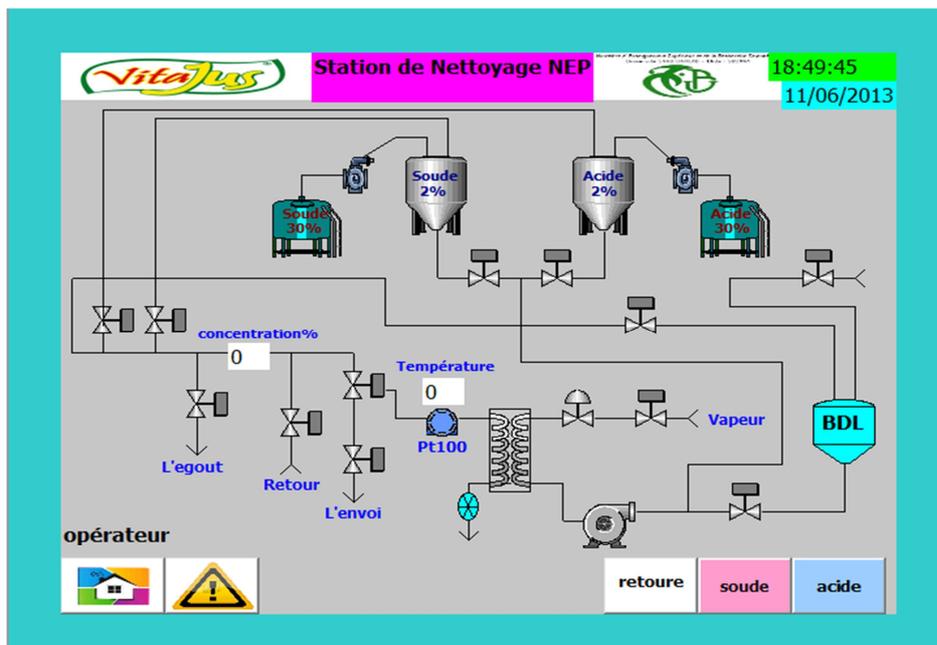


Figure 4.32. La vue d'auto-préparation pour opérateur.

➤ **Vue circuit OP :**

Cette vue (figure4.33) nous donne le choix du circuit à nettoyer avec prérogative opérateur.

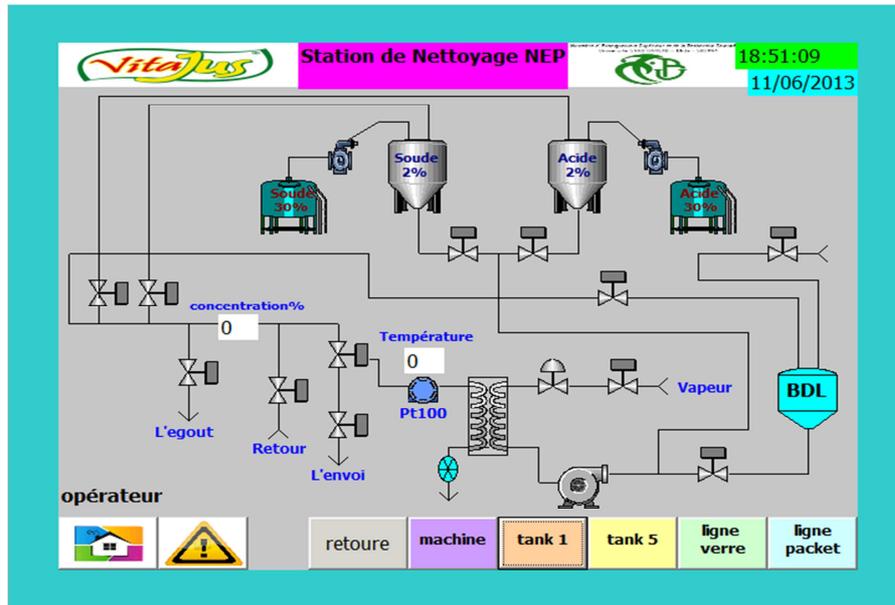


Figure 4.33. La vue des circuits pour opérateur.

➤ **Vue paramètre Ing :**

Cette vue (figure4.34) donne les prérogatives ingénieur pour modifié les paramètres du programme (temps, consigne...etc.).

Paramètre	Valeur	Unité
temp du vidange d'BDL	#####	seconde
temp de la circulation de la soude	#####	seconde
temp de la circulation de l'acide	#####	seconde
temp du remplissage de la soude ou l'acide (autopré)	#####	seconde
temp de mélange (autopré)	#####	seconde
temp de rinçage apret l'autoprépartion	#####	seconde
temp d'justement de la conductivité (autopré)	#####	seconde
temp de vidange autopréparation pour ajustement de la conductivité	#####	seconde
temp de rinçage des lignes avec l'eau froide	#####	seconde
temp de rinçage des tanks avec leau froide	#####	seconde

si vous avez modifier taper sur entrer du clavier puis enrgistrer
sinon appuyez direct sur enrgistrer

Figure 4.34. La vue des paramètres pour l'ingénieur.

➤ **Vue auto-préparation Ing :**

Cette vue (figure4.35) nous donne le choix du cycle de l'auto-préparation (soude /acide) pour l'ingénieur avec l'avantage de forcé n'importe quelles vannes ou pompes.

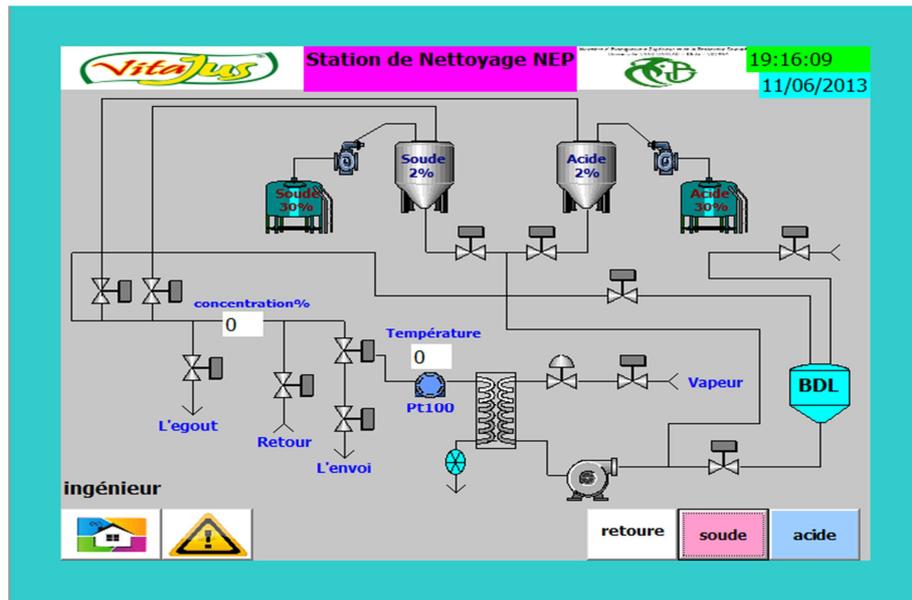


Figure 4.35. La vue d'auto-préparation pour l'ingénieur.

➤ **Vue circuit Ing :**

Cette vue (figure4.36) nous donne le choix du circuit a nettoyé avec prérogative ingénieur de forcer n'importe quelles vannes ou pompes.

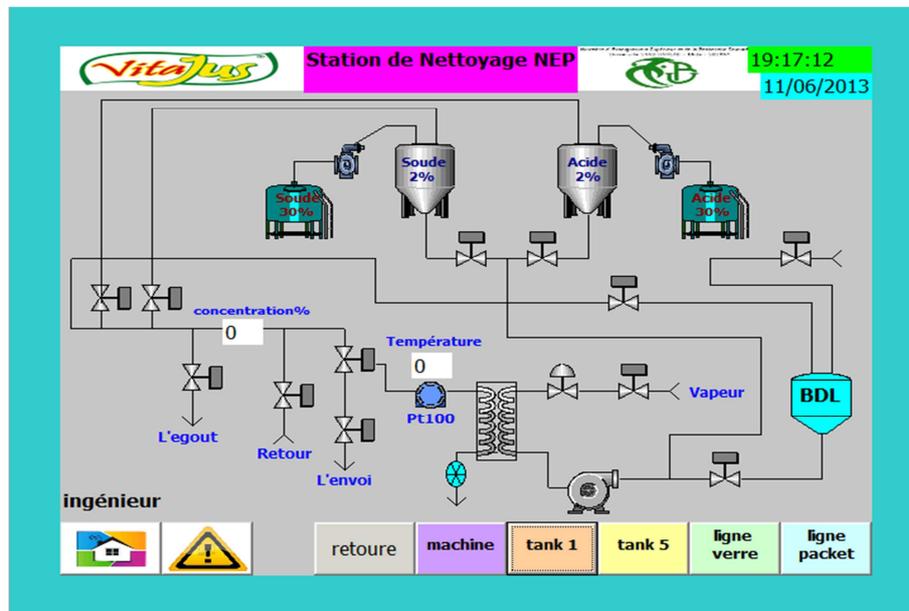


Figure 4.36. La vue des circuits pour ingénieur.

➤ **Vue auto-préparation Départ OP :**

Cette vue (figure4.37) c'est la dernière étape pour activer le cycle auto-préparation soude ou acide en appuyant sur le bouton départ.

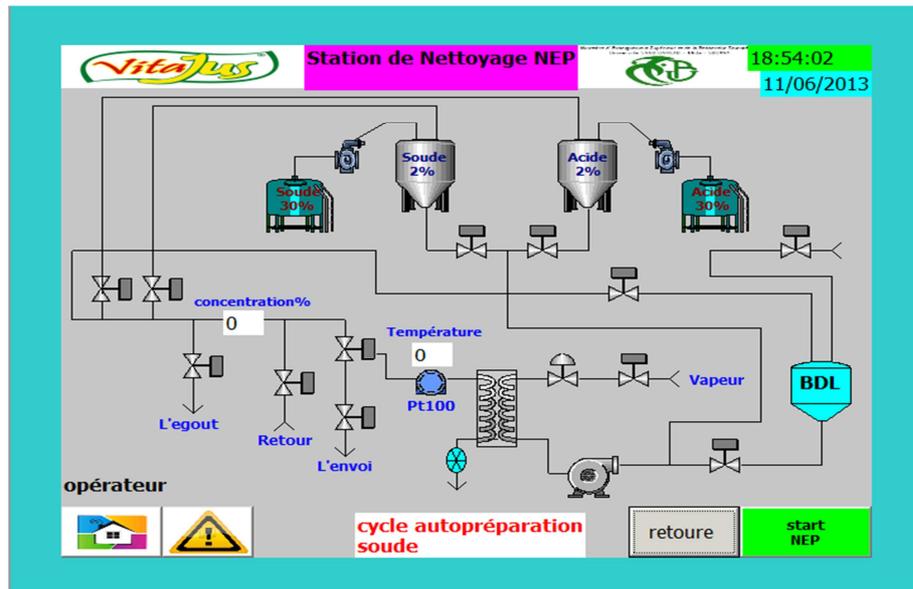


Figure 4.37. La vue d'auto-préparation départ pour opérateur.

➤ **Vue cycles circuit OP :**

Cette vue (figure4.38) a pour but de choisir le cycle du circuit a nettoyé avec privilège opérateur.

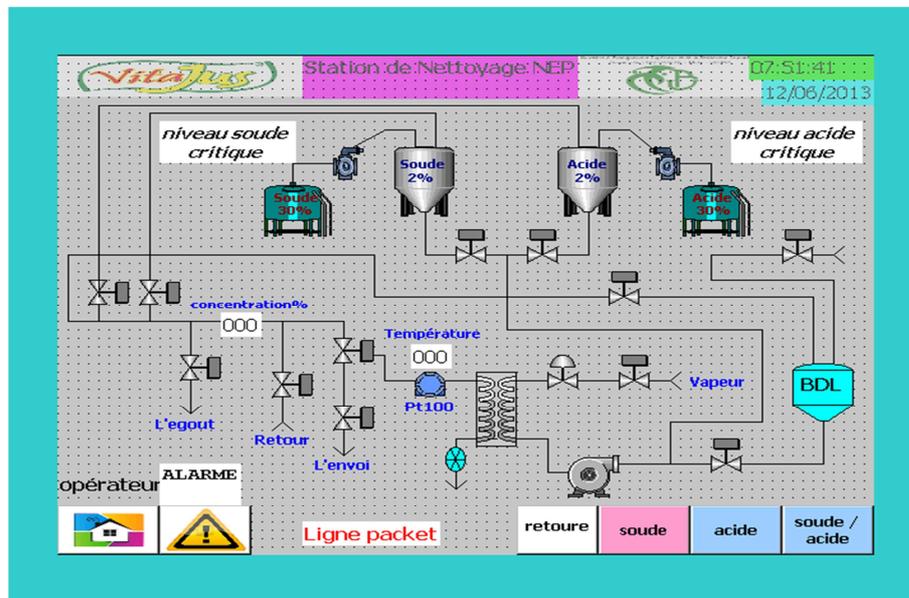


Figure 4.38. La vue cycle circuit pour opérateur.

➤ **Vue circuit Départ OP :**

Cette vue (figure4.39) c'est la dernière étape pour activé le nettoyage d'un circuit avec privilège opérateur.

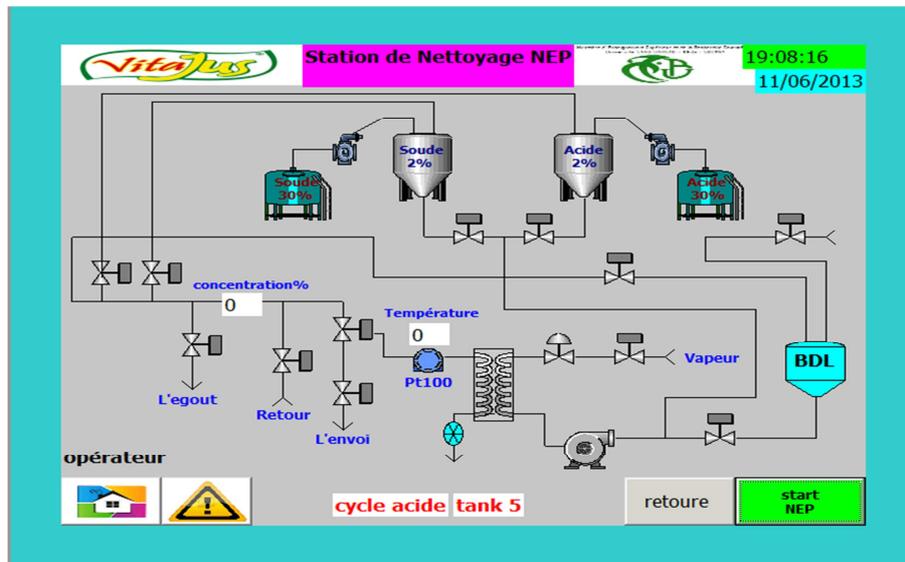


Figure 4.39. La vue circuit Départ pour opérateur.

➤ **Vue machine :**

La vue ci-dessous (figure4.40) est commune entre l'opérateur et l'ingénieur, elle permet de donner l'accès a la machine de se nettoyé elle-même en demandons les solutions (soude, acide), l'eau chaude, l'eau froide.

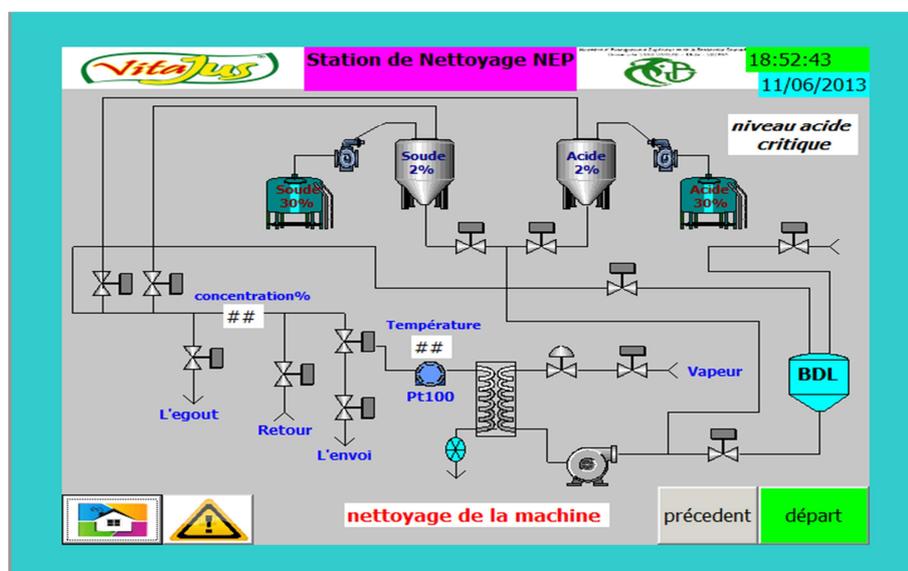


Figure 4.40. La vue machine.

➤ **Vue auto-préparation Départ Ing :**

Cette vue (figure4.41) c'est la dernière étape pour activer le cycle auto-préparation soude ou acide en appuyant sur le bouton départ.

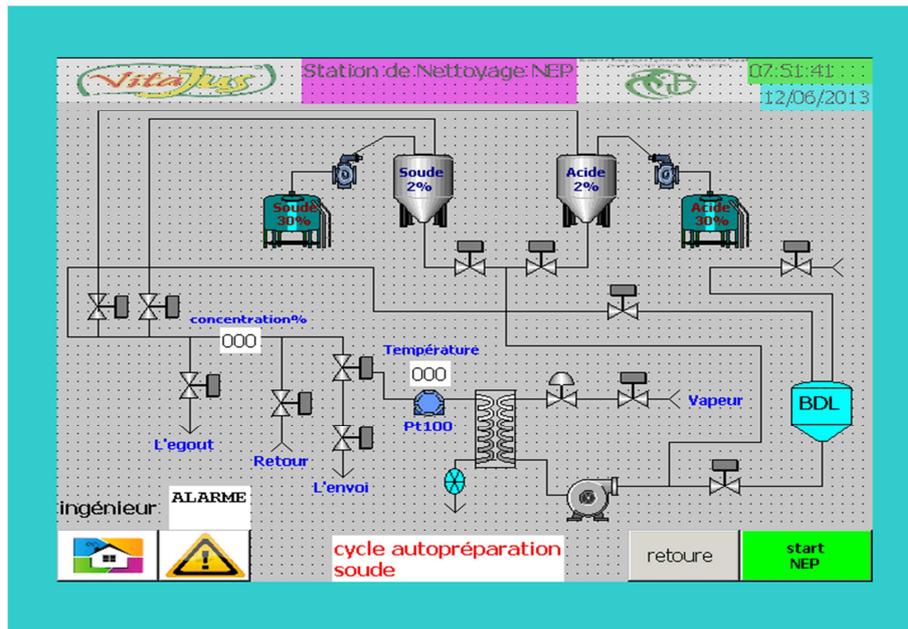


Figure 4.41. La vue d'auto-préparation ingénieur.

➤ **Vue cycles circuit Ing :**

Cette vue (figure4.42) a pour but de donné l'accès de choisir le cycle du circuit a nettoyer avec privilège ingénieur et l'accès de forcé n'importe quelles vannes ou pompes.

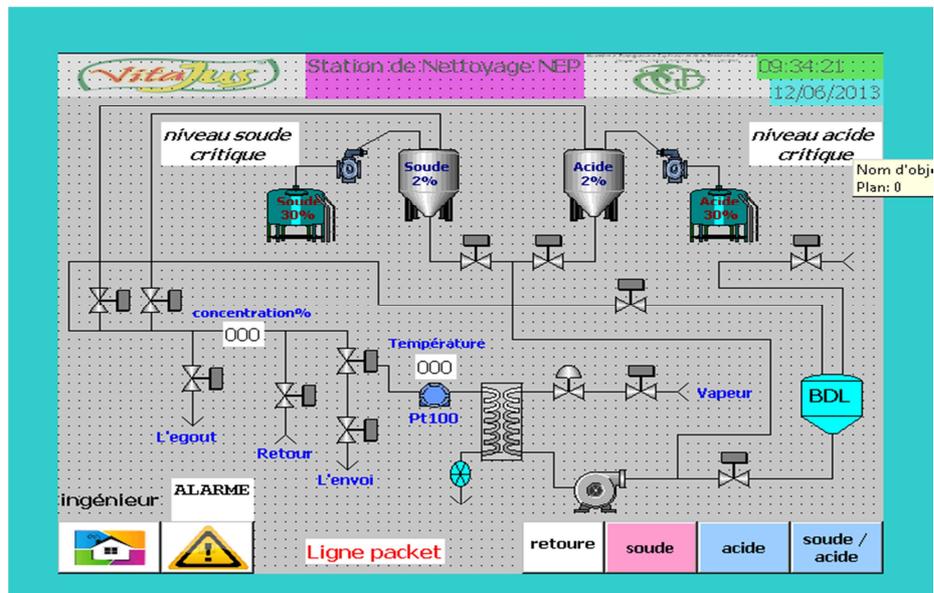


Figure 4.42. La vue des cycles circuit ingénieur.

➤ **Vue circuit Départ Ing :**

Cette vue (figure4.43) c'est la dernière étape pour activé le nettoyage d'un circuit avec privilège ingénieur avec possibilité de forcé n'importe quelles vannes ou pompes.

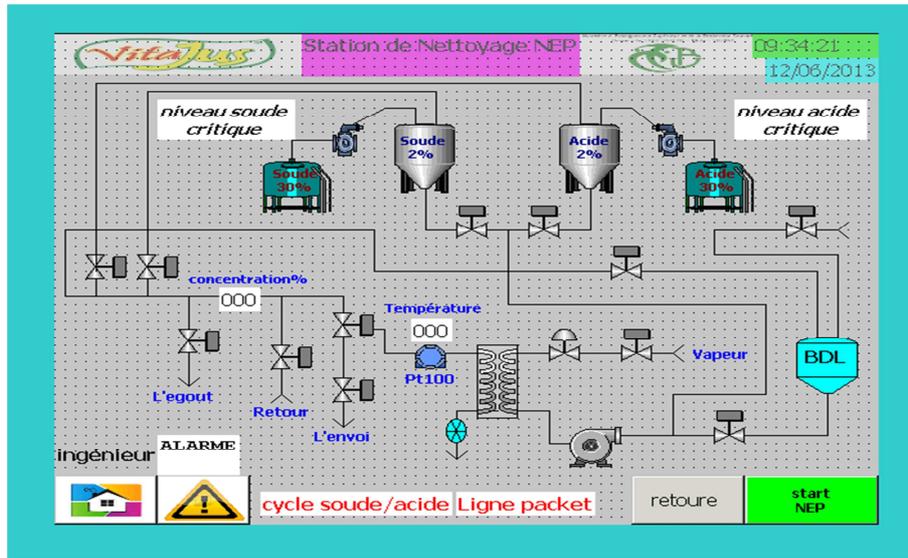


Figure 4.43. La vue circuit départ ingénieur.

➤ **Vue circuit marche**

C'est la vue qui affiche le déroulement de nettoyage du circuit choisi et elle est commune entre l'ingénieur et l'opérateur, le forçage des vannes et pompes n'est plus accessible (figure4.44).

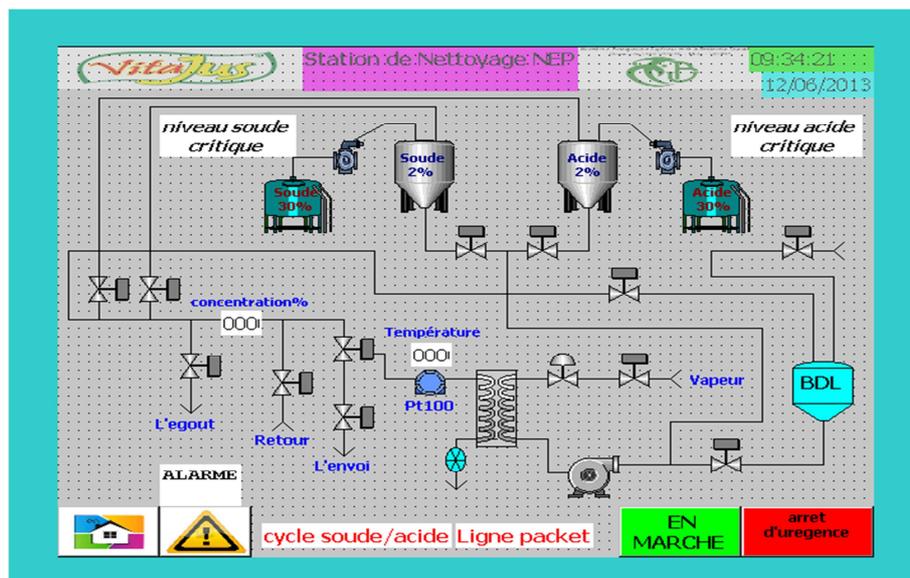


Figure 4.44. La vue circuit marche.

➤ **Vue manuel/automatique**

Cette vue (figure4.45) n'est pas d'écrit dans l'architecture, elle a pour but de donner la possibilité de forcer une vanne ou une pompe à « 0 » (fermé) on appuyant sur « man/off », « 1 » (ouverte) on appuyant sur « man/on » ou laisser le fonctionnement automatique d'apprêt les étapes du grafcet en appuyant sur « Auto ».

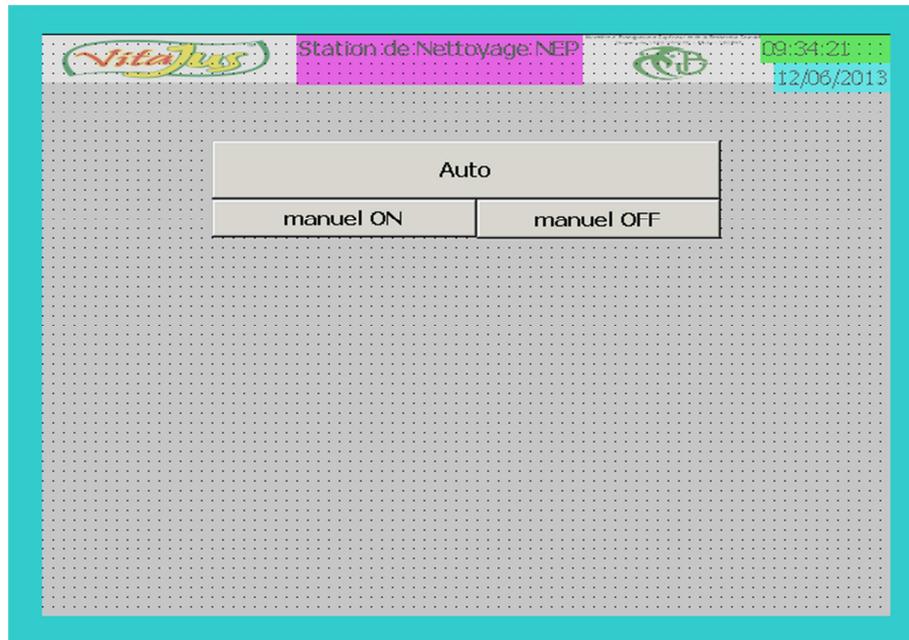


Figure 4.45. La vue man/Auto.

b. Communication

➤ Variables

Une variable sert à échanger des valeurs de process entre le pupitre et l'automate, la figure ci-dessous (figure4.46) montre la flexibilité de winCC pour intégrer les variables et leurs mnémoniques dans le projet winCC(figure4.46).

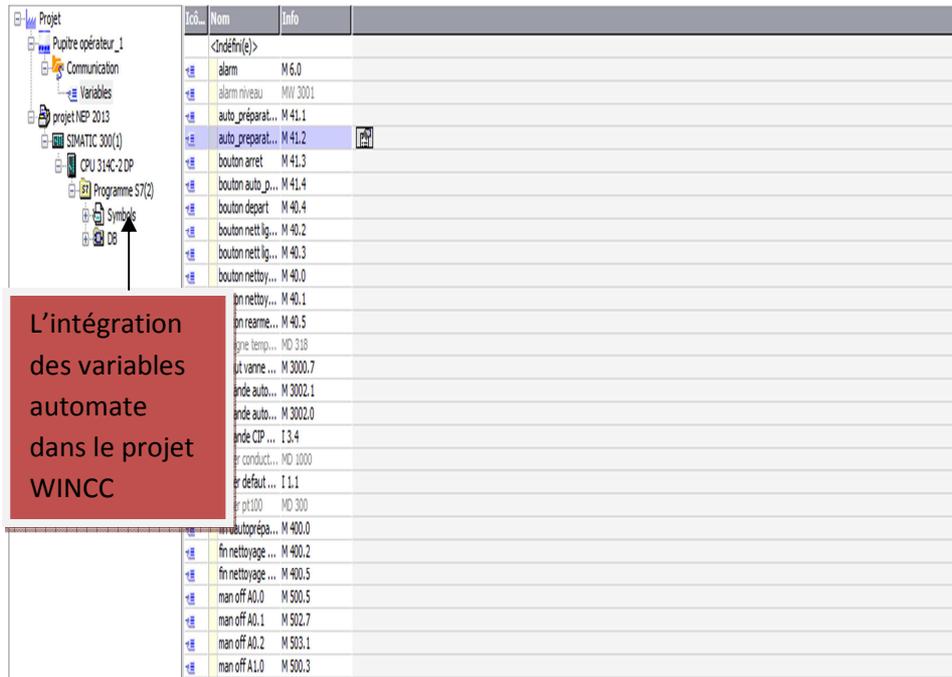


Figure 4.46. L'intégration des variables automate dans le projet winCC.

➤ Liaison

L'échange de données entre deux partenaires de communication est considéré comme une Communication. Les partenaires de communication peuvent être reliés via une liaison directe ou via un réseau, la figure ci-dessous montre la liaison entre l'automate et le pupitre (figure4.47).

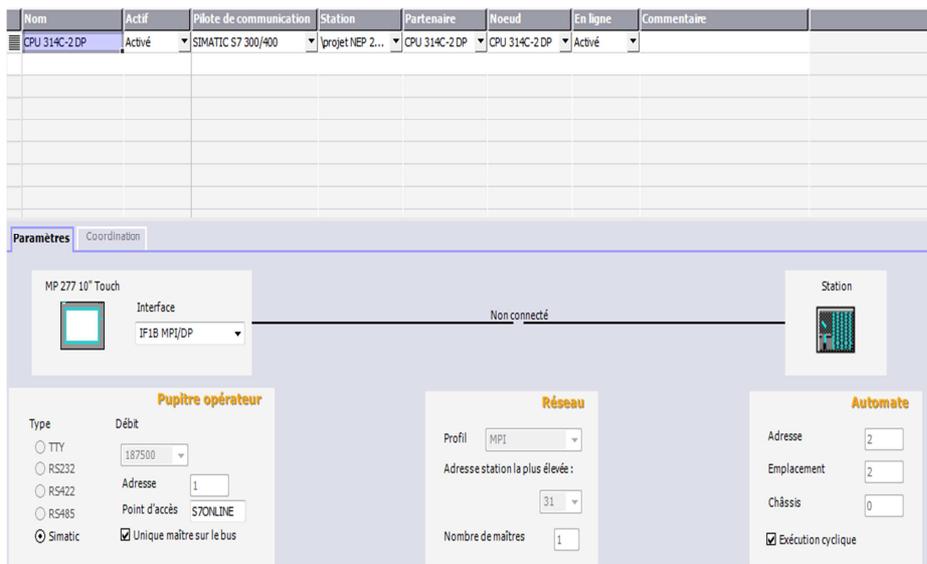


Figure 4.47. Liaison entre l'API et le pupitre

c. Paramétrage du pupitre

La figure ci-dessous (figure4.48) montre le paramétrage du pupitre (la configuration matérielle).

PARAMETRAGE DU PUPITRE

Général

Pupitre opérateur

Nom: Pupitre opérateur 1

Vue initiale: station NEP

Type de pupitre: MP 277 10" Touch 1.1.4.0

Résolution d'écran: 640x480

Auteur: []

Commentaire: []

Paramètres Runtime

Utiliser le clavier virtuel

Verrouiller la commutation de tâche

Afficher les info-bulles sur les valeurs limites

Afficher les commentaires de scripts

Sélection de bit dans les listes de texte et de graphique

Sélection de bit pour la représentation

Transférer les noms

ID projet: 0

Services Runtime

Sm@rtAccess ou Service : Démarrage Sm@rtServer

Sm@rtService : Pages HTML

Sm@rtAccess : Service Web (SOAP)

Sm@rtAccess : Serveur SIMATIC HMI HTTP

Servir de serveur OPC

Paramètres SMTP

Nom du serveur: []

Nom de l'émetteur: SIMATIC HMI Device

Adresse e-mail: []

Connexion: []

Figure 4.48. Paramétrage du pupitre.

4.3 Etude de régulation

Le travail qu'on a traité, on a été obligé de remplacer le régulateur PID externe (OMRON) par un autre interne, qui est intégré dans le logiciel STEP7-Manager. Dans ce cas il faut déterminer les paramètres de régulation K_p , K_i (T_i) et K_d (T_d), en utilisant les méthodes de synthèse dites empiriques.

4.3.1 les méthodes empiriques utilisées

a. Méthode de Ziegler et Nichols

Ziegler et Nichols ont proposé deux approches expérimentales destinées à fixer rapidement les paramètres des régulateurs P, PI et PID. La première nécessite l'enregistrement de la réponse indicielle du système à régler, alors que la deuxième exige d'amener le système en boucle fermée à sa limite de stabilité.

➤ Méthode de Ziegler et Nichols en boucle ouverte :

- Mode opératoire (figure4.49)

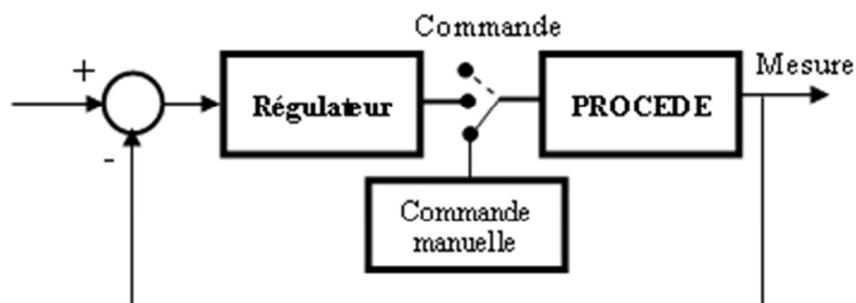


Figure 4.49. Schéma du système.

- Le régulateur est en mode automatique et la boucle est dans état stabilisé. La sortie du régulateur indique une commande u_0 et la sortie du procédé indique une valeur y_0 .
- On affiche la valeur u_0 sur le module de la commande manuelle.
- On met le régulateur en mode manuel c'est-à-dire qu'il est déconnecté de la boucle.
- On envoie une variation constante du signal de commande sur l'entrée procédé et on enregistre sur une table traçante la variation du signal de mesure à la sortie du procédé.

Il s'agit donc de l'enregistrement de la réponse indicielle du procédé seul.

- Exploitation de la réponse indicielle

Sur l'enregistrement de la réponse indicielle (figure 4.50), on trace le mieux possible la tangente au point d'inflexion Q de la courbe. On mesure ensuite le temps T_u correspondant au point d'intersection entre l'axe des abscisses et la tangente ainsi que le temps T_a « temps de montée de la tangente ».

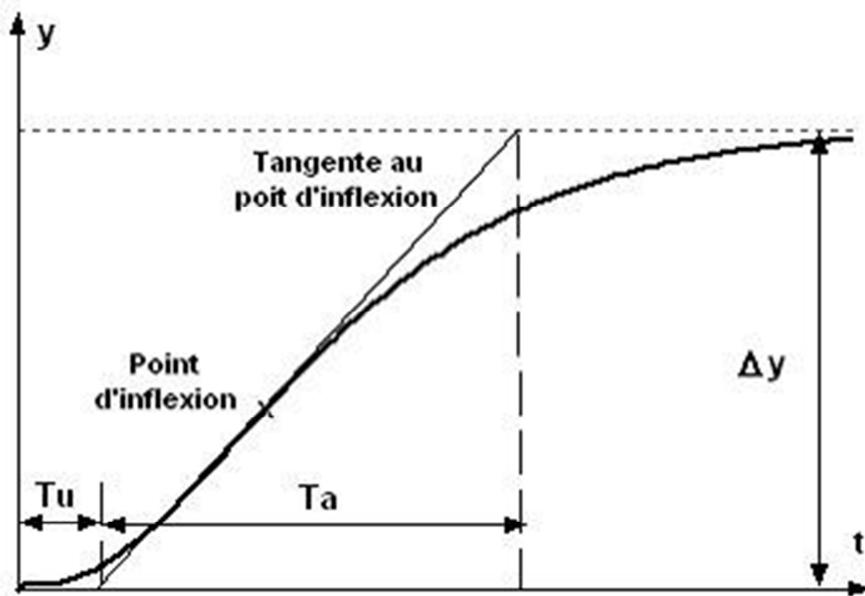


Figure 4.50. Réponse indicielle.

- Réglage du régulateur PID

Ziegler&Nichols proposent de calculer les paramètres du régulateur P, PI ou PID à l'aide des recommandations suivantes:

Réglage des paramètres			
Régulateur	K_p	K_i	K_d
P : $R(p) = K_p$		*	*
PI : $R(p) = K_p(1 + \frac{1}{T_i p})$	$\frac{T_a 0.9}{T_u K}$	3.33 Tu	*
PID : $R(p) = K_p(1 + T_d p + \frac{1}{T_i p})$	$\frac{T_a 1.2}{T_u K}$	2.0 Tu	0.5 Tu

Tableau 1. Type de régulateur selon la méthode de Ziegler & Nichols.

b. Méthode de réglabilité

Les méthodes précédentes peuvent conduire à des réponses en boucle fermée très oscillante, ce qui est particulièrement gênant lors des changements de consigne.

La présente méthode dite de réglabilité constitue une version adoucie des réglages précédents. Elle est basée sur la réponse indicielle du système (en boucle ouverte : pas de régulateur).

Le mode opératoire et l'exploitation de la réponse indicielle sont ceux de la première méthode de Ziegler & Nichols.

Les paramètres du régulateur sont déterminés en fonction du « coefficient de réglabilité » défini par le rapport :

$$r = \frac{T_u}{T_a}$$

avec {

Tu : temps de retard

Ta : constante de temps

Il s'agit du même rapport qui intervient dans la première méthode de Ziegler&Nichols basée sur la réponse indicielle. Il traduit l'importance du retard T_u par rapport à la constante de temps T_a .

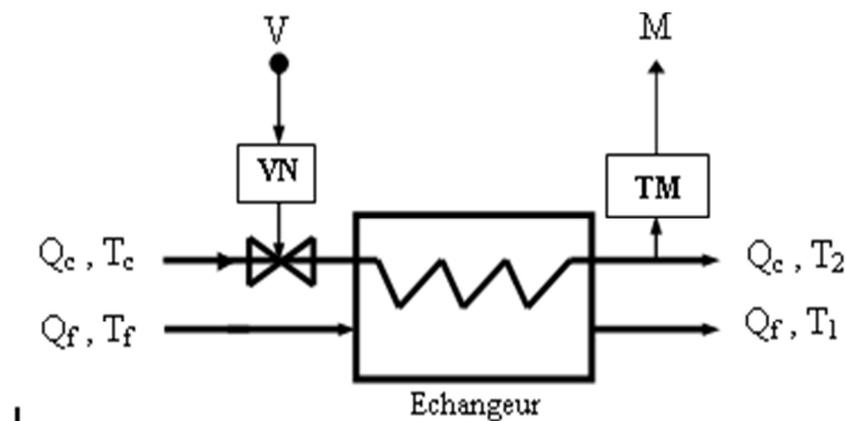
Le tableau suivant fournit des relations empiriques pour calculer les coefficients du régulateur et déterminer son type selon le rapport r :

r	K_p	K_i	K_d
0.05 à 0.1	$\frac{5}{K}$	T_a	0
0.1 à 0.2	$\frac{0.5}{K_r}$	T_a	0
0.2 à 0.5	$\frac{0.5 (1+0.5r)}{K_r}$	$T_a (1 + 0.5 r)$	$T_a \frac{0.5r}{0.5r+1}$
Au delà	PID non recommandé		

Tableau 2. Type de régulateur selon le rapport r .

4.3.2 Procédure expérimentale

La figure 4.51 représente le schéma simplifié d'un échangeur thermique de chaleur. Le but de l'installation est la régulation de la température T_c du fluide à la sortie de l'échangeur.



*Figure 4.51.*Echangeur.

On note:

- Q_c : débit du fluide chaud de température T_c , c'est la grandeur de réglage. Il est ajusté par un ensemble servomoteur-vanne en agissant sur la tension V .
- Q_f : débit du fluide froid de température T_f .
- T_2 : température à la sortie de l'échangeur, grandeur à régler.
- M : grandeur mesurée par le capteur-transmetteur de mesure TM .
- VN : servomoteur-vanne.

Le schéma fonctionnel de l'installation en boucle fermée est représentée par la figure 4.52 où H_1, H_2, H_3 représentent les fonctions de transfert perturbatrices (entre les entrées secondaires et la grandeur réglée) et G représente la fonction de transfert réglante (entre l'entrée réglante et la grandeur réglée).

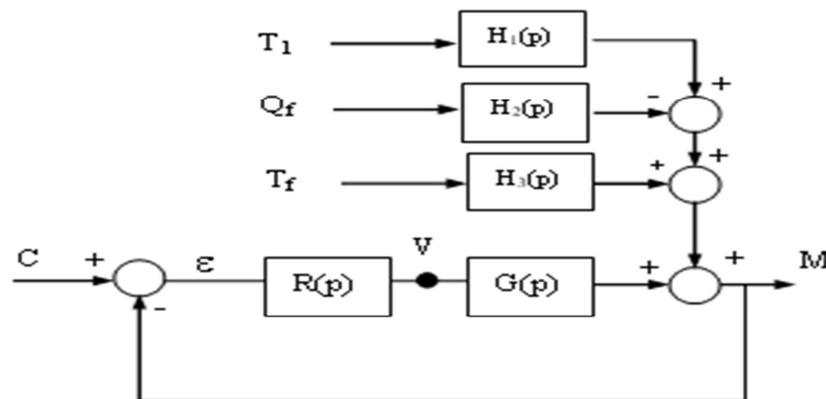


Figure 4.52. Schéma fonctionnel de l'installation.

$R(p)$ est un régulateur PID:

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

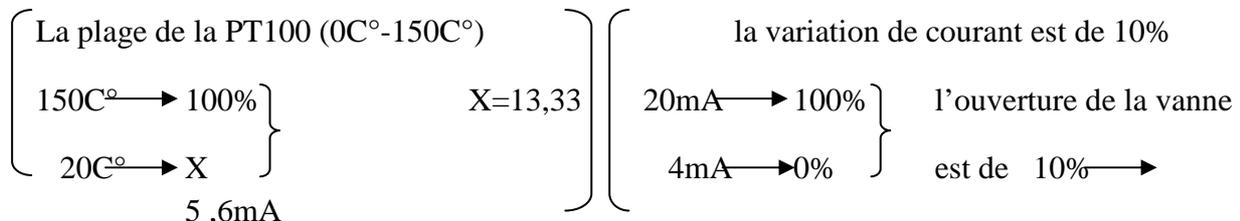
Les résultats des essais en boucle ouverte (réponse indicielle) avec la méthode de réglabilité sont présentés ci-dessous :

Méthode de réglabilité

Le régulateur est en mode manuelle. On provoque une variation $\Delta I = 10\%$ (un échelon ΔU) sur l'entrée de commande (la vanne). La réponse du procédé est dessinée par MATLAB à l'aide des points mesurés expérimentalement comme la montre la figure 4.53

Temps(s)	0	40	80	120	160	200	240	280	320	360	400	440
Température(C°)	20	20,5	23,5	30,4	43,6	54	63	67,9	71	71,7	72	72

Tableau 3. Les mesures des températures en C° par rapport au temps.



Temps(s)	0	40	80	120	160	200	240	280	320	360	400	440
Température(%)	13,33	13,7	15,3	20,3	29,1	36	42	45,33	47,33	47,8	48	48

Tableau 4. Les mesures des températures en % par rapport au temps.

Programme MATLAB pour dessiner la réponse indicielle (figure4.53):

Close all;

T(s) = 0:40:460; %le temps

P(%) = [13.33 13.7 15.3 20.3 29.1 36 42 45.33 47.33 47.8 48 48]; % pourcentage de la température mesuré T (c°).

xx = 0:10:460;

yy = **spline**(T(s), P(%), xx);

plot (T(s), P(%),'o',xx,yy); % la réponse indicielle du système

title ('la réponse indicielle du système')

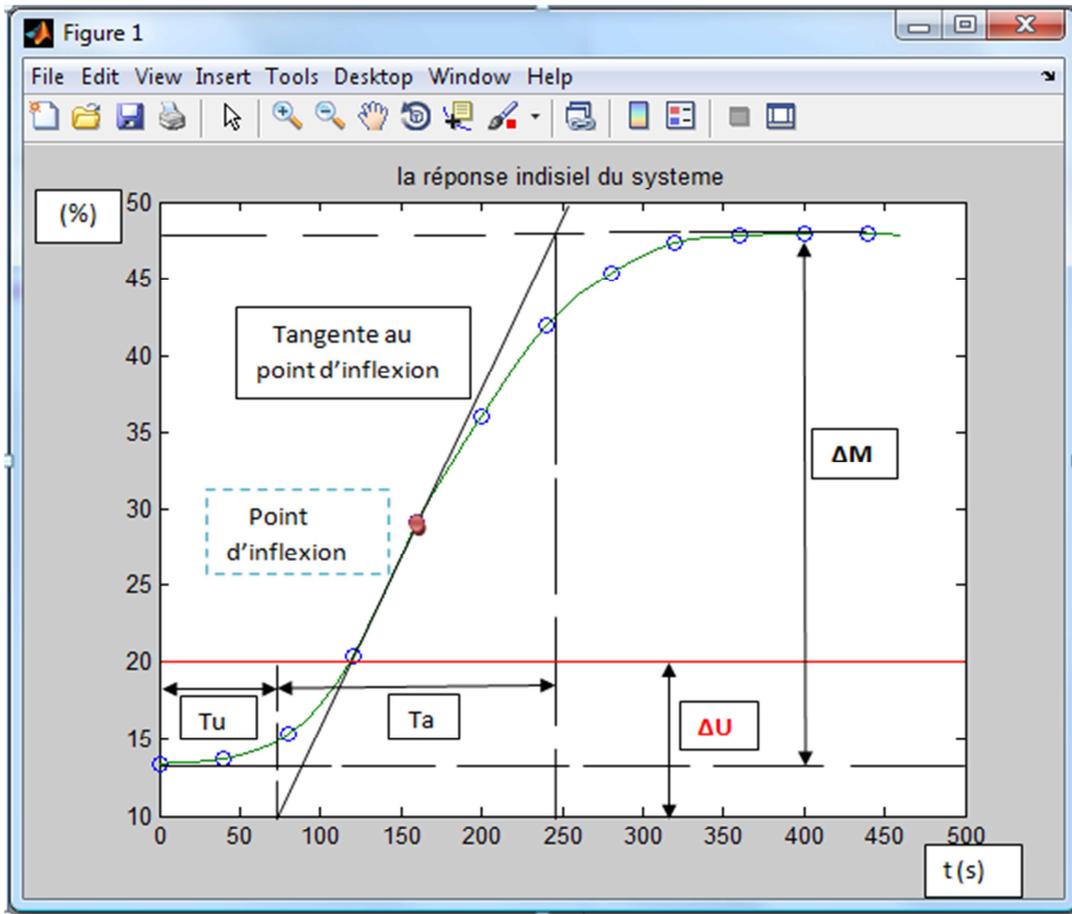


Figure 4.53. Réponse indicielle du système enregistré expérimentalement.

A partir de la réponse indicielle, on relève les paramètres suivants :

$$K = \Delta M / \Delta I = [(48 - 13.33) / 10]$$

$K = 3.47$ le gain statique.
 $T_u = 72$ s le temps de retard.
 $T_a = 168$ s la constante du temps.

Le coefficient de réglabilité $r = T_u / T_a = 0.428$

$0.2 < r < 0.5$ alors: $K_p = [0.5(1 + 0.5*r)] / K * r = 0.40$

$K_i = T_a (1 + 0.5*r) = 203.86$

$K_p = 0.40$

$k_i = 203.86$

$k_d = 29.54$

$K_d = T_a * (0.5*r / (0.5*r + 1)) = 29.54$

4.3.3 Intégration des paramètres de régulation dans logiciel STEP7

a. Le bloc FB41

Le bloc FB41 (figure 4.54) sert à régler des processus industriels à grandeurs d'entrée et de sortie continues sur les automates programmables SIMATIC S7. Le paramétrage permet d'activer ou de désactiver des fonctions partielles du régulateur PID et donc d'adapter ce dernier au système réglé.

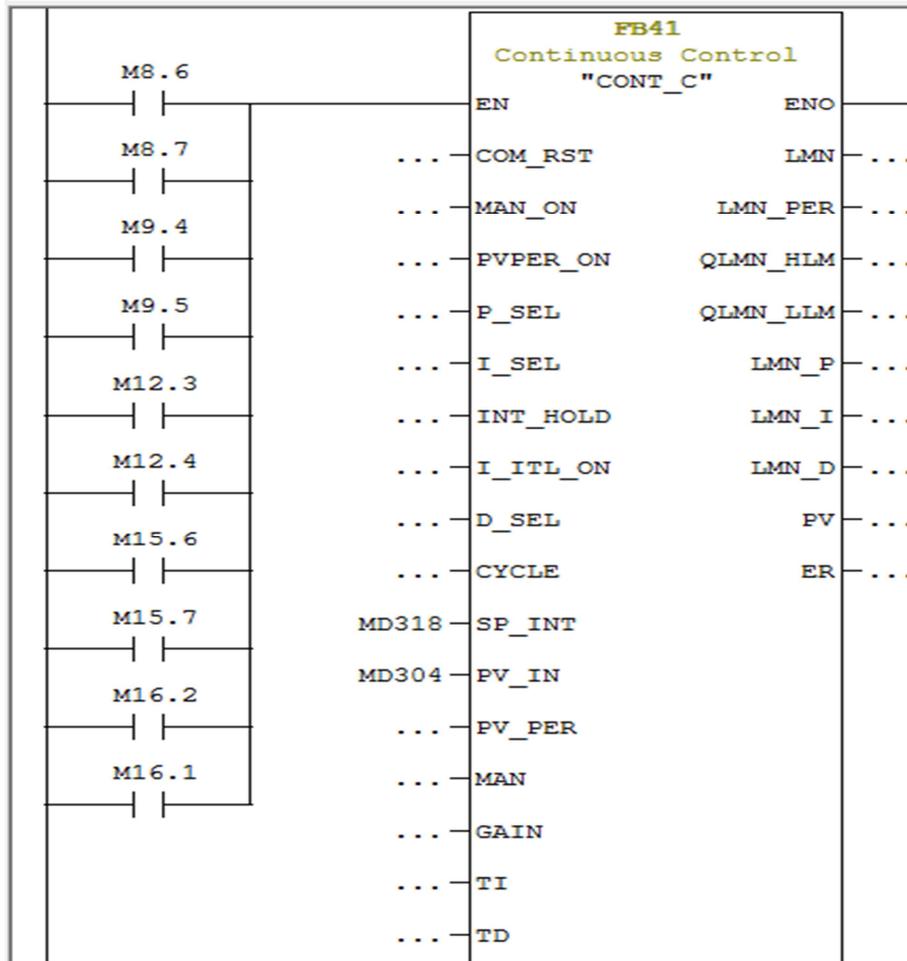


Figure 4.54. Bloc fonctionnel FB41.

b. Le bloc OB35

L'OB35 (figure 4.55) c'est un bloc d'alarme cyclique permettant de démarrer des programmes chaque 100ms.

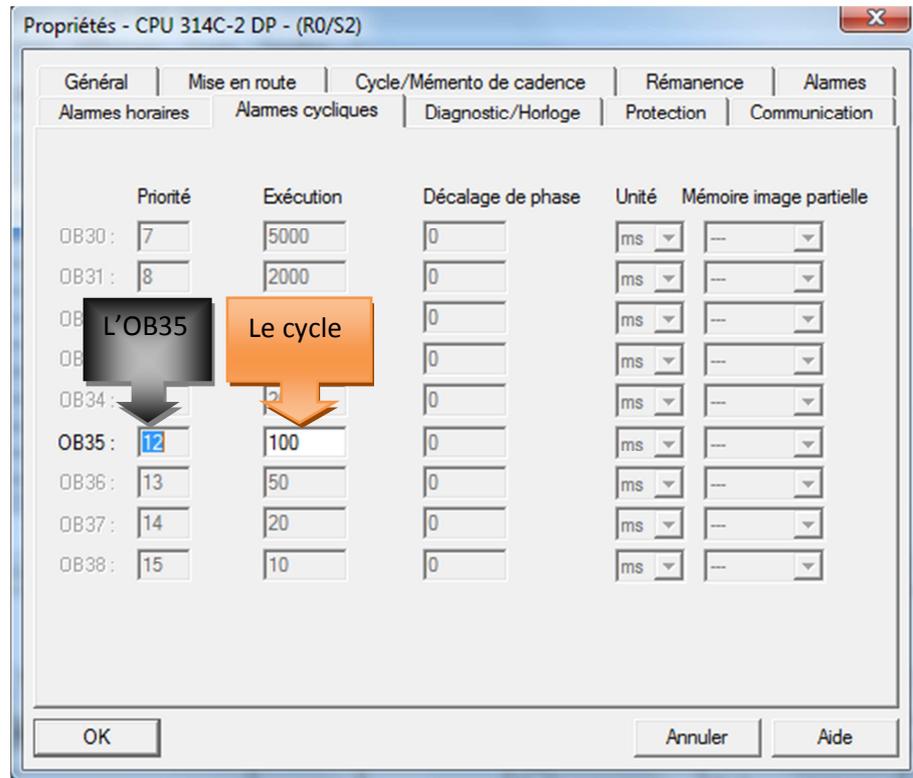


Figure 4.55. Bloc d'organisation OB35.

4.4 Conclusion

Dans ce chapitre on a mis en évidence le travail que nous avons développé afin de répondre le mieux au cahier de charges, à savoir : la modernisation et l'automatisation d'une station de nettoyage existante dans l'entreprise Vitajus, les résultats obtenus sont satisfaisantes, et pour la sécurité du personnels et du matériels nous avons prévus des alarmes et arrêt immédiat de la machine (selon le type d'alarme).

Table des matières

Introduction générale	1
Chapitre 01 : Présentation générale sur l'industrie VitaJus	
1.1 Présentation de l'usine VitaJus.....	3
1.2 L'unité de production	3
1.3 Moyens matériels	5
1.3.1 Machines de conditionnements	5
1.3.2 Cuves de préparation et stockage de jus.....	5
1.3.3 Les pasteurisateurs.....	6
1.4 La station de nettoyage et son rôle dans l'usine	7
1.5 Conclusion.....	8
Chapitre 02 : Étude détaillé sur la station de nettoyage NEP	
2.1 Introduction	9
2.2 Principe de fonctionnement de nettoyage en place NEP	9
2.3 Les cycles de nettoyage	10
2.3.1 Cycle court	10
2.3.2 Cycle Long	10
2.4 Description des composants utilisés	12
2.4.1 Automate programmable industriel S7-200 (SIEMENS).....	12
2.4.2 Les Pompes.....	13
2.4.3 Le réservoir de circulation (BDL)	14
2.4.4 L'Echangeur de chaleur à plaques.....	14
2.4.5 Transmetteur de conductivité (conductivimètre).....	15
2.4.6 Les capteurs.....	15
2.4.7 Les électrovannes	17
2.4.8 Le purgeur de vapeur.....	18
2.4.9 Le clapet anti-retour	19
2.4.10 Le régulateur de température PID externe.....	19
2.5 Préparation des solutions de nettoyage (l'Auto-préparation)	20
2.6 Conclusion.....	21

Chapitre 03 : Généralités sur l'API et le système de communication IHM

3.1	Introduction	22
3.2	Généralités sur les API	22
3.2.1	Historique	22
3.2.2	Définition.....	23
3.2.3	Logiciel de programmation des API SIEMENS <<Step7>>.....	25
3.2.4	Applications de STEP7.....	25
3.2.5	Organisation d'un programme utilisateur.....	31
3.2.6	Les langages de programmation.....	35
3.3	Le système IHM.....	38
3.3.1	Historique	38
3.3.2	Définition.....	38
3.3.3	Les avantages de système IHM	39
3.3.4	Logiciel de programmation WinCC flexible.....	39
3.4	Conclusion.....	46

Chapitre 04 : Amélioration de la station de nettoyage NEP (Partie Pratique)

4.1	Matériels proposés (Hardware)	47
4.1.1	CPU 314C-2DP (SIEMENS)	47
4.1.2	Les modules.....	49
4.1.3	Le câble de communication MPI.....	49
4.1.4	Le pupitre digital MP 277 10" Touch 1.1.4.0.....	49
4.1.5	Les capteurs.....	50
4.2	Programmation et mise en service (Software).....	52
4.2.1	Les GRAFCETS.....	52
4.2.2	Programmation STEP7.....	52
4.2.3	Programmation WinCC flexible.....	65
4.3	Etude de régulation.....	79
4.3.1	les méthodes empiriques utilisées	79
4.3.2	Procédure expérimentale	82
4.3.3	Intégration des paramètres de régulation dans logiciel STEP7	86
4.4	Conclusion.....	87

Conclusion générale	88
Annexe A.....	90
Annexe B.....	91
Bibliographie.....	97

Annexe B

Table des mnémoniques :

pompe de pression	A	0.0	BOOL
pompe doseuse acide	A	0.1	BOOL
pompe doseuse soude	A	0.2	BOOL
CIP pret	A	0.3	BOOL
CIP en marche	A	0.4	BOOL
CIP en panne	A	0.5	BOOL
vanne de pompe pression	A	1.0	BOOL
vanne après échangeur	A	1.1	BOOL
vanne de retour BDL	A	1.2	BOOL
vanne d'envoi d'soude	A	1.3	BOOL
vanne tank soude	A	1.4	BOOL
vanne d'envoi d'acide	A	1.5	BOOL
vanne tank acide	A	1.6	BOOL
vanne BDL	A	1.7	BOOL
vanne d'égout	A	2.0	BOOL
vanne d'envoi CIP	A	2.1	BOOL
vanne de retour	A	2.2	BOOL
vanne vapeur	A	2.3	BOOL
pompe retour cip	A	2.4	BOOL
temp	DB	3	DB 3
variable temp	DB	4	DB 4
les temps	DB	5	DB 5

feedback A1.0	E	0.0	BOOL
feedback A1.1	E	0.1	BOOL
feedback A1.2	E	0.2	BOOL
feedback A1.3	E	0.3	BOOL
feedback A1.4	E	0.4	BOOL
feedback A1.5	E	0.5	BOOL
feedback A1.6	E	0.6	BOOL
feedback A1.7	E	0.7	BOOL
feedback A2.0	E	1.0	BOOL
entrer défaut pompe A0.0	E	1.1	BOOL
défaut pompe A2.4	E	1.2	BOOL
feedback A2.1	E	1.3	BOOL
feedback A2.2	E	1.4	BOOL
feedback A2.3	E	1.5	BOOL
niveau bas BDL	E	1.6	BOOL
niveau bas tank acide	E	1.7	BOOL
niveau bas tank soude	E	2.0	BOOL
niveau haut BDL	E	2.1	BOOL
niveau haut tank acide	E	2.3	BOOL
niveau haut tank soude	E	2.4	BOOL
demande préincage	E	2.5	BOOL
demande l'eau chaude	E	2.6	BOOL
demande soude	E	2.7	BOOL
demande acide	E	3.0	BOOL
demande vidange	E	3.1	BOOL
niveau moyen TANK soude	E	3.2	BOOL
niveau moyen tank acide	E	3.3	BOOL
demande CIP machine	E	3.4	BOOL
autopréparation	FB	1	FB 1
nettoyage tank/ligne	FB	2	FB 2

nettoyage ligne	FB	3	FB	3	
nettoyage machine	FB	4	FB	4	
CONT_C	FB	41	FB	41	Continuous Control
machine	FC	1	FC	1	
sortie	FC	2	FC	2	
tank/ligne soude	FC	3	FC	3	
tank/ligne acide	FC	4	FC	4	
ligne acide	FC	5	FC	5	
ligne soude	FC	6	FC	6	
conductivité sup a 0.5 %	M	30.0	BOOL		
conductivité inf a 0.5%	M	30.1	BOOL		
conductivité entre 1.7&2	M	30.2	BOOL		
conductivité sup a 2.3	M	30.3	BOOL		
conductivité inf 1.7	M	30.4	BOOL		
bouton nettoyage Tank1	M	40.0	BOOL		
bouton nettoyage Tank5	M	40.1	BOOL		
bouton nett ligne pack	M	40.2	BOOL		
bouton nett ligne verre	M	40.3	BOOL		
bouton depart	M	40.4	BOOL		
bouton rearmement	M	40.5	BOOL		
nettoyage soude/acide	M	40.6	BOOL		
nettoyage soude	M	40.7	BOOL		
nettoyage acide	M	41.0	BOOL		
auto_préparation/acide	M	41.1	BOOL		
auto_preparation/soude	M	41.2	BOOL		
bouton arret	M	41.3	BOOL		
bouton auto_preparation	M	41.4	BOOL		
temperature demandé	M	50.0	BOOL		
fin autopréparation	M	400.0	BOOL		
réinitialiser	M	400.1	BOOL		
fin nettoyage circuit	M	400.2	BOOL		

fin nettoyage machine	M	400.5	BOOL
man on A1.7	M	500.0	BOOL
man off A1.7	M	500.1	BOOL
man on A1.0	M	500.2	BOOL
man off A1.0	M	500.3	BOOL
man on A0.0	M	500.4	BOOL
man off A0.0	M	500.5	BOOL
man on A1.1	M	500.6	BOOL
man off A1.1	M	500.7	BOOL
man on A2.1	M	501.0	BOOL
man off A2.1	M	501.1	BOOL
man on A2.2	M	501.2	BOOL
man off A2.2	M	501.3	BOOL
man on A2.0	M	501.4	BOOL
man off A2.0	M	501.5	BOOL
man on A1.4	M	501.6	BOOL
man off A1.4	M	501.7	BOOL
man on A1.6	M	502.0	BOOL
man off A1.6	M	502.1	BOOL
man on A1.3	M	502.2	BOOL
man off A1.3	M	502.3	BOOL
man on A1.5	M	502.4	BOOL
man off A1.5	M	502.5	BOOL
man on A0.1	M	502.6	BOOL
man off A0.1	M	502.7	BOOL
man on A0.2	M	503.0	BOOL
man off A0.2	M	503.1	BOOL
man on A1.2	M	503.2	BOOL
man off A1.2	M	503.3	BOOL
man on A2.3	M	503.4	BOOL
man off A2.3	M	503.5	BOOL

man on A2.4	M	503.6	BOOL	
man off A2.4	M	503.7	BOOL	
défaut vanne A1.7	M	3000.0	BOOL	
défaut vanne A1.0	M	3000.1	BOOL	
défaut vanne A1.1	M	3000.2	BOOL	
défaut vanne A2.1	M	3000.3	BOOL	
défaut vanne A2.2	M	3000.4	BOOL	
défaut vanne A2.0	M	3000.5	BOOL	
défaut vanne A1.2	M	3000.6	BOOL	
défaut vanne A2.3	M	3000.7	BOOL	
défaut vanne A1.3	M	3001.0	BOOL	
défaut vanne A1.5	M	3001.1	BOOL	
défaut vanne A1.6	M	3001.2	BOOL	
défaut vanne A1.4	M	3001.3	BOOL	
demande autopré soude	M	3001.4	BOOL	
demande autopré acide	M	3001.5	BOOL	
pompe A0.0 corrompu	M	3001.6	BOOL	
pompe A2.4 corrompu	M	3001.7	BOOL	
alarme	M	3002.0	BOOL	
entrée Pt100	MD	300	DWORD	
entrée conductivité	MD	1000	DWORD	
Grafcet Maitre	OB	1	OB 1	
CYC_INT5	OB	35	OB 35	Cyclic Interrupt 5
temp de rincage (froide)	T	1	TIMER	
temp de rincage (chaude)	T	2	TIMER	
temp de vidange BDL	T	3	TIMER	
vidange complet (tk/lig)	T	6	TIMER	
circulation de la soude	T	7	TIMER	
circulation de l'acide	T	8	TIMER	
remplissage soude acide	T	10	TIMER	
temp de mélange d'autop	T	11	TIMER	

temp rincage apré autopr	T 12	TIMER
temp dajout auto	T 13	TIMER
VAT_1	VAT 1	

Les Grafjets :

- 1- Grafjet maître.
 - 2- Grafjet Auto-préparation.
 - 3- Grafjet circuits lignes et tanks.
 - 4- Grafjet circuit machine.
- Les Grafjets sont présentés sur des feuil de format A3.